ПОДХОД К ОБНАРУЖЕНИЮ ВРЕДОНОСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА ОСНОВЕ ПОЗИЦИОННО-ЗАВИСИМОЙ ИНФОРМАЦИИ

КОМАШИНСКИЙ Д.В., КОТЕНКО И.В., ШОРОВ А.В.

УДК 004.49

Комашинский Д.В., Котенко И.В., Шоров А.В. **Подход к обнаружению вредоносного программного обеспечения на основе позиционно-зависимой информации.**

Аннотация. Проблема противодействия вредоносному программному обеспечению (ПО), остается довольно острой, несмотря на появление более эффективных механизмов его выявления, анализа, обновления баз его описаний и правил обнаружения. Важным аспектом этой проблемы является поиск эвристических методов детектирования, обладающих большей точностью обнаружения. В работе рассматривается применение методов интеллектуального анализа данных (Data Mining) для создания эвристических детекторов вредоносного ПО. Описываемый подход отличается от существующих направленностью на обработку статической информации, обеспечивающей формирование отдельных функциональных элементов эффективной модели детектирования вредоносных исполняемых объектов. В работе реализована и исследована общая методология формирования системы детектирования на базе применения методов выделения значимых признаков и методов классификации.

Ключевые слова: защита информации, вредоносное ПО, обнаружение вредоносного ПО, интеллектуальный анализ данных, методы статического анализа.

Komashinskiy D.V., Kotenko I.V., Shorov A.V. **Approach to detect malware based on postionally dependent information.**

Abstract. The problem of counteraction to malware remains quite severe, despite the emergence of more effective mechanisms for its identification, analysis, updating the database of its descriptions and detection rules. An important aspect of this problem is to find heuristic detection methods with better accuracy. This paper considers the application of data mining methods to create heuristic malware detectors. The described approach differs from existing by focusing on processing of static information, ensuring the formation of particular functional elements of an effective model to detect malicious executables. We implemented and studied a general methodology for creating detection system based on application of methods for selecting significant features and classification.

Keywords: information security, malware, detection of malicious software, data mining, methods of static analysis.

1. Введение. Одной из реально существующих проблем компьютерной безопасности является обнаружение вредоносного ПО (malisious software, malware). Выделяют два подхода для решения задачи защиты от вредоносного ПО:

1) принятие решений на основе данных, которые могут быть получены без выполнения анализируемой компьютерной программы (так называемая группа методов статического анализа);

2) принятие решений на основе данных, полученных при выполнении компьютерной программы (группа методов динамического анализа или обработки динамической или поведенческой информации).

Актуальность задачи обнаружения вредоносного ПО обусловливается текущими тенденциями развития технологий, применяемых в нем (таких как гибкая система управления, поэтапная схема инфицирования, модульность обновлений, наличие эффективных механизмов сокрытия своего наличия на пораженном хосте, использование активных механизмов противодействия средствам антивирусной защиты), а также недостаточностью методов статического анализа потенциальных контейнеров вредоносного кода.

Эффективному обнаружению вредоносного ПО в его активной фазе (т. е. фазе выполнения вредоносных функций) способствует то, что в момент своего выполнения приложение должно реализовать свои основные вредоносные функции. Вместе с тем следует учитывать существование проблем, вызываемых возможностью реализации поведенческого полиморфизма во вредоносных программах, которые схожи с проблемами детектирования при использовании структурной информации (например, при применении протекторов и упаковщиков для файлов формата PE32).

Существует большое количество релевантных исследований в области обнаружения вредоносного ПО. Наиболее показательными из них являются работы [1, 3, 5–7]. В работе [1] рассматривается подход, реализующий статический анализ исполняемых файлов. Публикация [2] посвящена проблеме обнаружения новых, ранее неизвестных вредоносных программ в формате Portable Executable. В работе [3] авторы в качестве признаков вредоносного ПО использовали 4-байтовые последовательности (так называемые n-граммы). В работе [4] выделение наиболее значимых последовательностей (признаков) производилось за счет вычисления для каждой из них значения информационного усиления и формирования списка 500 признаков с максимальными значениями. Работа [5] посвящена вопросам построения модели детектирования, использующей данные, полученные путем статического анализа исполняемых файлов формата PE32. В работе [6] использовался подход, базирующийся на более сложной технике анализа исполняемых файлов с привлечением отладочных средств. В работе [7] исследовалась применимость метода SVM для построения модели детектирования приложений на основе собираемой при их выполнении поведенческой информации.

Описываемый в настоящей работе подход отличается от существующих своей направленностью на обработку позиционно-зависимой статической информации, обеспечивающую формирование отдельных элементов эффективной модели детектирования вредоносных исполняемых объектов, которые в дальнейшем могут быть использованы совместно за счет комбинирования с уже существующими методами, отдельно для выполнения задач, решаемых при уточнении общего контекста задачи анализа объекта. В работе на основе результатов проведенных экспериментов делаются заключения, которые подлежат апробации в рамках дальнейших исследований по комплексному использованию методов статического и динамического анализа.

Структура статьи следующая: первый раздел вводный; во втором разделе дана краткая характеристика решаемой задачи детектирования, определены требования к ней и вытекающие особенности использования методов Data Mining; в третьем разделе рассмотрен подход, примененный для проведения исследований на текущем этапе работы; в четвертом разделе уточняются данные, использованные при проведении экспериментов, аспекты реализации программного комплекса моделирования, а также представлены проведенные эксперименты и полученные результаты. В заключении подводятся итоги проведенных исследований и определяются направления дальнейших исследований.

2. Особенности задачи обнаружения вредоносного ПО. При анализе релевантных работ выявлено, что хотя точность методов динамической обработки потенциально больше, методы статического и динамического анализа развиваются параллельно. Исторически сложилось, что группа методов статического анализа, или детектирования, возникла первой, что было обусловлено изначально небольшим количеством типов вредоносного ПО и отсутствием проработанных методик защиты от него. Простейшим средством статического детектирования является сигнатурное сканирование, которое позволяет быстро и точно идентифицировать опасность и определить последовательность действий по ликвидации угрозы и ее последствий. В дальнейшем с ростом количества вовлеченных в сферу компьютерных технологий пользователей и развитием программных средств стали появляться развитые методы обфускации участков программ, по которым можно произвести детектирование, упаковку и полиморфное преобразование кода. Эта проблема вызвала необходимость формирования технологически более сложных решений, опирающихся на частичное или полное выполнение анализируемой программы в защищенной среде (виртуальные машины, «песочницы» и т. д.).

Несмотря на преимущество методов динамического анализа по эффективности детектирования современного вредоносного ПО (по сравнению с методами статического анализа), эти методы имеют следующие недостатки: 1) низкую скорость принятия решений; 2) высокий уровень потребления ресурсов; 3) наличие специфических приемов, позволяющих вводить в заблуждение о функциональности объекта (например, передача управления на «ложную» ветвь выполнения при обнаружении защищенной среды, выполнение вредоносного кода по определенным условиям, в том числе по событию обновления и т. д.).

Текущее состояние дел показывает, что, возможно, усилия, прилагаемые для совершенствования методов динамического анализа, скоро достигнут своего технологического порога. Это объясняется наличием эффективных антиэмуляционных технологий, а также наблюдаемыми трендами в изменениях подходов, используемых для реализации основных фаз жизненного цикла вредоносных программ.

Таким образом, одним из перспективных направлений дальнейшего развития технологий детектирования вредоносного программного обеспечения является комплексное использование указанных групп методов. При построении оптимального (или рационального) механизма принятия решений о применении тех или иных механизмов это позволит объединить преимущества и нивелировать недостатки каждой из них. Данная задача может и должна быть рассмотрена с точки зрения применимости методов Data Mining для формирования и отдельных, и связанных блоков принятия решений. Представленная работа рассматривает способ построения фрагмента такой системы принятия решений на фазе начальной обработки доступной статической информации.

3. Сущность предлагаемого подхода. Данный подход рассматривается как элемент общей системы принятия решений о степени вредоносности анализируемых объектов (компьютерных программ).

Как было показано ранее, исследования методов статического анализа в основном базировались на выделении в качестве признаков так называемых n-грамм (строк, байтовых блоков) или байтовых последовательностей. При наличии необходимых вычислительных мощностей можно получить информацию о том, какие из n-грамм специфичны для целевых классов программ.

В отличие от использования поиска бинарных последовательностей, ограниченных по длине, идея подхода, предлагаемого в настоящей работе, заключается в *использовании особенностей формата*

файлов, которые могут включать в себя вредоносный код. Очевидно, что знание характера и структуры информации, включаемой в потенциально опасный объект, позволяет существенно сузить область данных, которые подлежат анализу в первую очередь. К примеру, знание того, что файлы формата OLE Structured Storage, используемые для хранения данных приложениями Microsoft Office, могут содержать отдельные маркированные контейнеры с вредоносным кодом, позволяет исключить из фокуса рассмотрения все остальные данные.

Для реализации наших задач использовалась база вредоносных файлов [8], чей контент был предварительно отфильтрован с целью выделить из общего доступного набора файлы формата PE32 [2], применяемого в качестве основного формата исполняемых файлов для 32-битных версий операционной системы Microsoft Windows. Коллекция безопасных файлов собрана на эталонной конфигурации, включающей в себя установленную операционную систему Microsoft Windows XP и ряд установленных сопутствующих приложений (всего 1656 файлов).

Для функционирования базовых классификаторов применялись методы Decision Table [9], C4.5 [10], RandomForest [11] и Naive Bayes [12]. В качестве признаков для обучения выбраны связки величин «Позиция—Значение».

На данном этапе исследований для упрощения требований к вычислительной мощности испытательного стенда использовались следующие допущения:

— важным для результатов детектирования вредоносных файлов формата PE32 является его контент в области точки входа, т. е. в так называемой OriginalEntryPoint (OEP), чей относительный виртуальный адрес указывается в структуре OptionalHeader набора заголовков NTHeaders;

— рассматривается область в регионе смещений [–127, 127] относительно OEP (т. е. значение байта по адресу OEP находится по нулевому смещению); таким образом, допустимое значение величины «Позиция» находится в пределах заданного выше интервала;

— по каждому адресу, определяемому значением величины «Позиция», сохраняется значение байта. Таким образом, значение величины «Позиция» находится в пределах [0, 255].

Результатом данных допущений стало начальное ограничение количества возможных признаков, используемых в дальнейшем для обучения классификаторов, в рамках возможного количества комбинаций для 2-байтовой величины (65536).

Кроме того, указанные допущения позволили уйти от искажений, связанных с необходимостью решения задачи дискретизации числовых данных [13], так как можно воспользоваться булевыми значениями true, false для определения наличия в каждом анализируемом объекте того или иного признака.

Таким образом, в проводимом эксперименте значение признака определяется 2-байтовой величиной, значение первого байта которой задает значение позиции, а второго — значение байта в данной позиции (см.таблицу).

Примеры значений признаков (связок величин «Позиция—Значение»)

Значение признака	Позиция	Значение байта в позиции
49917	–3	194
3326	–2	12
33660	124	131

Основным достоинством данного подхода к формированию пространства признаков является его сходство с методами формирования сигнатур, что, к примеру, позволяет предполагать потенциально высокую эффективность на данном пространстве признаков методов, основанных на деревьях решений [10, 11].

Недостатками данного подхода являются высокий уровень ошибок при обработке заведомо искаженных файлов, защищенных от статического анализа, и то, что в случаях, когда исполняемые файлы генерируются с использованием высокоуровневых сред разработки, их OEP указывает на код инициализации среды выполнения (runtime stub). Последняя упомянутая проблема на настоящий момент полностью не решается, и, в принципе, может быть решена за счет введения эвристик, позволяющих определять смещение на реальную, определенную пользователем точку входа.

Кроме того, следует учесть потенциальную возможность выхода за пределы секции кода при чтении значений байтов по позициям. В данном случае при обнаружении выхода за пределы адресного региона, в котором располагается секция, включающая значение, располагаемое по адресу OEP, ставится нулевое значение по данной позиции, что позволяет частично нивелировать данную проблему. Более того, необходимо учитывать и возможность присутствия излишней информации в поле «Значение байта в позиции», вызванную наличием данных адресации (операндов). Эта проблема решается за счет большого

количества данных, используемых при обучении, т. е. классификаторы используют при обучении наиболее часто встречающиеся значимые значения в позициях, соответствующих опкодам инструкций и стабильной (специфичной для вредоносных программ) информации, указывающей данные адресации для инструкций.

4. Проведенные эксперименты. Для проведения экспериментов выделены две базы исполняемых файлов, содержащие 5854 опасных файлов и 1656 неопасных соответственно.

Для извлечения наборов признаков разработана утилита разбора файлов формата PE32, ориентированная на доступ к контенту файла по относительным виртуальным адресам. Данная утилита генерировала на выходе файлы формата Attribute-Relation (ARFF), включающие в себя наборы всех возможных признаков.

Далее полученные данные переданы процедуре выделения значимых признаков, обеспечивающей формирование приоритезированного списка признаков, которая реализована на основе вычисления показателя информационного усиления.

Из полученного приоритезированного списка ~~были~~ извлечены подмножества наиболее значимых признаков размером 50, 100, 150, 200, 250.

Затем произведена повторная процедура генерации ARFF файлов, содержащих показатели значимых признаков, входящих в данные наборы для используемой базы исполняемых файлов.

Для оценки качества обученных классификаторов применялась процедура 10-кратной кросс-проверки.

Для осуществления эксперимента использовался программный пакет Weka 3.6.1.

При анализе выделенных значимых признаков исследованы зависимости значения информационного усиления признаков (далее — коэффициента InfoGain) от их базовых составляющих (смещения относительно точки входа в исполняемый файл и величины, располагаемой по нему).

Как было показано, наиболее значимый контент, анализ которого в значительной степени влияет на показатели точности классификаторов, располагается в верхних областях виртуальной памяти относительно точки входа.

Существует компактная группа значимых смещений (рис. 1), контент которых должен быть проверен в первую очередь. Другими словами, при реализации типового антивирусного сканера фокус должен быть сделан на поиске уникальных, свойственных программе байто-

вых последовательностях за точкой входа в программу (OEP). Вместе с тем следует уделить внимание тому факту, что существует и менее компактная и значимая группа признаков, соответствующих отрицательным смещениям. Максимально важные для анализа данные располагаются вблизи OEP в интервале смещений [–5, 20].

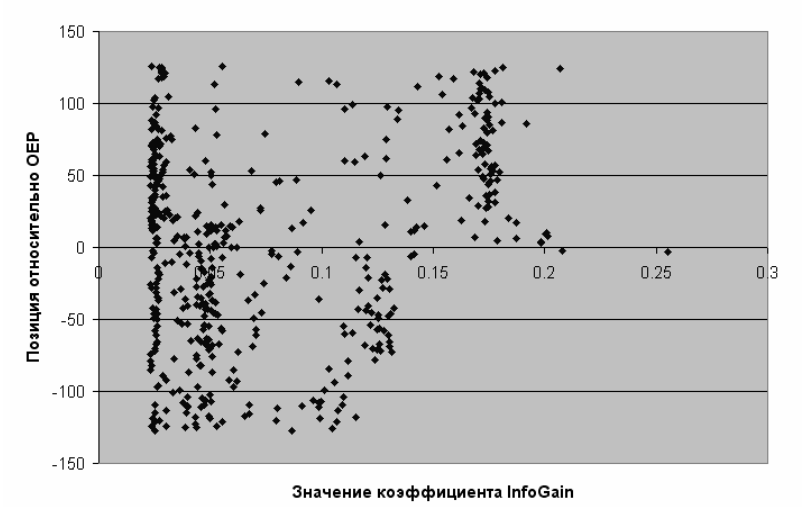


Рис. 1. Значимость позиций байтов, размещенных по определенным смещениям относительно OEP.

Графический анализ распределения данных значимых величин показывает, что существуют определенные данные, которые чаще других используются в значимых признаках, что демонстрируется довольно четко выраженными вертикальными линиями (рис. 2).

Данная тенденция может быть объяснена наличием специфичных для разных классов программ инструкций, формирующих стабильные бинарные участки в выбранном для исследования регионе (опкоды и операнды).

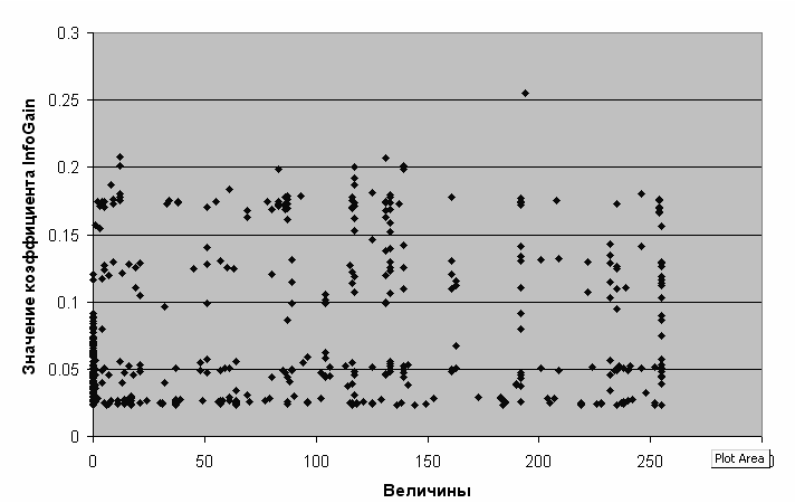


Рис. 2. Значимость определенных показателей байтов.

Как отмечено ранее, специфика алгоритмов классификации, основанных на построении деревьев решений (Decision Tree), позволяет получить интуитивно понятный набор правил, позволяющий при минимальных усилиях сформировать соответствующий им программный код, что может быть полезно при автоматизации правил детектирования сигнатур (рис. 3).

Результаты проведенных при оценке точности расчетов показали, что в большинстве случаев при увеличении количества признаков, описывающих исполняемые файлы, растет точность прогностической функции используемых классификаторов.

Наиболее эффективным показал себя метод RandomForest, чье преимущество по сравнению с другими классификаторами можно объяснить примененной в нем идеей с использованием ансамбля деревьев решений, обученных на отдельных фрагментах обучающей выборки и пространства признаков.

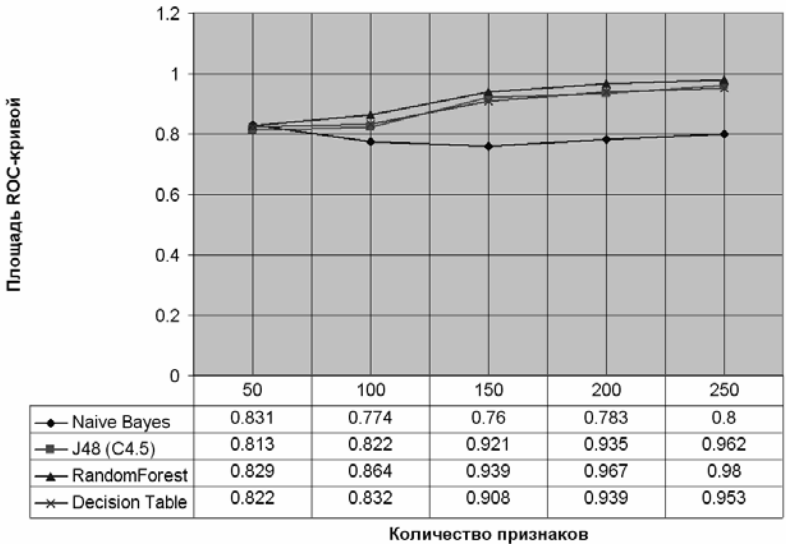
```
J48 pruned tree
------------------

sp_49917 = y: benign (1076.0/101.0)
sp_49917 = n
|   sp_29961 = y: benign (44.0/1.0)
|   sp_29961 = n
|   |   sp_65105 = y: benign (9.0/2.0)
|   |   sp_65105 = n
|   |   |   sp_29988 = y
|   |   |   |   sp_41254 = y: benign (9.0)
|   |   |   |   sp_41254 = n: malware (10.0/4.0)
|   |   |   sp_29988 = n
|   |   |   |   sp_22073 = y: benign (4.0/1.0)
|   |   |   |   sp_22073 = n
|   |   |   |   |   sp_29727 = y
|   |   |   |   |   |   sp_35592 = y: malware (2.0)
|   |   |   |   |   |   sp_35592 = n: benign (34.0/11.0)
|   |   |   |   |   sp_29727 = n
|   |   |   |   |   |   sp_22319 = y: benign (10.0/4.0)
|   |   |   |   |   |   sp_22319 = n
|   |   |   |   |   |   |   sp_3159 = y: benign (12.0/5.0)
|   |   |   |   |   |   |   sp_3159 = n
|   |   |   |   |   |   |   |   sp_33660 = y
|   |   |   |   |   |   |   |   |   sp_34136 = y: malware (4.0)
|   |   |   |   |   |   |   |   |   sp_34136 = n: benign (52.0/24.0)
|   |   |   |   |   |   |   |   sp_33660 = n
|   |   |   |   |   |   |   |   |   sp_3082 = y
|   |   |   |   |   |   |   |   |   |   sp_23813 = y: malware (3.0)
|   |   |   |   |   |   |   |   |   |   sp_23813 = n: benign (6.0/1.0)
|   |   |   |   |   |   |   |   |   sp_3082 = n: malware (6235.0/546.0)

Number of Leaves  :     15

Size of the tree :      29
```

Рис. 3. Структура дерева решений, построенная алгоритмом C4.5 на обучающем наборе с 50 признаками.

Он оказался максимально эффективным на всех используемых наборах признаков (рис. 4).

Для классификаторов C4.5, RandomForest и Decision Table показатели площади ROC-кривой, рассматриваемой как интегральный показатель точности предсказательной функции классификаторов, больше 0.8 для всех использованных наборов признаков, что в общем случае соответствует хорошему качеству классификации.

Однако следует отметить, что данные показатели не являются приемлемыми для обеспечения требуемого качества детектирования вредоносного программного обеспечения, функционирующего на пользовательских хостах.



	50	100	150	200	250
Naive Bayes	0.831	0.774	0.76	0.783	0.8
J48 (C4.5)	0.813	0.822	0.921	0.935	0.962
RandomForest	0.829	0.864	0.939	0.967	0.98
Decision Table	0.822	0.832	0.908	0.939	0.953

Рис. 4. Результаты оценки примененных классификаторов на наборах с различным количеством признаков.

Модели детектирования вредоносного ПО, реализованные на основе описанного подхода, являются полезными для предварительного быстрого анализа и могут быть использованы в комплексе с другими средствами эвристического анализа.

5. Заключение. Проведенное исследование показало, что применение позиционно-зависимых признаков, извлекаемых на этапе статического анализа исполняемых файлов, ~~является~~ достаточно эффективно при использовании методов Data Mining, относящихся к группам классификаторов, использующих генерацию правил и построение деревьев решений. Наиболее эффективным показал себя метод RandomForest, интегрирующий в себя общие принципы улучшения качества классификации за счет реализации принципов обобщения результатов работы нескольких сущностей, ответственных за принятие решения.

Показана и обоснована необходимость учета значимости отдельных регионов анализируемых объектов и данных, находящихся в них. Данный подход не гарантирует абсолютной точности детектирования вредоносного программного обеспечения, но, в силу показанных осо-

бенностей, может быть эффективен на определенных этапах принятия решения о способе дальнейшей обработки объекта и при построении средств детектирования отдельных семейств исполняемых программ. Очевидным примером может быть задача автоматизации обнаружения и идентификации использованных средств обфускации или защиты исполняемых файлов, что позволяет обеспечить четкую автоматическую процедуру генерации правил детектирования и тем самым предоставляет возможность более корректно определить путь дальнейшего анализа объекта (каким образом настроить средства динамического анализа, каким участкам исследуемого объекта следует уделить внимание в дальнейшем при подтверждении факта его обфускации и т. д.).

Это обусловливает необходимость использования иерархических моделей принятия решения при использовании эвристических методов анализа в целом и методов Data Mining в частности. Ближайшими задачами в данном направлении являются: 1) реализация методов и средств автоматической идентификации использованных средств защиты исполняемых файлов; 2) расширение пространства признаков дополнительными элементами, потенциально важными при обучении классификаторов; 3) и оценка применимости данных улучшений при построении иерархической модели принятия решения. Дальнейшая работа в данном направлении будет связана с построением иерархических моделей детектирования, включающих в себя элементы динамического анализа.

Литература

1. *Schultz M., Eskin E., Zadok E., Stolfo S.* Data Mining Methods for Detection of New Malicious Executables // Proc. of the 2001 IEEE Symposium on Security and Privacy, 2001.
2. *Pietrek M.* An In-Depth Look into the Win32 Portable Executable File Format // MSDN Magazine, February 2002.
3. *Kolter J., Maloof M.* Learning to Detect Malicious Executables in the Wild // Proc. of the 10th ACM SIGKDD Intern. Conf. on Knowledge Discovery and Data Mining, 2004.
4. *Mitchell T.* Machine Learning. The Mc-Graw-Hill Companies, Inc., 1997.
5. *Wang J.-H., Deng P.S., Fan Y.-S., Jaw L.-J. et al.* Virus Detection using Data Mining Techniques // Proc. of IEEE 37th Annual 2003 Intern. Carnahan Conf., 2003.
6. *Dai J., Guha R., Lee J.* Efficient Virus Detection Using Dynamic Instruction Sequences // J. of Computers. 2009. Vol.4, N 5, May.
7. *Zhang B.-Y., Yin J.-P., Hao J.-B., Zhang D.-X. et al.* Using Support Vector Machine to Detect Unknown Computer Viruses // Intern. J. of Computational Intelligence Res. 2006. Vol. 2, N 1.
8. *Heavens V.X.* // [Электронный ресурс] *http://vx.netlux.org/* (по состоянию на август 2008 г.).
9. *Kohavi R.* The Power of Decision Tables // Proc. of the 8th European Conf. on Machine Learning, 1995.

10. *Quinlan R.* C4.5 // Programs for Machine Learning, Morgan Kaufmann Publishers, 1993.
11. *Breiman L.* Random Forest // Machine Learning. 2001. Vol. 45, N 1, October.
12. *John G., Langley P.* Estimating Continuous Distributions in Bayesian Classifiers // Proc. of the 11th Conf. on Uncertainty in Artificial Intelligence, 1995.
13. *Bramer M.* Principles of Data Mining. London Limited: Springer-Verlag, 2007.

Комашинский Дмитрий Владимирович — аспирант лаборатории проблем компьютерной безопасности Учреждения Российской академии наук Санкт-Петербургского института информатики и автоматизации РАН (СПИИРАН). Область научных интересов: информационная безопасность, детектирование и анализ вредоносного программного обеспечения. Число научных публикаций — 9. komashinskiy@comsec.spb.ru, www.comsec.spb.ru; СПИИРАН, 14-я линия В.О., д.39, Санкт-Петербург, 199178, РФ; р.т. +7(812)328-2642, факс +7(812)328-4450. Научный руководитель — И.В. Котенко.

Komashinskiy Dmitriy Vladimirovich — Ph.D. student Laboratory of Computer Security Problems, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS). Research interests: information security, detection and analysis of malware. The number of publications — 9. komashinskiy@comsec.spb.ru, www.comsec.spb.ru; SPIIRAS, 39, 14th Line V.O., St. Petersburg, 199178, Russia; office phone +7(812) 328-2642, fax +7(812)328-4450.

Котенко Игорь Витальевич — д-р техн. наук, проф., заведующий лабораторией проблем компьютерной безопасности Учреждения Российской академии наук Санкт-Петербургского Института Информатики и автоматизации РАН (СПИИРАН). Область научных интересов: безопасность компьютерных сетей, в том числе управление политиками безопасности, разграничение доступа, аутентификация, анализ защищенности, обнаружение компьютерных атак, межсетевые экраны, защита от вирусов и сетевых червей, анализ и верификация протоколов безопасности и систем защиты информации, защита программного обеспечения от взлома и управление цифровыми правами, технологии моделирования и визуализации для противодействия кибер-терроризму. Число научных публикаций — более 450. ivkote@comsec.spb.ru, www.comsec.spb.ru; СПИИРАН, 14-я линия В.О., д. 39, Санкт-Петербург, 199178, РФ; р.т. +7(812)328-2642, факс +7(812)328-4450.

Kotenko Igor Vitalievich — Dr.Sc., Professor, Head of Laboratory of Computer Security Problems, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS). Research interests: Computer network security, including security policy management, access control, authentication, network security analysis, intrusion detection, firewalls, deception systems, malware protection, verification of security systems, digital right management, modeling, simulation and visualization technologies for counteraction to cyber terrorism; The number of publications — more 450. ivkote@comsec.spb.ru, www.comsec.spb.ru; SPIIRAS, 39, 14th Line V.O., St. Petersburg, 199178, Russia; office phone +7(812) 328-2642, fax +7(812)328-4450.

Шоров Андрей Владимирович — аспирант лаборатории проблем компьютерной безопасности Учреждения Российской академии наук Санкт-Петербургского Института Информатики и автоматизации РАН (СПИИРАН). Область научных интересов: безопасность компьютерных сетей, обнаружение вторжений, защита от вирусов. Число научных публикаций — 7. ashorov@comsec.spb.ru, www.comsec.spb.ru; СПИИРАН, 14-я линия В.О., д.39, Санкт-Петербург, 199178, РФ; р.т. +7(812)328-2642, факс +7(812)328-4450.

Shorov Andrey Vladimirovich — Ph.D. student of Laboratory of Computer Security Problems, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS). Research interests: computer network security, intrusion detection, antivirus protection. The number of publications — 7. ashorov@comsec.spb.ru, www.comsec.spb.ru; SPIIRAS, 39, 14th Line V.O., St. Petersburg, 199178, Russia; office phone +7(812) 328-2642, fax +7(812)328-4450.

Поддержка исследований. Работа выполнена при финансовой поддержке РФФИ, программы фундаментальных исследований ОНИТ РАН (контракт №3.2/03) и других проектов.

Статья поступила в редакцию 00.03.2010.

РЕФЕРАТ

Комашинский Д.В., Котенко И.В., Шоров А.В. **Подход к обнаружению вредоносного программного обеспечения на основе позиционно-зависимой информации.**

Работа рассматривает подход к обнаружению вредоносного ПО на основе позиционно-зависимой информации. Проблема противодействия вредоносному программному обеспечению, все еще остается актуальной, хотя в настоящее время появляются все более эффективные механизмы для детектирования вредоносного ПО. В работе рассматривается применение методов интеллектуального анализа данных (Data Mining) для решения этой проблемы. Новизна подхода, описываемого в исследовании, заключается в направленности на обработку позиционно-зависимой статической информации, обеспечивающую формирование отдельных элементов эффективной модели детектирования вредоносных исполняемых объектов.

Основная идея, предлагаемого в работе подхода, заключается в использовании особенностей формата файлов, которые могут включать вредоносный код. Знание характера и структуры информации, включаемой в потенциально опасный объект, позволяет уменьшить область данных, которую необходимо проанализировать.

Для функционирования базовых классификаторов использовались такие методы, как Decision Table, C4.5, RandomForest и Naive Bayes. В качестве признаков для обучения выбраны связки величин «Позиция—Значение».

Для экспериментов ~~были~~ выделены две базы исполняемых файлов, содержащие 5854 опасных и 1656 неопасных файлов соответственно.

Для извлечения наборов признаков разработана утилита разбора файлов формата PE32, ориентированная на доступ к контенту файла по относительным виртуальным адресам. Данная утилита позволяет генерировать на выходе файлы формата Attribute—Relation (ARFF), включающие наборы всех возможных признаков. Для осуществления экспериментов использовался программный пакет Weka 3.6.1.

Эксперименты показали, что применение позиционно-зависимых признаков является достаточно эффективным при использовании методов Data Mining, относящихся к группам классификаторов, использующих генерацию правил и построение деревьев решений. Наиболее эффективным показал себя метод RandomForest

Предлагаемый подход не дает абсолютной точности детектирования вредоносного ПО, но может быть эффективен на определенных фазах процесса принятия решения о способе дальнейшей обработки объекта и при построении средств детектирования вредоносного ПО. В качестве примера можно привести задачу автоматизации обнаружения и идентификации использованных средств обфускации или защиты исполняемых файлов.

SUMMARY

Komashinskiy D.V., Kotenko I.V., Shorov A.V. **Approach to detect malware based on positionally dependent information.**

The work examines an approach for detecting malware on the basis of positionally dependent information. The problem of counteracting malicious software is still relevant, although currently there are more effective mechanisms for detecting malware. This paper considers the application of methods of data mining to solve this problem. The novelty of the approach described in the paper is to focus on the processing of positionally dependent static information, ensuring the formation of particular elements of an effective model of detecting malicious executables.

The main idea of the proposed approach is to use the features of a file format, which can include malicious code. Knowledge of the nature and structure of the information included in a potentially dangerous object can reduce the amount of data that should be analyzed.

For the functioning of basic classifiers methods such as Decision Table, C4.5, RandomForest and Naive Bayes were used. The bunches of variables «Position—Value» as features for learning were selected.

For the experiments two bases of executable files were selected. They contain 5854 hazardous files and 1656 non-hazardous files.

To extract the sets of features, we developed the utility for parsing the files of PE32 format. It focuses on access to the file content using relative virtual addresses. This utility can generate output files of Attribute-Relation (ARFF) format, including the set of all possible features. To carry out experiments we used a software package Weka 3.6.1.

Experiments have shown that the use of positionally dependent characteristics is quite effective when Data Mining methods are used, which related to classifiers using generation rules and the construction of decision trees. The most effective method was RandomForest.

The proposed approach does not provide absolute accuracy of detecting malware, but may be effective at certain stages of decision-making on how to further process the object and under construction of malware detectors. As an example, the task of automating the detection and identification of used obfuscation or protection tools for executable files.