Научная статья

УДК 004.032.26:004.056

ББК 32.818.1

Ч 48

doi: 10.53598/2410-3225-2021-2-281-90-96

Разработка модели обнаружения вредоносных программ на основе анализа последовательностей API-запросов (Рецензирована)

Александр Николаевич Черкасов¹, Егор Алексеевич Туркин²

^{1,2} Кубанский государственный технологический университет, Краснодар, Россия

¹ cherk@mail.ru

² wolfutur@mail.ru

Аннотация. Анализ актуальности существующих работ по изучению нейросетевых алгоритмов обнаружения вредоносного программного обеспечения позволил определить оптимальный подход к решению задачи распознавания вирусов. Был сформирован набор данных, содержащий последовательности системных запросов для 2083 исполняемых файлов. Обучение модели рекуррентной нейронной сети на полученном наборе данных позволило получить классификатор с точностью распознавания вредоносных программ 97,6%. Ключевой аспект значимости работы заключается в практической ориентированности исследования – полученная модель будет применена в качестве компонента системы комплексного анализа вредоносного программного обеспечения.

Ключевые слова: информационная безопасность, нейронная сеть, вредоносная программа, классификация, матрица ошибок

Original Research Paper

Development of the malware detection model based on an analysis of API request sequences

Aleksandr N. Cherkasov¹, Egor A. Turkin²

^{1,2} Kuban State University of Technology, Krasnodar, Russia

¹ cherk@mail.ru

² wolfutur@mail.ru

Abstract. An analysis of the relevance of existing works on the study of neural network algorithms for detecting malicious software made it possible to determine the optimal approach to solving the problem of virus recognition. A dataset has been generated containing sequences of system requests for 2083 executable files. Training the recurrent neural network model on the generated dataset made it possible to obtain a classifier with a malware recognition accuracy of 97.6%. The key aspect of the significance of the work lies in the practical orientation of the research – the resulting model will be applied as a component of a system for complex analysis of malicious software.

Keywords: information security, neural network, malicious software, classification, confusion matrix

Введение

Вредоносное программное обеспечение (ВПО) активно развивается с тех пор, как персональные компьютеры стали повсеместно распространены. За несколько десятилетий появилось огромное множество различных семейств и типов ВПО, способных

осуществлять свое злонамеренное воздействие на различные платформы. С каждым годом проблема распространения вредоносных программ становится все серьезнее, что значительно повышает важность разработки актуальных методов их обнаружения.

В связи с этим сейчас активно развиваются методы обнаружения ВПО с помощью методов машинного обучения и искусственных нейронных сетей. На сегодняшний день можно выделить два основных направления анализа вредоносных программ – методы статического анализа и методы динамического анализа.

Существующие работы

В исследовании [1] благодаря свойствам, извлеченным из исполняемых файлов с помощью программного обеспечения IDA Pro, авторы разработали классификацию файлов на вредоносные и безопасные. Также авторы применили метод динамического анализа, базирующийся на анализе API-запросов приложений, для решения задачи классификации обнаруженных вредоносных программ. Данное исследование демонстрирует реализуемость и актуальность апробированных авторами методов анализа.

Авторы работы [2] генерировали вредоносные сигнатуры, используя вызовы Windows API вирусов. В данной работе был представлен метод идентификации и классификации ВПО, использующий полученные сигнатуры. Представленный авторами метод имел точность определения типа вредоносного приложения порядка 75–80% для каждого семейства. Однако, несмотря на высокие показатели точности, этот метод не является надежным, поскольку злоумышленники могут изменять сигнатуры ВПО или подвергать код программы обфускации.

Еще один подход к анализу ВПО применили авторы исследования [3]. В их работе обнаружение вредоносного программного обеспечения основывается не на характеристиках самого вируса, а на воздействии, которое он осуществляет на систему. То есть данный метод для обнаружения не использует извлеченные поведенческие или структурные свойства ВПО, а анализирует аномальные реакции системы в стандартных условиях. Авторы утверждают, что таким образом могут быть обнаружены продвинутого типа вредоносных программ, такие как полиморфные и метаморфические вирусы.

В статье [4] исследователи апробировали метод визуализации и глубокого обучения для решения задачи обнаружения ВПО. На основе извлеченных API-вызовов в соответствии с правилами сопоставления цветов были сгенерированы изображения, демонстрирующие поведение приложений. Для классификации полученного набора данных была применена сверточная нейронная сеть. Результаты эксперимента показали точность модели около 96%, что подтверждает эффективность данного метода.

Основная часть

В предыдущей работе [5] был рассмотрен один из методов статического анализа вредоносных исполняемых файлов с помощью сверточных нейронных сетей. В настоящей работе будет рассмотрен один из методов динамического анализа исполняемых файлов на основе их поведенческих характеристик. В качестве таких характеристик могут быть использованы такие данные, как вызовы Windows API, операции чтения/записи реестра или Интернет-трафик, связанный с деятельностью программы.

Данное исследование основано на анализе вызовов API, совершаемых программами в операционной системе Windows. Приложение, разработанное для работы в ОС Windows, вызывает интерфейсы, представленные как API, для реализации различных функций в операционной системе. Совокупность вызовов API, совершенных приложением в системе, может показать общий принцип действия этого приложения. Поэтому подход, основанный на вызовах API, широко применяется в динамическом анализе программ.

Описание подготовки к эксперименту

Для анализа был сформирован набор данных, содержащий извлеченные из приложений вызовы Windows API. Для извлечения данных было применено бесплатное общедоступное программное обеспечение – Cuckoo Sandbox. С помощью Cuckoo Sandbox можно запустить вредоносное ПО в виртуальной изолированной среде. В ходе исполнения файла Cuckoo Sandbox регистрирует действия программы и сохраняет их в отчете в формате JSON в базе данных PostgreSQL. Впоследствии из сформированных отчетов извлекались сведения об осуществленных API-вызовах. Общая схема, демонстрирующая подготовку набора данных, представлена на рисунке 1.

Рис. 1. Подготовка набора данных

Поскольку сформированный набор будет использован для обучения искусственной нейронной сети, он был отфильтрован. Так, были отброшены отчеты о действиях программ, содержащие менее 100 вызовов API. Дальнейшая обработка состояла в индексировании запросов: поскольку в ходе сбора данных были определены 287 различных вызова API, эти вызовы были проиндексированы значениями от 0 до 288 включительно. Последним этапом подготовки данных для обучения модели стала разметка набора на сведения о безопасных и вредоносных файлах.

Полученный набор данных получился несбалансированным – около 75% записей соответствовали классу вредоносных файлов. Впоследствии при обучении модели этот факт будет учтен.

Описание модели

В качестве модели классифицирующей программы на основе API-вызовов использовалась рекуррентная нейронная сеть с архитектурой LSTM. Данная архитектура была разработана в связи с тем, что RNN не были достаточно эффективны при обработке долговременных зависимостей. Сети с архитектурой LSTM могут запоминать и изучать любую долгосрочную зависимость через случайные промежутки времени. Это является оптимальным методом анализа данных или событий, имеющих определенную взаимосвязь, особенно распределенную во времени [6].

Реализация архитектуры нейронной сети была выполнена при помощи языка программирования Python 3.7 с использованием библиотеки Keras. Структура сформированной

рованной рекуррентной нейронной сети представлена на рисунке 2.

Данные подаются на Embedding слой нейронной сети. Embedding преобразует входные данные в Embedding-вектор. Embedding-слой предназначен для создания специального вектора, в котором вместо индексов входных данных используются значения, равные весам связи для этих индексов.

Рис. 2. Схема рекуррентной сети

После этого преобразованные данные подаются на слой SpatialDropout1D. Слой SpatialDropout1D с определенной вероятностью игнорирует значение наборов нейронов. Этот слой, так же как и Dropout, применяется для снижения вероятности переобучения модели.

Следующим этапом является обработка входных данных в нескольких слоях рекуррентной нейронной сети архитектуры LSTM. По завершении обработки в рекуррентных слоях данные обрабатываются в нескольких Dense слоях. Последний слой определяет, какому классу соответствуют входные данные.

Описание эксперимента. Анализ результатов

Обучение модели производилось методом обратного распространения ошибки в течение 2-х эпох. Метрикой для оценки качества обучения на каждой из эпох была выбрана precision. Данная метрика была определена, как «нечувствительная» к дисбалансу классов, который явно наблюдается в имеющемся экспериментальном наборе.

В результате обучения классификатор показал точность 98% на тестовой выборке данных. Для оценки качества полученной модели и определения количества ошибок 1-го и 2-го родов рассмотрим матрицу ошибок. Сформированная в ходе тестирования модели матрица ошибок приведена на рисунке 3.

Общее количество ошибок второго рода равно 15, в то время как ошибки первого рода отсутствуют вовсе. Однако в рассматриваемой задаче ошибки второго рода являются более критичными, поскольку при них истинно вредоносный файл классифици-

руется как безопасный.

Для оценки качества классификатора также рассмотрим полученную ROC-AUC кривую. Построенный график кривой представлен на рисунке 4.

Полученный график демонстрирует, что точность классификации модели близка к точности идеального классификатора. Таким образом, несмотря на наличие ошибок второго рода, общее число правильно определенных классов свидетельствует об успешном обучении модели.

Рис. 3. Матрица ошибок

Рис. 4. ROC-кривая

В итоге обучения была получена нейронная сеть, способная распознавать вредоносные программы с низким количеством ложноположительных и ложноотрицательных срабатываний. В дальнейшем полученную модель планируется применять в систе-

ме комплексного анализа ВПО в составе комплексной системы обеспечения безопасности ситуационного центра. Такая система позволит повысить адаптивность технологий, применяемых при анализе вредоносного программного обеспечения [7].

Вывод

1. В ходе данного исследования был проведен анализ динамических методов распознавания ВПО с использованием различных подходов. В результате анализа было определено, что анализ приложений, базирующийся на рассмотрении их API-вызовов, является одним из самых эффективных.

2. Была установлена и настроена локальная платформа для извлечения API-вызовов приложений. В результате с помощью данной платформы был сформирован набор данных, содержащий 500 файлов с отчетами о действиях безопасных приложений, и 1583 файла с отчетами о действиях ВПО.

3. В результате проведенных исследований удалось разработать модель, успешно решающую задачу распознавания вредоносного программного обеспечения. В дальнейшем данная модель будет использована как часть системы обнаружения вредоносных программ.

Список литературы:

1. Bruce Ndibanje, Ki Hwan Kim, Young Jin Kang. Cross-Method-Based Analysis and Classification of Malicious Behavior by API Calls Extraction // *Applied Sciences*. 2019. Vol. 9 (2). P. 239. DOI: 10.3390/app9020239
2. MEDUSA: metamorphic malware dynamic analysis using signature from API / V. Nair, H. Jain, Y. Golecha, M. Gaur, V. Laxmi // *Proceeding of the 3rd International Conference on Security of Information and Networks-SIN 2010*. 2010. P. 263–269. DOI: 10.1145/1854099.1854152
3. Ahmed M.E., Nepal S., Kim H. MEDUSA: Malware Detection Using Statistical Analysis of System's Behavior // *IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*. 2018. P. 272–278. DOI: 10.1109/CIC.2018.00044
4. Mingdong Tang, Quan Qian. Dynamic API call sequence visualisation for malware classification // *IET Information Security*. 2019. No. 4. P. 367–377. DOI: 10.1049/iet-ifs.2018.5268
5. Черкасов А.Н., Туркин Е.А. Разработка модели распознавания вредоносного программного обеспечения на основе сверточных нейронных сетей // *Инновационное развитие техники и технологий в промышленности: сб. материалов Всерос. науч. конф. молодых исследователей с междунар. участием (ИНТЕКС-2021)*. Москва, 2021. Ч. 4. С. 72–76.
6. Hochreiter S., Schmidhuber J. Long Short-Term Memory // *Neural Computation*. 1997. Vol. 9 (8). P. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735
7. Адаптивная система комплексного обеспечения безопасности как элемент инфраструктуры ситуационного центра / М.М. Путято, А.С. Макарян, А.Н. Черкасов, И.Г. Горин // *Прикаспийский журнал: управление и высокие технологии*. 2020. № 4 (52). С. 75–84.

References:

1. Bruce Ndibanje, Ki Hwan Kim, Young Jin Kang. Cross-Method-Based Analysis and Classification of Malicious Behavior by API Calls Extraction // *Applied Sciences*. 2019. Vol. 9 (2). P. 239. DOI: 10.3390/app9020239
2. MEDUSA: metamorphic malware dynamic analysis using signature from API / V. Nair, H. Jain, Y. Golecha, M. Gaur, V. Laxmi // *Proceeding of the 3rd International Conference on Security of Information and Networks-SIN 2010*. 2010. P. 263–269. DOI: 10.1145/1854099.1854152
3. Ahmed M.E., Nepal S., Kim H. MEDUSA: Malware Detection Using Statistical Analysis of System's Behavior // *IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*. 2018. P. 272–278. DOI: 10.1109/CIC.2018.00044

4. Mingdong Tang, Quan Qian. Dynamic API call sequence visualisation for malware classification // IET Information Security. 2019. No. 4. P. 367–377. DOI: 10.1049/iet-ifs.2018.5268

5. Cherkasov A.N., Turkin E.A. Development of a model for recognizing malicious software based on convolutional neural networks // Innovative development of engineering and technology in industry: coll. of proceedings of the Russian Scientific Conference of Young Researchers with International Participation (INTEX-2021). Moscow, 2021. Pt. 4. P. 72–76.

6. Hochreiter S., Schmidhuber J. Long Short-Term Memory // Neural Computation. 1997. Vol. 9 (8). P. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735

7. Adaptive integrated security system as an element of the infrastructure of the situation center / M.M. Putyato, A.S. Makaryan, A.N. Cherkasov, I.G. Gorin // Caspian Journal: Management and High Technologies. 2020. No. 4 (52). P. 75–84.

Авторы заявляют об отсутствии конфликта интересов.

The authors declare no conflicts of interests.

Статья поступила в редакцию 08.04.2021; одобрена после рецензирования 10.05.2021; принята к публикации 15.05.2021.

The article was submitted 08.04.2021; approved after reviewing 10.05.2021; accepted for publication 15.05.2021.

© А.Н. Черкасов, Е.А. Туркин, 2021