УДК 004.032.26:004.056.523 DOI: 10.18698/2541-8009-2023-4-886

СИСТЕМА РАСПОЗНАВАНИЯ ВРЕДОНОСНЫХ ПРОГРАММ НА ОСНОВЕ ПРЕДСТАВЛЕНИЯ БИНАРНОГО ФАЙЛА В ВИДЕ ИЗОБРАЖЕНИЯ С ПРИМЕНЕНИЕМ МАШИННОГО ОБУЧЕНИЯ

Н.И. Панчехин panchekhinni@student.bmstu.ru
А.Г. Десятов dag21um015@student.bmstu.ru
А.Д. Сидоркин sidorkinad@student.bmstu.ru

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

Аннотация

В связи с увеличением числа и сложности угроз, исходящих от вредоносного программного обеспечения, активно исследуются подходы к автоматическому обнаружению и классификации вредоносных программ. В то же время авторы вредоносных программ разрабатывают средства обхода методов обнаружения вредоносного программного обеспечения на основе сигнатур, используемых антивирусными компаниями. С недавних пор для решения этой проблемы в классификации вредоносных программ применяется глубокое обучение. В данной работе рассмотрена модель сверточной нейронной сети (CNN), полученная экспериментальным путем и предназначенная для обнаружения вредоносных программ на основе статического анализа. Наибольшая точность, достигнутая моделью в ходе исследования, составила 95 %.

Ключевые слова

Обнаружение вредоносных программ, статический анализ, глубокое обучение, искусственная нейронная сеть, программное обеспечение, Python, Keras, набор данных

Поступила в редакцию 27.03.2023

Введение. Вредоносное программное обеспечение (ПО) — это глобальная проблема, количество новых вредоносных программ и их разновидностей экспоненциально растет во всем мире. Их основная цель — получить доступ к удаленным компьютерным сетям или денежную прибыль. На данный момент существует множество определений вредоносного ПО. Например, в работе [1] вредоносный код определяют как «любой код, добавленный, измененный или удаленный из программной системы с целью преднамеренного причинения вреда или нарушения предполагаемой функции системы». В работе [2] авторы определяют вредоносное ПО как универсальный термин, который подразумевает вирусы, трояны, шпионские программы и более агрессивные ПО.

Часто разработчики вредоносных программ проявляют большой интерес к созданию нового варианта вредоносного ПО с помощью различных автоматизированных инструментов (например, metamorphic engine — метаморфический двигатель). Такие инструменты имеют тенденцию внедрять скрытые вредонос-

ные команды в исполняемый файл с использованием однотипных модулей, что генерирует новые варианты вредоносных программ из того же класса. В различных вариантах, генерируемых подобными инструментами, могут применяться методы обфускации (приведение исполняемого кода программы к виду, сохраняющему её функциональность, но затрудняющему анализ, понимание алгоритмов работы и модификацию при декомпиляции) первого порядка, которые маскируют вредоносные команды таким образом, чтобы они успешно проходили традиционный процесс обнаружения антивирусных программ на основе сигнатур. Но данные инструменты сохраняют подозрительные закономерности на байтовом уровне, которые можно идентифицировать.

В данной работе для обнаружения вредоносных программ применен статический анализ на основе представления приложений в виде последовательности байтов и дальнейшего перевода полученных данных в формат изображения. Полученные данные проанализированы с помощью машинного обучения [3], а именно сверточных нейронных сетей.

Обзор литературы. В подходах, использующих машинное обучение, необходимо определить особые признаки для передачи их в модель. Зачастую они извлекаются путем проведения статического и динамического анализа. Например, в работе [4] определены статические признаки, такие как длины функций и информация о стоках вывода. В качестве динамических признаков авторы [4] извлекли функции API (Application Programming Interface — интерфейс прикладного программирования) в результате запуска всех исполняемых файлов и регистрации вызовов Windows API. Число появлений вызовов функций API кодируется как динамический вектор признаков. Как только интегрированные статические и динамические функции получены, для классификации вредоносных программ применяют стандартные классификаторы, такие как SVM (Support Vector Machine — метод опорных векторов) и random forest (алгоритм случайного леса). В [5] авторы также извлекают статические и динамические признаки из устройств на базе Android, такие как последовательности команд и последовательности системных вызовов. Вместо того чтобы напрямую использовать извлеченные признаки, авторы [5] применяют глубокое автоматическое кодирование для изучения и объединения новых признаков для классификации.

Многие работы были сосредоточены на преобразовании бинарных исполняемых файлов в изображения. Например, в работе [6] авторы группируют двоичные последовательности исполняемых файлов по 8-битным векторам. Преобразованные 8-битные векторы затем преобразуются в черно-белые изображения. После процесса преобразования авторы непосредственно применяют алгоритм random forest для классификации вредоносных программ, используя значения пикселей в качестве объектов. В исследовании [7] авторы извлекают визуальные признаки с помощью классических экстракторов объектов компьютерного зрения.

В последнее время сверточные нейронные сети (CNN) продемонстрировали очень хорошие результаты в задачах классификации изображений. Предложено множество основанных на CNN подходов к классификации вредоносных программ, использующих преобразования бинарных исполняемых файлов в изображения. В работе [8] авторы применили различные модели глубокого обучения, такие как CNNs (Convolutional Neural Networks — сверточная нейронная сеть) и GRUs (Gated Recurrent Units —– управляемые рекуррентные блоки), для классификации семейств вредоносных программ. Авторы также применили функцию активации Leaky ReLU (линейный выпрямитель с «утечкой») и функцию потерь L2-SVM (L2 Support Vector Machine —– метод опорных векторов с квадратной суммой фиктивных переменных) в предложенных ими методах. В работе [9] авторы сравнивают предложенные ими методы, основанные на CNN, с традиционными подходами к машинному обучению. Они показывают, что методы классификации вредоносных программ, основанные на CNNs, способны работать лучше, чем методы, основанные на машинном обучении, поскольку необходимые признаки могут быть выявлены автоматически. В работе [10] авторы предлагают классификацию вредоносных программ на основе изображений с использованием точно настроенных CNNs (IMCFN). В отличие от других подходов, основанных на CNN, алгоритм IMCFN преобразует необработанные бинарные файлы вредоносных программ в цветные изображения. Вместо того чтобы обучать CNN с нуля, авторы используют предварительно обученную модель с набором данных ImageNet. Кроме того, в данной работе также применяется аугментация данных (увеличение выборки данных для обучения через модификацию существующих данных) для решения проблемы дисбаланса в классификации вредоносных программ. В работе [11] эксперты по обнаружению вредоносных программ для Android изучили APK-файлы из разных точек входа. Авторы преобразуют двоичные данные в изображения RGB или берут фрагменты данных (входной размер: 512, 1024, 2048, 4096) для классификации на основе CNNs.

Статический анализ на основе представления приложения в виде последовательности байтов. Признаки извлекают из необработанного файла, а именно из его представления в виде последовательности байтов. Преимущество такого подхода заключается в том, что никакой другой информации не требуется. Нет необходимости разбираться в структуре PE-файла (Portable Executable — переносимый исполняемый файл), а также запускать в песочнице (изолированной программной среде с жестко ограниченными ресурсами для выполнения в рамках этой среды программного кода) и исследовать его поведение, поскольку в данном подходе используются простые байты. Эти признаки могут быть сложными для интерпретации, поскольку они отображают только общую информацию о файле. В данной работе полученные данные преобразуются в изображения.

Процедура преобразования приложения в черно-белое изображение представлена на рис. 1. Чтобы преобразовать бинарный файл вредоносного ПО в черно-белое изображение, конвертер последовательно считывает двоичные данные в байтах, преобразует каждый байт в десятичное число в диапазоне [0…255], затем сохраняет число в одномерный массив. Например, 0110000 преобразуется в 96. Каждое число в массиве соответствует пикселю на черно-белом изображении. Необходимо учитывать, что файлы образцов вредоносных программ имеют разные размеры, следовательно, разрешения (размеры по ширине и высоте) полученных из них изображений будут различаться.

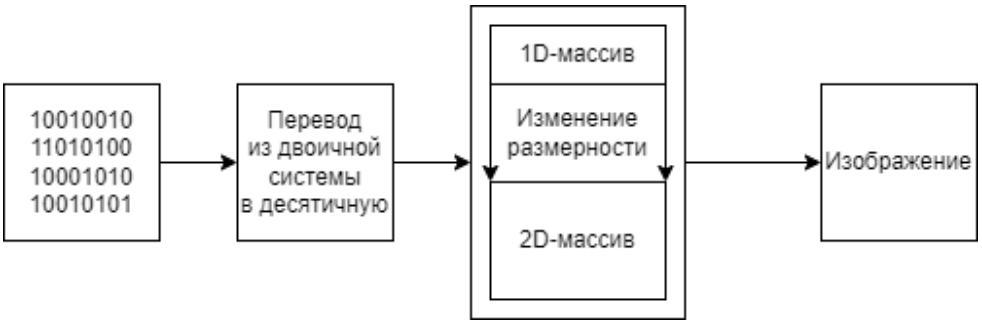

Рис. 1. Процедура преобразования бинарного файла в черно-белое изображение

Существует простой подход к изменению формы данных изображения (массива), при котором используют рекомендуемую фиксированную ширину с переменной высотой в соответствии с размером файла, который соответственно преобразуется в двухмерное черно-белое изображение, соответствующее формату portable network graphics (портативная сетевая графика, расширение .png) с помощью встроенной функции imwrite (записать) или imsave (сохранить) в библиотеке cv2 [12]. Рекомендуемая фиксированная ширина изображения для различных размеров файлов вредоносных программ приведена в табл. 1.

Таблица 1

Разрешение изображения в зависимости от размера

Размер файла, кБе	Ширина	Высота
Менее 10	32	(0, 312]
10–30	64	(156, 468]
30–60	128	(234, 468]
60–100	256	(234, 390]
100–200	384	(260, 520]
200–500	512	(390, 976]
500–1000	768	(651, 1302]
Более 1000	1024	(976, ∞]

В данной работе используется преобразование бинарного файла в RGB-изображение. Подход схож с подходом, рассмотренным ранее, однако на каждом шагу считывается не один бит, а последовательность из трех (каждый бит соответствует своему цвету — красный/зеленый/синий), таким образом образуя трехмерный массив.

Обучение модели обнаружения вредоносных программ на основе изображений, полученных из бинарных файлов. В этом разделе подробно описаны подготовка к обучению, настройка необходимых параметров, само обучение и анализ результатов.

Используемый набор данных (датасет) содержит порядка 3700 вариантов образцов вредоносных файлов, а также порядка 600 вариантов образцов обычных файлов. В качестве обучающих данных использовалось 75 % объема датасета, оставшиеся 25 % использовались для проверки результатов обучения (тестовая выборка).

Экспериментируя с количеством слоев, их последовательностью и размерностью ядер свертки с целью уменьшения количества ошибок первого и второго рода, получена модель, структура которой представлена на рис. 2.

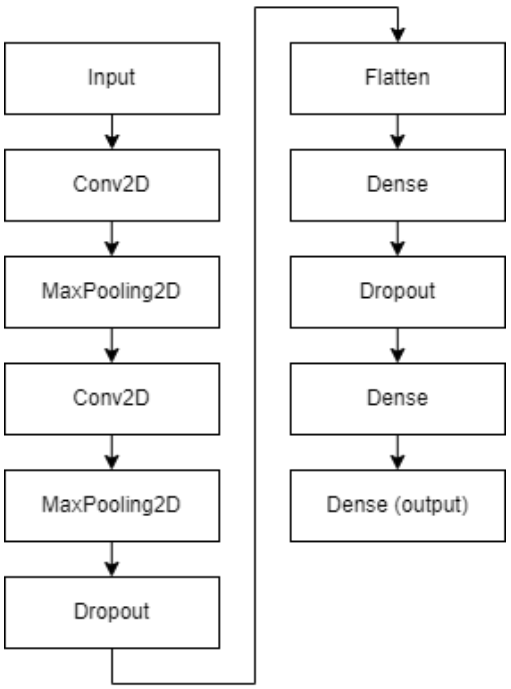

Рис. 2. Структура модели сверточной нейронной сети:
Input — входной слой, Conv2D — слой двухмерной свертки, MaxPooling2D — слой подвыборки, Dropout — слой исключения, Flatten — слой изменения размерности, Dense — полносвязный слой, Dense (output) — полносвязный выходной слой)

Обучение модели проводилось на протяжении 32 эпох. Результаты обучения представлены на рис. 3.

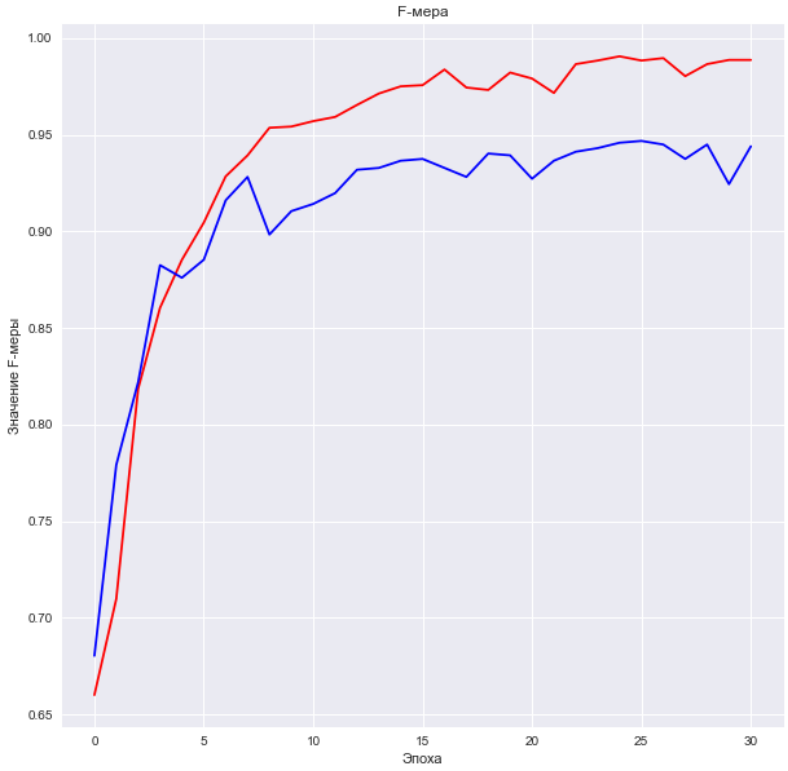

Рис. 3. Графики F-меры при обучении (*красный цвет*) модели нейронной сети и тестировании (*синий цвет*)

На основе полученных графиков можно сделать вывод, что наиболее полного обучения модель достигла начиная с 20-й эпохи, поскольку с тех пор значение отслеживаемых метрик почти не изменятся. Также стоит отметить, что *F*-мера имеет значения, близкие к единице, что свидетельствует о способности системы хорошо обнаруживать и различать исследуемые классы. В результате обучения точность модели составила 95 %.

По результатам тестирования модели составлена матрица ошибок, представленная в табл. 2.

Таблица 2

Матрица смежности результатов работы модели

Элемент матрицы	*P* (истинный класс)	*N* (истинный класс)
P (определенный класс)	891	9
N (определенный класс)	40	133

Проанализировав полученные результаты, можно сделать вывод, что модель достаточно точно распознает вредоносные файлы. Кроме того, ошибка второго рода составляет меньше 1 %, что также улучшает результаты распознавания.

Заключение. В статье описан практический способ распознавания вредоносных программ на основе статического анализа, а именно, представления бинарного файла в виде изображения, с применением CNN. В статье построены графики *F*-меры, определена таблица смежности, проанализированы результаты обученной модели.

Литература

[1] McGraw G., Morrisett G. Attacking Malicious Code: A Report to the Infosec Research Council. *IEEE Software*, 2000, vol. 17 (5), pp. 33–41. http://doi.org/10.1109/52.877857

[2] Vasudevan A., Yerraballi R. SPiKE: engineering malware analysis tools using unobtrusive binary-instrumentation. *Computer Science 2006, Twenty-Nineth Australasian Computer Science Conference (ACSC2006)*, Hobart, Tasmania, Australia, 2006. http://doi.org/10.1145/1151699.1151734

[3] Басараб М.А., Коннова Н.С. *Интеллектуальные технологии на основе искусственных нейронных сетей*. Москва, МГТУ им. Н.Э. Баумана, 2017, 56 с.

[4] Rafiqul Islam M., Tian R., Batten L., Versteeg S. Classification of malware based on integrated static and dynamic features. *Journal of Network and Computer Applications*, 2013, vol. 36 (2), pp. 646–656. http://doi.org/10.1016/j.jnca.2012.10.004

[5] Xu L., Zhang D., Jayasena N., Cavazos J. HADM: Hybrid Analysis for Detection of Malware. *Proceedings of SAI Intelligent Systems Conference*, 2018. http://doi.org/10.1007/978-3-319-56991-8_51

[6] *Random Forest for Malware Classification*. URL: https://arxiv.org/abs/1609.07770 (accessed March 17, 2023).

[7] Nataraj L., Karthikeyan S., Jacob G., Manjunath B.S. *Malware Images: Visualization and Automatic Classification*, 2011. http://doi.org/10.1145/2016904.2016908

[8] *Towards Building an Intelligent Anti-Malware System: A Deep Learning Approach using Support Vector Machine (SVM) for Malware Classification*. URL: https://arxiv.org/abs/1801.00318 (accessed March 18, 2023).

[9] Kalash M., Rochan M., Mohammed N., Bruce N.D.B. Classification with Deep Convolutional Neural Networks. *9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, 2018. http://doi.org/10.1109/NTMS.2018.8328749

[10] Vasan D., Alazab M., Wassan S., Naeem H. IMCFN: Image-based malware classification using fine-tuned convolutional neural network architecture. *Computer Networks*, 2020, vol. 171, pp. 107138. http://doi.org/10.1016/j.comnet.2020.107138

[11] *R2-D2: ColoR-inspired Convolutional NeuRal Network (CNN)-based AndroiD Malware Detections*. URL: https://arxiv.org/pdf/1705.04448.pdf (accessed March 19, 2023).

[12] *Malware-on-the-Brain: Illuminating Malware Byte Codes with Images for Malware Classification*. URL: https://arxiv.org/abs/2108.04314 (accessed March 20, 2023).

Панчехин Никита Игоревич — студент кафедры «Информационная безопасность», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Десятов Александр Геннадьевич — студент кафедры «Информационная безопасность», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Сидоркин Антон Дмитриевич — студент кафедры «Информационная безопасность», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Ссылку на эту статью просим оформлять следующим образом:
Панчехин Н.И., Десятов А.Г., Сидоркин А.Д. Система распознавания вредоносных программ на основе представления бинарного файла в виде изображения с применением машинного обучения. *Политехнический молодежный журнал*, 2023, № 04 (81). http://dx.doi.org/10.18698/2541-8009-2023-4-886

MALWARE RECOGNITION SYSTEM BASED ON THE REPRESENTATION OF A BINARY FILE IN THE FORM OF AN IMAGE WITH MACHINE LEARNING

N.I. Panchekhin panchekhinni@student.bmstu.ru
A.G. Desyatov dag21um015@student.bmstu.ru
A.D. Sidorkin sidorkinad@student.bmstu.ru

Bauman Moscow State Technical University, Moscow, Russian Federation

Abstract

With the increasing number and complexity of threats posed by malware, approaches to automatically detect and classify malware are being actively explored. At the same time, malware authors are developing tools to circumvent malware detection methods based on the signatures used by antivirus companies. Recently, deep learning has been used to solve this problem in malware classification. In this paper, we consider a model of a convolutional neural network (CNN), obtained experimentally and designed to detect malware based on static analysis. The highest accuracy achieved by the model during the study was 95 %.

Keywords

Malware detection, static analysis, deep learning, artificial neural network, software, Python, Keras, dataset

Received 27.03.2023

References

[1] McGraw G., Morrisett G. Attacking Malicious Code: A Report to the Infosec Research Council. *IEEE Software*, 2000, vol. 17 (5), pp. 33–41. http://doi.org/10.1109/52.877857

[2] Vasudevan A., Yerraballi R. SPiKE: engineering malware analysis tools using unobtrusive binary-instrumentation. *Computer Science 2006, Twenty-Ninth Australasian Computer Science Conference (ACSC2006)*, Hobart, Tasmania, Australia, 2006. http://doi.org/10.1145/1151699.1151734

[3] Basarab M.A., Konnova N.S. *Intellektual'nye tekhnologii na osnove iskusstvennykh neyronnykh setey* [Intelligent technologies based on artificial neural networks]. Moscow, BMSTU Press, 2017, 56 p. (In Russ.).

[4] Rafiqul Islam M., Tian R., Batten L., Versteeg S. Classification of malware based on integrated static and dynamic features. *Journal of Network and Computer Applications*, 2013, vol. 36 (2), pp. 646–656. http://doi.org/10.1016/j.jnca.2012.10.004

[5] Xu L., Zhang D., Jayasena N., Cavazos J. HADM: Hybrid Analysis for Detection of Malware. *Proceedings of SAI Intelligent Systems Conference*, 2018. http://doi.org/10.1007/978-3-319-56991-8_51

[6] *Random Forest for Malware Classification*. URL: https://arxiv.org/abs/1609.07770 (accessed March 17, 2023).

[7] Nataraj L., Karthikeyan S., Jacob G., Manjunath B.S. *Malware Images: Visualization and Automatic Classification*, 2011. http://doi.org/10.1145/2016904.2016908

[8] *Towards Building an Intelligent Anti-Malware System: A Deep Learning Approach using Support Vector Machine (SVM) for Malware Classification*. URL: https://arxiv.org/abs/1801.00318 (accessed March 18, 2023).

[9] Kalash M., Rochan M., Mohammed N., Bruce N.D.B. Classification with Deep Convolutional Neural Networks. *9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, 2018. http://doi.org/10.1109/NTMS.2018.8328749

[10] Vasan D., Alazab M., Wassan S., Naeem H. IMCFN: Image-based malware classification using fine-tuned convolutional neural network architecture. *Computer Networks*, 2020, vol. 171, pp. 107138. http://doi.org/10.1016/j.comnet.2020.107138

[11] *R2-D2: Color-inspired Convolutional NeuRal Network (CNN)-based AndroiD Malware Detections*. URL: https://arxiv.org/pdf/1705.04448.pdf (accessed March 19, 2023).

[12] *Malware-on-the-Brain: Illuminating Malware Byte Codes with Images for Malware Classification*. URL: https://arxiv.org/abs/2108.04314 (accessed March 20, 2023).

Panchekhin N.I. — Student of Department of Information Security, Bauman Moscow State Technical University, Moscow, Russian Federation.

Desyatov A.G. — Student of Department of Information Security, Bauman Moscow State Technical University, Moscow, Russian Federation.

Sidorkin A.D. — Student of Department of Information Security, Bauman Moscow State Technical University, Moscow, Russian Federation.

Please cite this article in English as:
Panchekhin N.I., Desyatov A.G., Sidorkin A.D. Malware recognition system based on the representation of a binary file in the form of an image with machine learning. *Politekhnicheskiy molodezhnyy zhurnal*, 2023, no. 04 (81). (In Russ.).
http://dx.doi.org/10.18698/2541-8009-2023-4-886