УДК 004.056

Д.В. Комашинский

ОБНАРУЖЕНИЕ И ИДЕНТИФИКАЦИЯ ВРЕДОНОСНЫХ ИСПОЛНЯЕМЫХ ПРОГРАММНЫХ МОДУЛЕЙ С ПОМОЩЬЮ МЕТОДОВ DATA MINING

Комашинский Д.В. **Обнаружение и идентификация вредоносных исполняемых программных модулей с помощью методов Data Mining.**
Аннотация. Исследование затрагивает проблему улучшения основных характеристик систем обнаружения и идентификации вредоносных исполняемых файлов на основе методов Data Mining. Определяется общая структура процессов построения и эксплуатации систем данного класса. На ее основе уточняется перечень нефункциональных требований к подобным системам. Задача работы определяется в виде поиска эффективных моделей представления исполняемых объектов, позволяющих получать компактные и информативные вектора описаний анализируемых объектов. Излагается суть предлагаемых подходов к обнаружению и выявлению вредоносных программ на основе статической позиционно-зависимой информации и низкоуровневых динамических признаков. Представляется архитектура разработанной системы выявления вредоносных программ и результаты практической проверки разработанных моделей представления.
Ключевые слова: разрушающие программные воздействия, анализ исполняемых файлов, интеллектуальный анализ данных.

Komashinskiy D.V. **Detecting and identifying malicious executable binaries with Data Mining methods.**
Abstract. The paper touches on the problem of improving vital characteristics of Data Mining - based systems responsible for detecting and identifying malicious executable binaries (malware). The common structure of learning and operating procedures for such systems is defined. The main non-functional requirements to the systems are specified on this structure's basis. The research's task is formulated as a look for a new, efficient representatin models for executable binaries. The models are to give compact, informative description vectors for such file objects. The essence of suggested approaches is expounded: the first one is focused on malware detection and based on positionally-dependent static data; the second uses dynamic low-level execution data for malware identification. The architecture of the developed system is represented as well as validation results for the developed representation models.
Keywords: malicious software, executable binaries analysis, data mining.

1. Введение. Концепция применения методов Data Mining (DM) для обнаружения вредоносных программ (ВП) была сформулирована Кефартом [4] и др. в 90-х годах прошлого века и получила практическое продолжение в работе Столфо, Шульца и др. [10] в 2001 году. На протяжении всего последующего времени она продолжает развиваться. Несмотря на наличие ряда ценных результатов, феноменальная изменчивость ВП продолжает ставить перед исследователями новые задачи, направленные на формирование систем его обнаружения, оптимальных в базисе требований к

характеристикам точности, производительности и ресурсопотребления. Текущее состояние дел в данной предметной области обобщается в монографии Масуда, Кхана и др. [8].

Одной из ключевых задач данной предметной области является поиск новых групп структурных и поведенческих признаков, эффективно характеризующих тот или иной аспект ВП в целом или отдельных семейств ВП в частности. Процесс их поиска требует предварительного определения того, в рамках какой модели представления анализируемых объектов производится исследование. Модель представления определяет специфику процесса извлечения начального набора признаков анализируемых объектов, что, в конечном счете, позволяет получить конечный компактный набор признаков, используемых при обучении и эксплуатации моделей принятия решения, и существенно влияет на показатели качества сформированных на базе данных моделей систем обнаружения и идентификации ВП.

2. Методы Data Mining в обнаружении ВП. Анализ потенциально опасных исполняемых объектов формата Portable Executable (PE32) является сложной задачей, требующей применения комплексного подхода к рассмотрению его отдельных характеристик. Существует две основные группы методов их анализа, обобщающие статические и динамические подходы к их обработке и принятию решения о степени их вредоносности.

Статические подходы предоставляют более быстрые и менее затратные способы анализа объектов без необходимости их выполнения в интерпретирующей среде. Примером публикаций, изучающих применимость статических признаков для выявления исполняемых файлов, являются упомянутые выше работы Кефарта и др. [4], Столфо, Шульца и др. [10], монография Масуда, Кхана и др. [8], работа Матика [9]. Все авторы подчеркивают то, что основным недостатком данной группы подходов является их неспособность эффективно решать проблему множественности статических представлений объекта, обладающего уникальным поведенческим паттерном (поведением). Это ограничивает применимость статических методов и, в итоге, объясняет недостаточную точность при использовании их в отрыве от других подходов.

Группа динамических подходов объединяет более медленные и дорогие с точки зрения используемых вычислительных ресурсов способы получения достоверной информации о функциональности анализируемого объекта, позволяющие нивелировать проблемы,

присущие статическим подходам. Например, Ланци и др. в своей работе [7] ориентируются на сбор и обработку данных о взаимодействии анализируемого объекта с окружающей средой (операционной системой) на уровне системных вызовов. Даи и др. [3] в качестве основного источника информации используют данные о потоке выполняемых приложением инструкций процессора. Вместе с тем, динамический анализ тоже обладает рядом недостатков. В первую очередь, это относится к проблеме соответствия моделируемой при анализе внешней среды (окружения) ожидаемой. Во-вторых, цена разработки и поддержки корректных моделей оказывается иногда непомерно высокой за счет трудоемкости и сложности подобной работы. Как правило, любой программный инструментарий, поддерживающий процедуры динамического анализа, имеет определенные недостатки, позволяющие вредоносному коду успешно выявлять их наличие и противостоять ему.

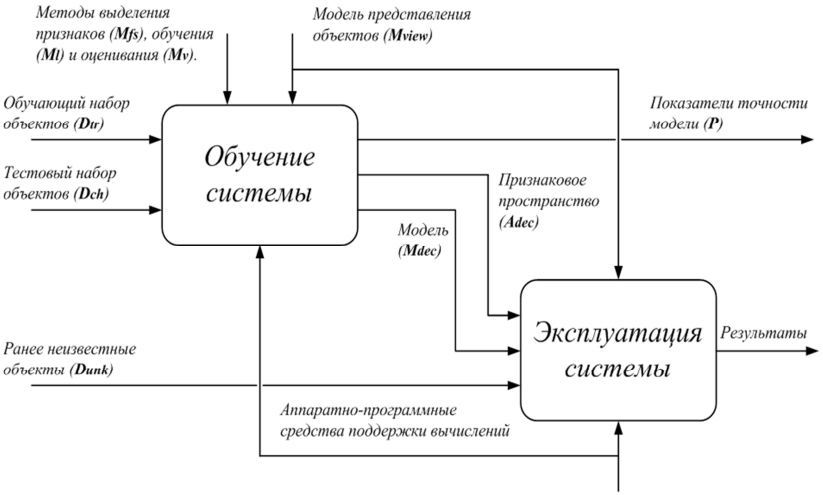

Рис. 1. Общее представление процессов обучения и эксплуатации систем выявления ВП на основе методов DM

На основе анализа [1] основных работ, посвященных теме использования методов DM для построения систем автоматического обнаружения ВП, было осуществлено моделирование основных процессов их функционирования с помощью методологии SADT. Построение моделей [2] осуществлялось с точки зрения исследователя.

На рисунке 1 представлена структура основных фаз жизненного цикла систем такого класса. Показано, что для их построения исследователь должен определить набор сущностей, устанавливающий используемые наборы данных, средства поддержки вычислений, используемую совокупность методов выделения значимых признаков, обучения и оценивания и модель представления объектов. Анализ существующих работ показывает существенную зависимость качественных показателей систем (точность принятия решения, количество ложных срабатываний, время обучения и принятия решения) от используемых моделей представления анализируемых объектов. Это определило направленность данной работы на разработку, формализацию и анализ применимости моделей представления потенциально опасных объектов формата PE32.

4. Разработанные модели представления. Современные подходы к выявлению ВП основаны на использовании сильных сторон отдельных практик, доказавших свою эффективность по отношению к тем или иным группам ВП. Это достигается за счет их комбинирования на уровне данных, групп признаков и методов обучения [8]. Этим обусловлен интерес данной работы к различным аспектам ВП. Ниже представлены разработанные модели представления исполняемых объектов на основе их структурных (статических) и динамических особенностей [5,6].

Статической моделью приложения является набор M_{PE32S}, включающий множество символов алфавита значений A, множество признаков F, радиус выделения значений r и преобразование T: $M_{PE32S} = (A, F, r, T), r \in \mathbb{N}$.

Алфавит $A = \{a_1, \dots, a_k\}$, где $k = |A|$ – мощность алфавита, и $a_i \in A, 1 \le i \le k$.

Множество признаков $F = \{f_1, \dots, f_k\}$, где $k = |F|$ – мощность множества признаков и $f_i \in F, 1 \le i \le k$.

Радиус выделения признаков r, определяющий множество значений смещений $R \in \mathbb{Z}$, как набор целых чисел в интервале $[-r, +r]$.

Преобразование $T: \langle a, i \rangle \to f$, $f_i \in F, a \in A, i \in R$, ставящее в однозначное соответствие каждой паре $\langle a, i \rangle$ где $a \in A, -r \le i \le r$, элемент множества признаков F.

Представляемая позиционно-зависимая модель позволяет частично нивелировать указанные недостатки за счет введения привязки позиций отдельных символов и их последовательностей относительно центра некоторой области анализируемого бинарного

потока. Основными структурными элементами формата PE32, необходимыми для проведения процедуры извлечения признаков, являются значение смещения точки входа опционального заголовка анализируемого объекта и объект секции, содержащий ее. Основанный на предложенной модели представления подход использует регион смещений относительно точки в хода [-127, 127] и использует в качестве величин байтовые значения по данным смещениям. Таким образом, максимально возможное количество признаков не превышает 2^{16}.

Динамической моделью приложения является набор M_{PE32D}, включающий набор P потоков выполнения T, множество символов алфавита A и подмножество терминальных символов A^T, $A^T \in A$:$M_{PE32D} = (P, A, A^T), A^T \in A$.

Алфавит $A = \{a_1, \dots, a_k\}$, где $k = |A|$ – мощность алфавита, и $a_i \in A, 1 \le i \le k$.

Поток выполнения $T = (x_1, \dots, x_n)$, $x_j \in A, 1 \le j \le n$, как упорядоченный конечный набор символов алфавита A.

Приложение $P = \{t_1, t_2, \dots, t_m\}$ представлено как конечное множество потоков выполнения.

Множество терминальных символов A^T, как подмножество $\{a_1^T, \dots, a_t^T\}$ символов алфавита A, $a_l^T \in A, 1 \le l \le t$: $A^T \subset A$.

Терминальная цепочка символов C как упорядоченный конечный набор символов алфавита A длины l, $C = (c_1, c_2, \dots, c_l)$, где только первый и последний символы принадлежат множеству терминальных символов A^T:

$$chain(\mathrm{C}) = \begin{cases} c_1 \in A^T \\ c_l \in A^T \\ (\forall i, 2 \le i \le l-1)(c_i \in A, c_i \notin A^T) \end{cases}$$

Тогда поток выполнения T может быть представлен как последовательность терминальных цепочек символов: $T = (C_1, C_2, \dots, C_l), (\forall i)(chain(C_i))$.

На практике предложенная модель используется для формального описания анализируемого объекта (программного приложения) как набора непрерывных цепочек инструкций, ограниченных инструкциями управления переходами (условного и безусловного перехода). Набора получаемых признаков (цепочек) не включает цепочки инструкций, относящихся к импортируемым библиотекам и средствам загрузки (для случая динамического анализа библиотечных файлов).

Предлагаемый на основе данной модели представления подход использует предложенную модель анализируемого объекта со следующими параметрами: (1) размер набора потоков исполнения P равен единице: $m = 1$. В рамках прикладной области это допущение обосновано тем, что любое анализируемое приложение имеет один первоначальный поток, начинающий выполнение анализируемого объекта (приложения) от точки входа; (2) состав множества алфавита символов определяется набором инструкций процессора, под управлением которого будет выполняться анализируемый объект (приложение). Для сокращения размера алфавита было принято решение ограничить размер символа двумя байтами, таким образом, $k \leq 2^{16}$; (3) множество терминальных символов A^T определяется множеством инструкций процессора, реализующих выполнение условной и безусловной передачи управления.

5. Предлагаемая архитектура системы. Для обеспечения вычислительной поддержки исследований используется программный пакет RapidMiner. Основными преимуществами данной вычислительной среды являются (1) простота и наглядность проведения экспериментов, (2) открытость ее реализации и наличие документации, обеспечивающей разработку приложений, использующих его возможности, (3) поддержка основных алгоритмов DM, и (4) полная совместимось с традиционно используемой для задач данного класса среды WEKA. На основе данной программной среды подготовлен набор схем экспериментов, для организации фаз обучения и эксплуатации систем обнаружения ВП. В качестве основных методов обучения решающих моделей применяются методы Naive Bayes, Decision Tree (C4.5), Decision Table. При работе с методами комбинирования решающих моделей используется метод RandomForest. Для оценивания точности получаемых моделей принятия решения применяется метод десятикратной кросс-валидации.

В методике обнаружения ВП на базе статической позиционно-зависимой модели объектов формата PE32 процесс извлечения признаков основан на использовании программного средства разбора файлов данного формата (т.н. парсера). Данное программное средство способно идентифицировать (1) программную точку входа анализируемого объекта, (2) непрерывный физический участок (секцию) объекта, включающий точку входа и (3) обеспечивать операцию чтения идентифицированной секции по допустимому региону относительных виртуальных адресов. Процедура извлечения

признаков в методике обнаружения ВП на основе низкоуровневой динамической информации использует средства динамического исследования и отладки модулей программных приложений. Данные средства способны произвести (1) предварительную загрузку и инициализацию анализируемого объекта, (2) фиксацию факта начала его выполнения (выполнение первой инструкции, расположенной по адресу программной точки входа) и (3) дальнейшее выполнение трассировки модуля.

6. Оценка эффективности. Практические работы по проверке методики обнаружения ВП, основанной на статической позиционно-зависимой модели представления, показали, что показатель точности обнаружения AUC (площадь ROC-кривой) достигает значения 0.98 при использовании классификатора RandomForest, обученного на пространстве из 250 признаков [6]. Результаты сравнимы с результатами оценивания существующих быстрых статических методик обнаружения ВП, основанных на использовании n-грамм и простых подходах анализа кода (дизассемблирования). Показатели точности могут быть улучшены при дальнейшем расширении пространства признаков. С точки зрения характеристик времени обучения (принятия решения) и ресурсопотребления данный подход выгодно отличается в лучшую сторону за счет ограничения потенциально возможного количества признаков, определяемых комбинацией значения и его смещения.

Работа с методикой на основе представленной динамической модели представления [5] подтвердила широкую вариативность низкоуровневой реализации ВП. Для отдельных семейств ВП ее применение дает общий показатель точности обнаружения F-Measure вплоть до 0.8 (методы дерева решений C4.5 и наивный Байесовский классификатор при количестве признаков, равном 500) при практически полном отсутствии ложных срабатываний. Это показывает, что отдельные подмножества экземпляров ВП, относящихся к одному семейству, имеют высокую степень схожести на низком уровне, что в первую очередь объясняется применением алгоритмически схожих средств защиты и обфускации вредоносного кода. Подтверждено существование ряда семейств ВП, в которые заложены механизмы противодействия анализу и обнаружению подозрительных низкоуровневых паттернов за счет полиморфизма и активного противодействия инструментальным средствам динамического анализа. В отличие от схожих низкоуровневых динамических подходов к обнаружению ВП, предлагаемый подход

ориентирован на использование групп признаков, представляющих непрерывные цепочки команд процессора, ограниченные командами условного и безусловного перехода, расположенные в адресном пространстве анализируемого исполняемого модуля. Он используется для построения систем идентификации ВП, ориентированных на поиск отдельных семейств ВП, имеющих алгоритмически схожую низкоуровневую функциональность и созданных с использованием единого набора средств компиляции, компоновки и (или) программной защиты. Предложенный подход ориентируется на анализ исполняемых объектов на начальных фазах их функционирования, что ограничивает необходимое время анализа по сравнению с другими подходами.

7. Заключение. Разработана модель статического представления исполняемых объектов формата PE32. На ее основе разработана методика статического обнаружения исполняемого ВП на основе позиционно-зависимых признаков. Обоснована значимость использования информации в регионе точки входа (Entry Point) исполняемых файлов для процессов принятия решения о степени его опасности. С практической точки зрения предложенный подход может быть использован для построения систем быстрой идентификации верхнего слоя программных средств компиляции, компоновки, упаковки и программной защиты, используемых при создании ВП.

Разработана модель динамического представления исполняемых объектов формата Portable Executable. На ее основе разработан подход к идентификации исполняемого ВП на основе динамических низкоуровневых признаков. Обоснована значимость использования факта наличия отдельных непрерывных последовательностей машинных инструкций в коде анализируемого объекта для принятия решения о степени его вредоносности. На практике данный подход может применяться как для идентификации отдельных алгоритмических конструкций, свойственных ВП, так и для выявления факта применения средств упаковки, обфускации и программной защиты, используемых при его создании.

Литература

1. *Комашинский Д.В., Котенко И.В.* Концептуальные основы использования методов Data Mining для обнаружения вредоносного программного обеспечения // Защита информации. Инсайд, 2010. № 2, С.74-82.
2. *Комашинский Д.В.* Особенности задачи применения Data Mining для обнаружения разрушающих программных воздействий. // Сборник «Инновации в науке»: материалы XVI международной заочной научно-практической конференции. №1, Новосибирск: Изд. «СибАК», 2013. С.74-78.

3. *Dai J., Guha R., Lee J.* Efficient Virus Detection Using Dynamic Instruction Sequences. // Journal of Computers, Vol. 4, No. 5, P. 405-414, 2009.
4. *Kephart J.O., Sorkin G.B., Arnold W.C., Chess D.M., Tesauro G.J., White S.R.* Biologically inspired defenses against computer viruses // Proceedings of 14th International Joint Conference on Artificial Intelligence, 1995, P. 985–996.
5. *Komashinskiy D.V, Kotenko I.V.* Using Low-Level Dynamic Attributes for Malware Detection Based on Data Mining Methods. // Proceedings of the 6th International Conference on Mathematical Methods, Models and Architectures for Computer Network Security, Saint-Petersburg, 2012, P. 254-269.
6. *Komashinskiy D.V., Kotenko I.V.* Malware Detection by Data Mining Techniques Based on Positionally Dependent Features. // Proceedings of the 18th Euromicro International Conference on Parallel, Distributed and network-based Processing. Los Alamitos, California. IEEE Computer Society. 2010. P.617-623.
7. *Lanzi A., Balzarotti D., Kruegel C., Christodorescu M., Kirda E.* AccessMiner: Using System-Centric Models for Malware Protection // Proceedings of 17th ACM conference on Computer and Communication Security, 2010. P. 399-412.
8. *Masud M.M.,Khan L., Thuraisingham B.* Data Mining Tools for Malware Detection. CRC Press Taylor & Francis Group, 2012.
9. *Muttik I.* Malware Mining // Proceedings of 21st Virus Bulletin Conference, 2011, P. 46-51.
10. *Schultz M., Eskin E., Zadok E., Stolfo S.* Data Mining Methods for Detection of New Malicious Executables // Proceedings of the IEEE Symposium on Security and Privacy, 2001 P. 38-49.

Комашинский Дмитрий Владимирович — аспирант лаборатории проблем компьютерной безопасности СПИИРАН. Область научных интересов: обнаружение и анализ вредоносных программ, машинное обучение. Число научных публикаций — 27. komashinskiy@comsec.spb.ru, http://comsec.spb.ru/ru/staff/komashinskiy; СПИИРАН, 14 линия, 39, Санкт-Петербург, 199178, РФ; р.т. +7(812)328-2642, факс +7(812)328-4450. Научный руководитель — И.В. Котенко.

Komashinskiy Dmitriy Vladimirovich — postgraduate student, Laboratory of Computer Security Problems, SPIIRAS. Research interests: intrusion detection and analysis, machine learning. The number of publications — 27. komashinskiy@comsec.spb.ru, http://comsec.spb.ru/ru/staff/komashinskiy; SPIIRAS, 14-th line, 39, St. Petersburg, 199178, Russia; office phone +7(812) 328–2642, fax +7(812)328–4450. Scientific advisor — I.V. Kotenko.

Поддержка исследований. В публикации представлены результаты исследований, поддержанные Министерством образования и науки Российской Федерации (государственный контракт 11.519.11.4008), грантами РФФИ, программой фундаментальных исследований ОНИТ РАН и проектами Седьмой рамочной программы Европейского Союза SecFutur и MASSIF.

Рекомендовано лабораторией Проблем компьютерной безопасности СПИИРАН, заведующий лабораторией Котенко И.В., д-р техн. наук, проф.
Статья поступила в редакцию 10.03.2013.

РЕФЕРАТ

Комашинский Д.В. **Обнаружение и идентификация вредоносных исполняемых программных модулей с помощью методов Data Mining.**

Исследование направлено на улучшение подходов к обнаружению и идентификации новых, ранее неизвестных вредоносных исполняемых программных модулей (далее вредоносных программ, ВП) на основе методов Data Mining (DM).

Анализ существующих подходов к решению этой задачи позволяет формализовать общую структуру работ по созданию и эксплуатации систем автоматического выявления ВП и выделить набор элементов, определяющих суть данных подходов. Определяется перечень основных нефункциональных требований, включающий время обучения системы, время принятия решения и его точность.

Разработана методика статического обнаружения ВП, позволяющая объединить в рамках обобщенного пространства признаков понятия N-граммы и ее расположения в интересующем участке анализируемого бинарного объекта. В отличие от существующих традиционных подходов к извлечению низкоуровневых элементов потока данных, подход позволяет нивелировать проблему потенциально большого объема начального пространства признаков за счет ограничения размера анализируемого участка потока с одновременным снижением размерности N-грамм при сохранении точности обнаружения.

Разработана методика динамического обнаружения ВП. Она развивает идеи статического анализа бинарного кода приложений за счет ввода процедуры динамического сбора информации о выполняемых приложением инструкциях процессора. В отличие от традиционных динамических подходов к формированию поведенческого профиля приложений за счет анализа его взаимодействия с внешней средой, данная методика ориентируется на сбор данных о внутренней логике стартового кода исполняемых объектов с их последующим обобщением в виде набора уникальных непрерывных последовательностей инструкций.

Для обеспечения вычислительной поддержки исследований используется программный пакет RapidMiner. Его основными преимуществами являются поддержка графической среды разработки, обеспечивающей простоту и наглядность проведения экспериментов; открытость ее реализации, позволяющая производить необходимые расширения функциональности; наличие документированных интерфейсов взаимодействия с средой, обеспечивающее возможность разработки клиентских приложений, использующих ее возможности; поддержка большого количества базовых алгоритмов DM, включая полную интеграцию традиционно используемой для задач данного класса среды WEKA (Waikato Environment for Knowledge Analysis).

SUMMARY

Komashinskiy D.V. **Detecting and identifying malicious executable binaries with Data Mining methods.**

The research is focused on improving Data Mining (DM) – based approaches for detecting and identifying new, previously unseen malicious binary executables.

The analysis of existing ways of preparing such systems gives an opportunity to formalize generic structure of the systems' implementation workflow and select a set of mandatory elements necessary for their instantiation. A set of main non-functional requirements is specified (learning time, decision making time, accuracy).

The static approach to detect malware is developed. It allows to combine N-grams' values with their positions in a part of analyzed binary stream. In contrast to existing traditional N-gram based approaches it minimizes resoures necessary for feature space preparation by restricting size of analyzed stream's block with simultaneous decreasing on N-grams size. At the same time it shows comparable accuracy results.

The dynamic approach to detect malware is developed. It expands ideas of static analysis of binary code af application by introducting a procedure of dynamic harvesting of data on executed machine instructions. Unlike traditional dynamic approaches to form behaviour profiles of investigated applications based on loggig their communication transictions with environment (operating system) it focuses on keeping data about internal logic of startup routines with their further generalization into a set of features representing machine code instruction sequences.

In order to supply the research with necessary algorithmic base RapidMiner software is adopted. It provides following advantages: support of WYSIWYG development paradigm; possibility to implement own client applications and contribute new functionality; support of main well-known learning algorithms; compatibility with traditionally popular and admittedly useful for such research tasks Waikato Environment for Knowledge Analysis (WEKA).