Исследование структурных особенностей вредоносных документов методами Data Mining

Д.В. Комашинский, И.В. Котенко

Аннотация. Работа посвящена проблеме выявления вредоносных документов с помощью методов Data Mining. Предлагается общий подход к обнаружению подобных документов за счет выявления характерных особенностей их структуры и содержимого. Проводится оценка предлагаемого подхода на примере документов формата Portable Document Format. В ходе проведения экспериментов осуществляется оценивание эффективности применения как отдельных методов классификации, так и методов комбинирования классификаторов, построенных на отдельных группах признаков.

Ключевые слова. Информационная безопасность, вредоносные документы, Portable Document Format, классификация, Data Mining.

Введение

Проблема вредоносных документов в области информационной безопасности имеет достаточно долгую историю. Около пятнадцати лет назад появились так называемые макровирусы [2], использующие в качестве среды распространения документы Microsoft Office, а в качестве исполнительного механизма - встроенную в данную систему среду разработки на языке программирования Visual Basic for Applications. Простота процесса создания и эффективность функционирования обеспечили крайне благоприятные условия для их существования на протяжении последующих пяти лет, несмотря на то, что меры противодействия им и в техническом, и в организационном плане были достаточно очевидны и просты. Примерно в это же время с продвижением Интернет и ростом количества доступных и понятных пользователям сетевых сервисов началась эпоха вредоносных Web-документов, основанных на динамической композиции языка гиперразметки текста (Hypertext Markup Language) и языков программирования JavaScript и Visual Basic Script. В данном случае, как правило, вредоносные документы обрабатывались в контексте Web-браузеров и не имели возможности саморепликации, но активно использовались для загрузки и запуска на стороне пользователей вредоносных бинарных файлов. Со временем компании-производители приложений браузеров смогли найти некоторый набор компромиссных решений, удовлетворяющих как общим требованиям безопасности, так и ожиданиям пользователей. Это спровоцировало усиление внимания представителей киберкриминалитета к вопросам эксплуатации так называемых уязвимостей, что перевело процесс противостояния вредоносным документам на качественно новый уровень.

Начиная с 2008 года злоумышленники начинают активно искать и исследовать возможности применения уязвимостей приложений, работающих с одним из самых известных файловых форматов электронных документов - Portable Executable Format (известный под аббревиатуре файлового расширения PDF). В отличие от

предыдущих типов угроз, этот тип базировался на эксплуатации технических особенностей операционных систем и программных ошибках эксплуатируемых приложений. Нужно отметить, что проблема уязвимостей характерна и для других сред создания и просмотра документов, например, в настоящее время схожие проблемы характерны для упомянутого программного пакета Microsoft Office [22] и практически всех приложений-браузеров.

Работа по противодействию уязвимостям ведется в нескольких плоскостях. Усилия производителей аппаратного и системного программного обеспечения выливаются в появление новых платформ и приложений, использующих ряд таких известных технологий, как Hardware Data Execution Prevention (DEP), Address Space Layout Randomization (ASLR), Software Data Execution Prevention (SAFESEH) [14, 23]. Однако на данный момент улучшение уровня защищенности распространенных операционных систем и развитие средств программной защиты от традиционных вредоносных программ только стимулируют развитие подходов к поиску и эксплуатации новых уязвимостей наиболее распространенных приложений и методов их выполнения [13]. Примером того, что проблема угрозы со стороны документов формата Portable Document Format несмотря ни на что остается актуальной, является выявленная и подтвержденная в середине 2011 года и в настоящее время используемая злоумышленниками уязвимость CVE-2011-2462 [26], специфичная для некоторых продуктов компании Adobe и срабатывающая при обработке вложенных в документ специально подготовленных блоков данных формата Universal 3D (U3D). Другим примером одной из последних потенциальных уязвимостей для пакета Microsoft Office является уязвимость CVE-2011-1983 [20], основанная на использовании приложением Microsoft Word ранее освобожденной им же памяти (“Use-after-free Error”). Пример статистики последних лет для значимых зарегистрированных ошибок типа “ошибка работы с буфером”, “Buffer Error” для некоторых популярных продуктов работы с электронными документами представлен на Рис. 1, данные получены из базы NIST [27].

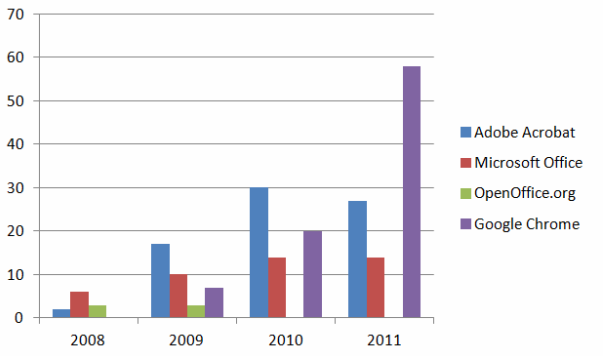

Рис. 1. Статистика количества уязвимостей типа “ошибка работы с буфером”, зарегистрированных для программных пакетов работы с электронными документами

Существующие требования к современным электронным документам как к информационным хранилищам, способным агрегировать не только текстовую информацию, но и активное содержимое, мультимедийный контент и данные других источников данных, обуславливают значительную сложность их структуры и процедур их обработки. Данный факт, наряду с большим объемом существующего кода, отвечающего за реализацию их парсинга, интерпретации и вывода содержимого пользователю, и интенсивным внедрением новых технологий обработки документов, к сожалению, оставляет возможность появления новых, ранее не известных уязвимостей в приложениях, обеспечивающих их просмотр и редактирование. В общем случае, проблему анализа и выявления документов, имеющих вредоносное активное содержимое и специально созданные информационные блоки, обработка которых способна вызвать “срабатывание” уязвимостей, можно охарактеризовать как задачу анализа сложных иерархических структур, определяющих информационное содержимое отдельных элементов документа и их взаимосвязей. Это характерно для всех существующих форматов документов, отвечающих современным требованиям, таких как HTML, используемый для представления Web-документов в браузерах, структурированные OLE-хранилищ (Object Linking Embedding), используемые в документах Microsoft Office, и Portable Document Format (PDF), детально рассматриваемого в предлагаемой работе.

Наряду с этим следует отметить ряд аспектов, делающих возможной разработку достаточно простых, но эффективных методик выявления вредоносных документов. Феномен вредоносных документов в настоящее время является массовым. Практически любой персональный компьютер, связанный с сетью Интернет и работающий с документами, является потенциально уязвимым. В результате появились пакеты использования уязвимостей (exploit kits), работающие, как правило, в контексте Web-серверов. Обычно для того, чтобы обеспечить возможность эксплуатации уязвимостей на стороне пользователя и выполнение на его компьютере вредоносного кода эти пакеты используют отдельные программы - генераторы вредоносных документов, обеспечивающие создание уникального экземпляра вредоносного документа для каждого атакуемого пользователя. Это обуславливает "полиморфность" бинарного образа документа и затрудняет его анализ средствами антивирусной защиты.

Необходимость генерации подобных документов заставляет злоумышленников следовать базовым требованиям спецификаций форматов документов и, тем самым, оставляет в генерируемых документах "следы" генераторов – определенные структурные статические паттерны, свойственные заложенным в ним функциональным особенностям. Примером таких особенностей могут служить наличие или отсутствие определенных предусмотренных (и не предусмотренных) блоков данных, которые сами по себе не способны нанести вред пользователю, но могут дать опосредованную информацию об источнике документа.

Одним из подходов к противодействию вредоносному программному обеспечению является применение методов машинного обучения и интеллектуального анализа данных (Data Mining) [15]. Их использование обусловлено возможностью построения систем детектирования и категоризации отдельных типов угроз с характеристиками, удовлетворяющими требованиям некоторых процессов поддержки принятия решения в области компьютерной безопасности [1].

Проверка рассматриваемого данной работой подхода была осуществлена на примере формата Portable Document Format (PDF). Выбор именно этого формата обусловлен рядом причин:

1) угрозы, свойственные приложениям, работающим с данным форматом, по-прежнему актуальны;

2) существует достаточно большое количество данных, позволяющих провести необходимые эксперименты и получить и проверить необходимые обобщения;

3) использование для проведения экспериментов подобных данных не нарушает картину в целом, полученные обобщения могут быть перенесены на другие типы документов;

4) данный формат является открытым, что подразумевает наличие достаточного количества информации для раскрытия сути его составляющих и существование инструментария его обработки;

5) проведение экспериментов именно с этим форматом позволяет получить хотя бы грубую сравнительную оценку эффективности предлагаемого подхода в силу наличия публикаций, посвященных теме обнаружения вредоносных документов данного формата с помощью других подходов.

В настоящей работе предлагается статический подход к обнаружению вредоносных электронных документов на основе методов Data Mining. Статья структурирована следующим образом. Во втором разделе типовой документ PDF рассматривается как сложная и упорядоченная структура, имеющая данные определенного типа. Дается краткий обзор особенностей уязвимых приложений, работающих с данным форматом, представляется статический подход к обработке документов, приводится обобщенный и конкретизированный перечень используемых признаков, а также определяется перечень экспериментов, необходимых для выполнения в рамках данного подхода. Третий раздел посвящен детальному описанию проведенных экспериментов, представлению результатов и их интерпретации. Обсуждение полученных результатов проводится в четвертом разделе настоящей работы. В пятом разделе выполняется краткий обзор существующих работ, посвященных тематике выявления и анализа вредоносных документов и доступного инструментария, облегчающего

выполнение задачи проведения описанных документов.

1. Предлагаемый подход к обнаружению вредоносных документов

Для более глубокого понимания предлагаемого подхода следует отдельно остановиться на предпосылках популярности явления использования уязвимостей приложений, работающих с форматом PDF. Эти предпосылки могут быть обобщены по отношению и к другим популярным файловым форматам и приложениям, используемым злоумышленниками для внедрения вредоносного программного обеспечения.

Данный формат является стандартом дефакто для всех видов электронного документооборота. Среди множества доступных систем создания и просмотра электронных документов с самого начала существовала лидирующая линейка продуктов от компании Adobe Systems [19]. Причина этого успеха в том, что данная компания сумела объединить роли основного разработчика спецификации данного формата и соответствующих ему программных продуктов и фактически стала законодателем в этой области. Обратной стороной этого факта является существование на данный момент огромного количества программного кода, который разрабатывался с 90-х годов прошлого века и по размеру сопоставим с объемом кода современных операционных систем [17]. Это обуславливает потенциальное наличие проблем контроля безопасности программных решений, использующих его. Уже к середине 2000-х годов сложилась ситуация, когда на значительной части персональных компьютеров вне зависимости от используемой операционной системы были установлены программные продукты данной компании [8], фактически формируя единую, потенциально уязвимую кроссплатформенную программную среду. По мере эволюции вредоносного программного обеспечения и перехода к методу заражения Drive-by-Download [4] в 2008 году это положение вещей стали использовать компьютерные злоумышленники.

Формат Portable Document Format описывается открытым стандартом ISO 32000 [21]. Он является одним из наиболее популярных в

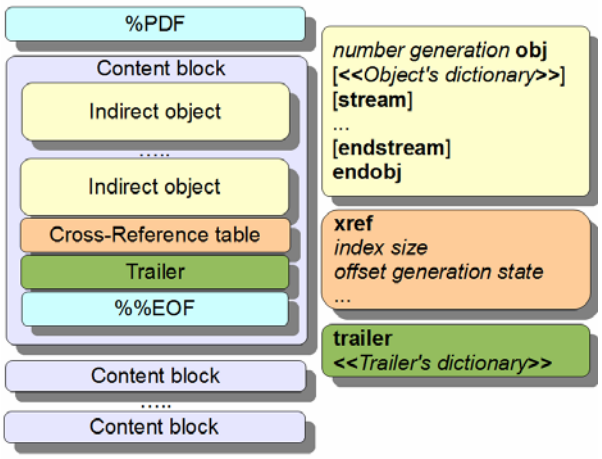

Рис. 2. Структура типового документа PDF

электронном документообороте кроссплатформенным форматом и активно развивается за счет постоянного внедрения новых возможностей, связанных с поддержкой мультимедиаконтента, интерактивных средств ввода и обработки информации, криптографических методов ограничения доступа к информации.

Структура типового документа представлена на Рис. 2. Каждый документ традиционно должен иметь заголовок (%PDF header), идентифицирующий его формат и версию спецификации, которой он соответствует, и как минимум один признак завершения (%%EOF ending). Для ускорения доступа к отдельным элементам документа предусмотрено понятие так называемой таблицы перекрестных ссылок (cross-reference table), обеспечивающей точное позиционирование в теле документов отдельных структурных блоков документа, представляющих собой косвенные объекты (Indirect Objects).

В зависимости от версии спецификации документа существует два основных способа поиска таблицы перекрестных ссылок: смещение начала таблицы перекрестных ссылок указывается непосредственно перед признаком окончания документа; таблица перекрестных ссылок упакована в косвенный объект, и его смещение указано в одном из полей специальной структуры, называемой трейлером (trailer).

Кроме позиционирования таблицы перекрестных ссылок, функцией трейлера является позиционирование некоторых других ключевых объектов документа, таких как его корневой каталог, хранилище информации об источнике

документа, ссылки на следующий трейлер и так далее. В общем представлении конечным структурным элементом данного формата можно считать уже упомянутый косвенный объект, являющийся терминальным объектом, содержащим информацию об определенном структурном элементе документа. Все косвенные объекты имеют тот или иной тип, определяющий место объекта в общей иерархической структуре документа и процедуру его интерпретации.

Одна из особенностей основного набора приложений, обеспечивающих просмотр и редактирование документов формата Portable Document Format, состоит в том, что они толерантны по отношению к внутренним ошибкам описания структуры документов. Необходимость обеспечить максимальные возможности отображения содержимого всего множества существующих документов, которые далеко не всегда полностью соответствуют спецификации формата, привела к появлению данной особенности программных продуктов.

Например, в соответствии с текущим стандартом [21], файловый заголовок документа должен находиться в самом начале документа по нулевому смещению. Однако на практике это не совсем верно [8]. Современные утилиты просмотра документов данного формата, как правило, способны успешно обработать документ даже в том случае, если смещение заголовка значительно отличается от нуля и находится не в первой строке. Это обуславливает возможность формирования документа, который имеет заведомые ошибки во внутренней структуре, но, тем не менее, будет успешно интерпретирован и отображен большинством доступных приложений, работающих с документами. На практике данная возможность в той или иной степени вступает в противоречие с большинством программных компонентов сканирования документов на предмет наличия в них вредоносного кода и контента, вызывающих срабатывание уязвимостей, и в ряде случаев используется злоумышленниками [3, 6, 8, 12, 16].

Таким образом, одной из актуальных структурных особенностей вредоносных документов является потенциальное наличие в них ряда структурных аномалий, обуславливающих присутствие (равно как и отсутствие) информации, в той или иной мере влияющей на степень глубины обработки документов программными средствами, соответствующими стандарту ISO 32000 [21]. Кроме того, в соответствии с работами [8, 16], посвященными статическому анализу вредоносных документов, существует ряд дополнительных косвенных признаков, в той или иной мере свойственных им. Например, Кубек и др. [8] сообщают о том, что большинство вредоносных документов содержат не более одной страницы и во многих случаях вообще не имеют содержимого, отображаемого в ее графическом представлении. Тзермиас и др. [16] приводят пример комбинированного использования техники обфускации имен и использования значений атрибутов происхождения документа для затруднения анализа вложенного кода JavaScript. Описанные выше и некоторые другие особенности вредоносных документов доступны на этапе их статического разбора, что ставит вопрос о применимости методов машинного обучения в системах их автоматического выявления и категоризации.

Суть предлагаемого в данной работе подхода состоит в применении методов Data Mining для обнаружения новых, ранее не известных вредоносных документов формата Portable Document Format за счет выявления ряда аномалий в их структуре и содержании. Работа рассматривает применимость методов классификации для формирования систем детектирования вредоносных документов в контексте использования на фазе обучения отдельных групп структурных признаков.

В качестве опорных классификаторов используются дерево решений (Decision Tree, DT), наивный байесовский классификатор (Naive Bayes, NB) и метод k ближайших соседей (k Nearest Neighbors, k-NN).

Рассматривается применимость данных классификаторов для следующих групп признаков (общее количество признаков равно 1602 без метки класса).

- *Группа происхождения документов (Origin).* Применительно к формату PDF (далее это справедливо по отношению ко всем последующим группам признаков), данная группа включает в себя численные и булевы признаки

наличия в документе тех или иных информационных атрибутов (пункт 14.3.3 ISO 32000 [21]) информационного каталога документа.

- *Группа типов информационного наполнения (Types).* Включает численные и булевы признаки наличия в документе косвенных объектов тех или иных типов (тип указывается значением атрибута /Type).
- *Группа используемых методов упаковки косвенных объектов (Filters).* Включает численные и булевы признаки использования в документе методов упаковки содержимого потоковых косвенных объектов (в большинстве случаев тип указывается значением атрибута /Filter).
- *Группа использования техники обфускации имен (Obfuscation).* Эта группа охватывает все численные и булевы признаки наличия в документе обфусцированных (пункт 7.3.5 ISO 32000:2008 [21]) имен.
- *Группа использования автоматических событий (Actions),* состоящая из численных и булевых признаков наличия в документах основных типов автоматических событий (пункт 12.6.4.1 ISO 32000:2008 [21]).
- *Группа использования XML форм (XML Forms Architecture, XFA).* Включает численные и булевы признаки наличия аномалий в XFA-содержимом (максимальное количество элементов в одном XFA-контейнере, отсутствие атрибута имени у первого элемента XFA-контейнера, отсутствие в документе косвенных объектов, указанных в XFA-контейнере, неправильное форматирование начала и конца XFA-данных).
- *Основная группа аномалий и структурных особенностей (Basic).* Данная группа включает булевы признаки наличия в документе признаков XFA, Actions, Origin, Filters, Obfuscation; все признаки группы XFA; булевы признаки группы Actions; булевы признаки наличия тех или иных значений атрибута /Filter потоковых косвенных объектов и их цепочек (пункт 7.3.8 ISO 32000 [21]); булевы признаки наличия в документе ряда потенциально важных обфусцированных лексем (например, обфусцированных названий методов упаковки, соответствующих спецификации [21]); численные и булевы признаки наличия аномалий в основных структурах документа (количество страниц, отсутствие контента первой страницы, отсутствие ссылки на информацию о происхождении в трейлерах документа, отсутствие ссылки на корневой каталог документа в трейлерах документа, отсутствие в документе косвенного объекта с типами /Info и /Root, наличие в структурах документа некорректных смещений на косвенные объекты и перекрестные таблицы ссылок).

Помимо качественного анализа значимости тех или иных групп признаков для процедуры обнаружения вредоносных докуменов указанными выше методами классификации рассматривается частный вопрос эффективности методов комбинирования отдельных полученных классификаторов с помощью голосования (Vote) и стекинга с использованием в качестве опорного классификатора метода дерева решений (Stacking).

Основной вклад предлагаемой работы состоит в анализе применимости наиболее распространенных методов классификации для обнаружения вредоносных документов формата Portable Document Format и в оценке значимости отдельных групп статических структурных признаков для поиска вредоносных документов. Насколько известно авторам данной работы, проблема поиска вредоносных документов на основе использования методов Data Mining в указанном выше аспекте ранее не обсуждалась. Следует отметить, что наиболее близкой по форме проведения исследований и подачи результатов является работа [8]. Однако ее авторами не затрагивалась задача автоматической генерации правил детектирования вредоносных документов. Эта задача частично решается в данной работе в контексте анализа результатов, полученных в ходе использования классификатора, основанного на деревьях решений, и предыдущих исследований авторов [7, 29-31].

2. Эксперименты по исследованию вредоносных документов

В число учитываемых групп признаков были включены как ранее известные группы, детально рассмотренные в [3, 6, 8, 12, 16], так и новые группы признаков, значительно расширяющие понимание природы и внутренних зависимостей некоторых характеристик содержимого докумен-

тов (группы происхождения документов и типов их информационного наполнения). В качестве исходных данных для проведения процедур обучения и валидации полученных моделей классификации использовались материалы коллекции VxHeawens [28] (4382 файлов) и набор безопасных PDF файлов (5111 файлов), собранный из разнообразных источников Интернет.

Для извлечения признаков из исходных данных применялся частично модифицированный программный пакет Open PDF Analysis Framework [24]. Формирование рабочих схем эксперимента и его вычислительная поддержка осуществлялась с помощью программного пакета RapidMiner 5.1 [25].

Проверка применимости описанных в предыдущем разделе групп статических атрибутов была проведена для следующих методов классификации: дерево решений (Decision Tree, DT), наивный байесовский классификатор (Naive Bayessian classifier, NB) и метод k ближайших соседей (k nearest neighbors, k-NN) с использованием как всего набора доступных признаков, так и их отдельных групп. Кроме того, была проведена серия экспериментов по комбинированию отдельных групп полученных DT-классификаторов на основе применения двух методов мета-классификации - стекинга (Stacking) и голосования (Vote).

Вычисление характеристик точности классификаторов осуществлялось методом десятикратной кросс-валидации.

В качестве основного показателя было выбрано понятие точности (Accuracy или Acc), определяемое как отношение количества правильно классифицированных объектов к общему количеству объектов:

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn}, \qquad (1)$$

где *tp* (true positive) – количество правильно классифицированных вредоносных документов, *tn* (true negative) – количество правильно классифицированных безопасных документов, *fp* (false positive) – количество безопасных документов, ошибочно классифицированных как опасные и *fn* (false negative) – количество опасных документов, ошибочно классифицированных как безопасные.

Начальный эксперимент проводился с использованием всех доступных 1602 атрибутов. Лучший результат по точности (99.88 %) продемонстрировал классификатор на основе DT, наименее точным оказался метод NB (98.81). На Рис.3 показана структура полученного дерева решений.

Дерево решений, полученное при обучении на всем наборе доступных атрибутов, является несбалансированным двоичным деревом. Корневому узлу дерева соответствует признак “types_num_/Unknown”, определяющий в анализируемом документе количество косвенных объектов, не имеющих установленного значения атрибута “/Type”. Концевые узлы дерева

```
types_num_/Unknown > 6.500
|   filters_num_/ASCIIHexDecode > 0.500: malicious {malicious=22, benign=0}
|   filters_num_/ASCIIHexDecode = 0.500
|   |   declared_info_len_/hrtg > 7.500: malicious {malicious=2, benign=0}
|   |   declared_info_len_/hrtg = 7.500
|   |   |   actions_has_/JavaScript = 0: benign {malicious=0, benign=4688}
|   |   |   actions_has_/JavaScript = true
|   |   |   |   declared_info_len_/Title > 46: malicious {malicious=10, benign=1}
|   |   |   |   declared_info_len_/Title = 46: benign {malicious=3, benign=73}
types_num_/Unknown = 6.500
|   declared_info_len_/Producer > 0.500: benign {malicious=0, benign=336}
|   declared_info_len_/Producer = 0.500
|   |   types_has_/Metadata = 0
|   |   |   types_has_/XObject = 0: malicious {malicious=4345, benign=1}
|   |   |   types_has_/XObject = true: benign {malicious=0, benign=3}
|   |   types_has_/Metadata = true: benign {malicious=0, benign=9}
```

Рис. 3. Дерево решений для всего набора атрибутов

представляют конечные решения о степени опасности документа – безопасный (benign) или вредоносный (malicious).

Пороговым значением узла ветвления в корне дерева является числовая величина 6.5, определяющая дальнейшую последовательность принятия решения о степени вредоносности документа. В упрощенной интерпретации это означает, что если количество подобных косвенных объектов в структуре анализируемого документа больше шести, то дальнейший процесс его обработки будет соответствовать поддереву с вершиной “filters_num_/ASCIIHexDecode”. Примечательно то, что фактически это определяет значимость оценки потенциальной сложности содержимого документа на начальном этапе его анализа. По нашим оценкам, числовое значение, характеризующее величину “types_num_/Unknown”, прямо пропорционально степени сложности (объему разнородных данных) анализируемого документа.

Таким образом, данное дерево представляет две основные стратегии анализа документа в зависимости от степени его сложности: 1) если структура документа достаточно сложная (количество косвенных объектов с неопределенным типом больше шести), он анализируется с помощью поддерева с вершиной “filters_num_/ASCIIHexDecode”; 2) если структура документа относительно проста (количество косвенных объектов с неопределенным типом меньше или равно шести), он анализируется с помощью поддерева с узлом “declared_info_len/Producer”.

В первом случае для принятия конечного решения о степени опасности документа используются данные о применении в документе преобразования потоковых данных с помощью метода ASCIIHexDecode (узел “filters_num_/ASCIIHexDecode”, характеризующий количество в документе потоков, обработанных данных методом); происхождении документа; наличии в нем активного скриптового содержимого (узел “actions_has_/JavaScript”).

При учете происхождения документа особый интерес представляют установленные пороговые величины для соответствующих узлов дерева “declared_info_len_/hrtg” и “declared_info_len_/Title”, характеризующих длину соответствующих атрибутов информационного блока документа “/hrtg” и “/Title”. В отличие от атрибута “/Title”, атрибут “/hrtg” не установлен спецификацией [21], и его наличие в анализируемом документе может рассматриваться как косвенный признак вредоносности. Анализ доступных вредоносных документов показывает, что внедрение неизвестных атрибутов (зачастую сгенерированных случайным образом) в информационный блок документа является одним из стандартных способов, используемых злоумышленниками для затруднения процедур их структурного разбора и интерпретации результатов. Таким образом, установленное для данного атрибута пороговое условие должно интерпретироваться как проверка его наличия (для условия “>7.5”). Это обуславливает некоторые сложности в интерпретации получаемых порогов для числовых атрибутов и показывает положительные стороны использования методов предварительной дискретизации числовых величин. Числовой порог для узла “declared_info_len_/Title” показывает, что в общем случае слишком длинное (больше 46 символов) название документа (указанный заголовок документа в информационном блоке не обязательно соответствует его реальному заголовку, размещаемому на его первой странице) является потенциальным признаком вредоносности.

Во втором случае документ анализируется с помощью поддерева с узлом “declared_info_len/Producer”. Этот атрибут является числовой характеристикой длины поля “/Producer” информационного блока документа, и в данном случае его пороговое значение характеризует наличие или отсутствие атрибута, указывающего название программного приложения, сгенерировавшего документ. Как показано на рисунке, наличие указанного признака в большей степени свойственно безопасным документам. Кроме того, в поддереве используются булевы признаки наличия в документе косвенных объектов с типами “/Metadata” (объекты данного типа представляют расширенную альтернативу информационного блока документа) и “/XObject” (подобные объекты являются потоками данных, содержащими последовательности графических объектов), представленные узлами “types_has_/Metadata» и “types_has_/XObject” соответственно. Наличие объектов

данных типов более свойственно безопасным документам с простой структурой.

Результаты начального эксперимента показали, что тремя наиболее значимыми атрибутами, определяющими основные ветви дальнейшего принятия решения, являются: 1) количество в документе косвенных объектов с неизвестным типом (другими словами, количество косвенных объектов с неопределенным значением атрибута /Type) с пороговым значением 6.5; 2) фактическое использование в документе метода упаковки ACSIIHexDecode с пороговым значением 0.5; 3) длина строкового значения атрибута /Producer в информационном каталоге документа.

Понимание данных признаков можно упростить до следующего набора правил.

- **Правило 1.1 -** "*в документе количество косвенных объектов с неопределенным типом больше шести*". На практике данное правило устанавливает некоторый порог на информационный объем документа. Наличие косвенных объектов с неопределенным типом является показателем структурной сложности документа. Можно показать, что сложность документа определяется разнообразием его содержимого. Наличие такового более свойственно для документов большого объема, содержащего разнородные блоки контента. Очевидно, что структурная сложность в значительной мере определяет общий подход к анализу содержимого документа, что и объясняет значимость данного правила и соответствующего ему атрибута.
- **Правило 1.2 -** "*в документе используется метод упаковки ACSIIHexDecode*". Это правило напрямую устанавливает зависимость между вредоносностью документа и наличием в нем контента, использующего данный метод упаковки. Следует отметить, что данный результат полностью соответствует выкладкам авторов работы [8] и подтверждает, что в некоторых случаях использование метода преобразования ASCIIHexDecode свойственно исключительно вредоносным документам.
- **Правило 1.3 -** "*в документе указано название программного продукта, с помощью которого документ был создан*"; подробное описание атрибута /Producer приведено в пункте 14.3.3 стандарта ISO 32000 [21]. Проведенный эксперимент показал, что в большинстве случаев наличие информации о происхождении анализируемого документа свойственно безопасным документам.

В общем случае данные результаты указывают на то, что атрибуты, харатеризующие косвенные признаки содержимого документа (например, как в рассмотренном выше правиле 1.1), являются важными для опосредованного принятия решения о степени опасности документа. Это заключение близко к тому наблюдению, что файлы с минимальным количеством содержимого с большей долей вероятности имеют потенциально опасное содержимое [8]. Однако, по понятным причинам, группы признаков Types и Origin не могут быть использованы для однозначного принятия решения о степени угрозы.

Данное наблюдение обусловило проведение двух последующих этапов эксперимента: на втором этапе из всего списка доступных атрибутов была исключена группа атрибутов Types, а на третьем шаге, кроме того, были исключены атрибуты группы Origin. На обоих шагах наилучшие показатели точности продемонстрировали классификаторы k-NN и DT (с небольшим преимуществом метода k-NN).

В тройку наиболее значимых признаков для второго шага эксперимента входят правила, относящиеся к атрибутам групп Origin и Filters.

- **Правило 2.1 -** "в трейлере документа определено расположение информационного каталога документа". Данное правило является обобщением правила 1.3. Существует три основных метода определения информационного блока документа: 1) его определение в трейлере, 2) его определение в корневом каталоге документа, 3) его прямое определение через установку значения атрибута /Types без указания в других источниках. Первый вариант наиболее свойственнен безопасным документам и, судя по всему, является крайне важным признаком на фоне остальных.
- **Правило 2.2 -** "в документе используется метод упаковки FlateDecode как минимум для содержимого пяти косвенных объектов". Наличие большого количества разнообразного контента обуславливает появление в структуре большого количества косвенных объектов, упа-

кованных стандартными методами упаковки (FlateDecode). Таким образом, в указанном случае данное правило косвенно определяет некоторые границы структурной сложности документа, важные для дальнейшего процесса принятия решения.

- **Правило 2.3 -** "в документе используется метод упаковки ASCIIHexDecode как минимум для одного косвенного объекта". Данное правило полностью соответствует правилу 1.2.

Как показало дерево решений для третьего шага эксперимента, наиболее существенными для принятия решения являются следующие правила:

- **Правило 3.1 -** "*в документе используется метод упаковки FlateDecode как минимум для содержимого трех косвенных объектов*". Данное правило является вариацией правил 1.1 и 2.2 и определяет значимость учета структурной сложности документа на первых шагах принятия решения.
- **Правило 3.2 -** "в документе используется метод упаковки ASCIIHexDecode как минимум для одного косвенного объекта". Данное правило было рассмотрено на примере правила 1.2.
- **Правило 3.3 -** "по крайней мере одно из файловых смещений начала таблицы перекрестных ссылок в документе неверно". Наличие данного правила показывает значимость атрибутов группы Basic (отклонения в структуре документов) при отсутствии признаков групп Origin и Types. Тем самым подтверждается общее положение о в значимости атрибутов, характеризующих структурные аномалии, свойственные вредоносным файлам.

Четвертый шаг эксперимента проводился на группе признаков Basic, содержащей информацию об основных типах структурных отклонений, присущих вредоносным документам в соответствии с [8]. Перечень десяти наиболее существенных атрибутов, использованных для построения дерева решений, и общая статистика их наличия в документах обеих категорий представлены в Табл. 1.

Общий перечень шагов эксперимента с указанием примененных методов классификации и групп признаков приведен в Табл. 2.

Дальнейшие этапы эксперимента касались задачи построения моделей детектирования за счет обучения классификаторов на наборах признаков, принадлежащих отдельным группам. В общей сложности было проведено шесть отдельных шагов для признаков групп Types, Origin, Filters, Obfuscation, Actions и XFA. Полный перечень этапов проведенных экспериментов показан в Табл. 2), их результаты представлены в Табл. 3 и на Рис. 4 и 5.

Как можно видеть по результатам шагов FS11 – FS16, наиболее качественные результаты детектирования были продемонстрированы на группах атрибутов Types, Filters и Origin, что подкрепляет значимость результатов для первых трех шагов эксперимента и атрибуты категории Basic, указанные в Табл. 1.

Использование атрибутов групп Actions, XFA и Obfuscation вне контекста групп других признаков не дает возможности построить модели детектирования с приемлемыми характеристиками (точность детектирования в пределах 52-77%).

Табл. 1. Перечень существенных атрибутов при построении дерева решений на пространстве атрибутов группы Basic

Группы признаков и существенный признак	Вредоносные документы (%)	Безопасные документы (%)
[Origin, Basic] Расположение информационного блока документа (информация о происхождении) указано в его трейлере	0.479	89.884
[Filters, Basic] Используется метод упаковки ASCIIHexDecode	1.985	0
[Basic] Расположение корневого каталога документа указано в его трейлере	99.269	91.919
[Basic] В документе неправильно указано смещение на перекрестную таблицу ссылок	62.619	8.315
[Filters, Basic] Один из косвенных объектов документа упакован только с помощью метода упаковки DCTDecode	0	19.604
[XFA, Basic] В документе используются XML-формы (XFA)	50.638	0.215
[Actions, Basic] В документе используются события с типом JavaScript	49.840	1.447
[Actions, Basic] В документе есть автоматические события, определенные атрибутом OpenAction	0.98	0.40716
[Filters, Basic] Используется метод упаковки JPXDecode	0	1.154
[Filters, Basic] При использовании некоторых методов упаковки используется технология предварительной нормализации контента (Predictors)	0.365	21.111

Табл. 2. Описание шагов эксперимента

Шаг эксперимента	Группы признаков	Дополнительная информация
FS1	Все доступные группы	Использованы алгоритмы классификации DT, NB и k-NN
FS2	Все группы за исключением Types	
FS3	Все группы за исключением Types и Origin	
FS4	Группа Basic	
FS11	Группа Types	
FS12	Группа Origin	
FS13	Группа Filters	
FS14	Группа Obfuscation	
FS15	Группа Actions	
FS16	Группа XFA	
CC1	Все доступные группы	Использованы методы мета-классификации Stacking и Vote для комбинирования DT моделей, полученных на шагах FS11-FS16 и FS4
CC2	Все доступные группы	См. шаг CC1. Использованы те же модели, за исключением FS4

Табл. 3. Сводные результаты проведенных экспериментов

Шаг эксперимента	DT			NB			k-NN		
	Acc, %	FP	FN	Acc, %	FP	FN	Acc, %	FP	FN
FS1	99.88	4	7	98.81	108	5	99.85	7	7
FS2	99.84	9	6	84.99	1421	4	99.87	7	5
FS3	99.85	9	5	74.92	2379	2	99.92	2	6
FS4	99.53	7	38	94.85	477	12	99.83	5	11
FS11	99.82	10	7	98.53	96	44	99.82	8	9
FS12	94.42	521	9	89.76	960	12	94.46	519	7
FS13	94.86	239	249	87.96	1143	0	95.10	225	240
FS14	54.02	0	4365	52.81	4480	0	53.57	3965	443
FS15	76.70	13	2199	60.38	3747	14	76.60	250	1971
FS16	77.10	11	2163	46.24	5100	3	49.31	4592	220

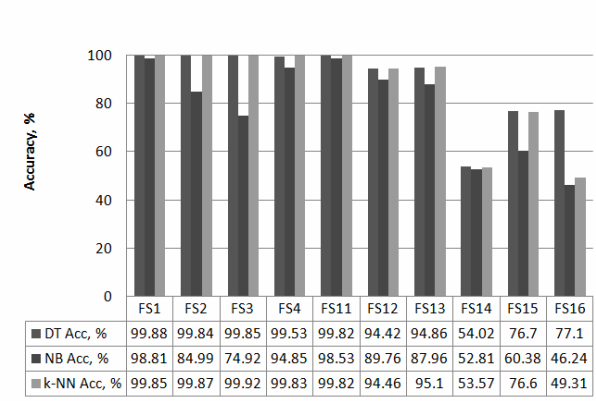

Рис. 4. Показатели точности для экспериментов с отдельными классификаторами, обученными на основных группах признаков

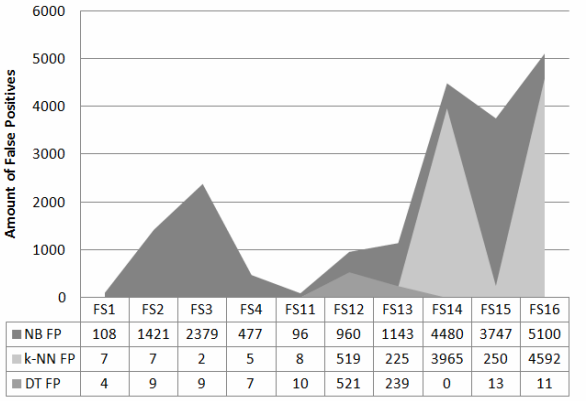

Рис. 5. Показатели ошибки первого рода (FP) для экспериментов с отдельными классификаторами

Относительно скромные результаты использования данных некоторых групп обозначили вопрос потенциальной применимости и полезности этих данных в контексте обобщения результатов, получаемых от отдельных классификаторов. Для ответа на него в рамках данной работы была поставлена задача улучшить результаты применения методов Data Mining для обнаружения вредоносных документов за счет применения отдельных подходов к комбинированию полученных классификаторов (Табл. 2). В частности, был проведен эксперимент по определению значимости классификатора, построенного на данных группы Basic за счет

исключения его из перечня использованных классификаторов.

Результаты проведенных экспериментов по комбинированию классификаторов приведены в Табл. 4.

Рисунки 6 и 7 демонстрируют общие результаты проверки полученных мета-классификаторов. Применение данных методов объединения отдельных классификаторов позволяет обеспечить дополнительное улучшение точности результатов детектирования с наилучшим результатом по точности 99.94% (99.81% для метода Vote) для метода Stacking при использовании моделей, полученных на шагах FS11 – FS16, FS4. Исключение из списка применяемых классификаторов классификатора, обученного на атрибутах группы Basic, ведет к незначительному снижению точности до 99.92 % и 99.80% для методов Stacking и Vote соответственно.

На рисунках 8 и 9 показаны деревья решений, использующие для принятия решений предсказания, получаемые от классификаторов, сформированные на шагах FS11 – FS16 и FS4 эксперимента.

Табл. 4. Результаты комбинирования DT-классификаторов

Шаги эксперимента	Stacking (DT)			Vote		
	Acc, %	FP	FN	Acc, %	FP	FN
FS11 – FS16, FS4	99.94	1	5	99.81	0	18
FS11 – FS16	99.92	3	5	99.80	1	18

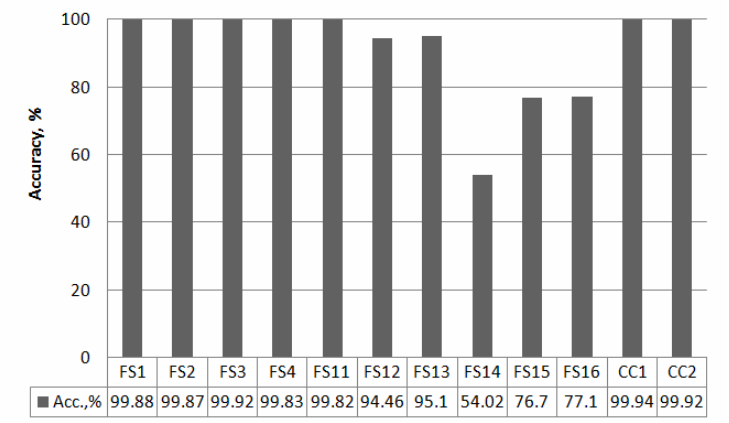

Рис. 6. Наилучшие показатели точности для всех проведенных шагов экспериментов

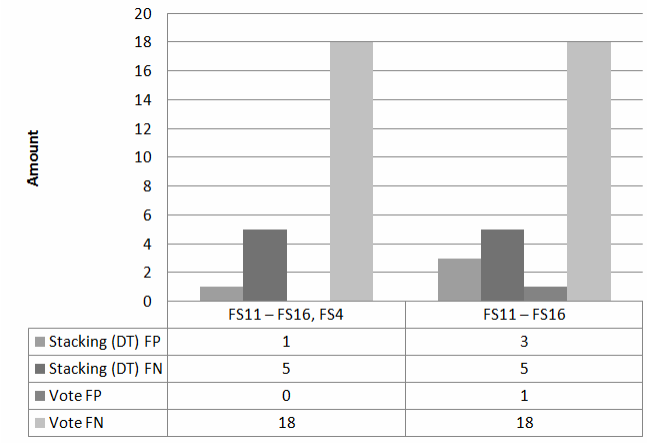

Рис. 7. Показатели ошибок первого (FP) и второго (FN) родов для комбинированных классификаторов

```
types = benign
|   actions = benign
|   |   basic = benign: benign {malicious=0, benign=5057}
|   |   basic = malicious
|   |   |   origin = benign: malicious {malicious=1, benign=1}
|   |   |   origin = malicious: benign {malicious=1, benign=39}
|   actions = malicious
|   |   basic = benign: benign {malicious=0, benign=10}
|   |   basic = malicious: malicious {malicious=3, benign=0}
types = malicious
|   basic = benign
|   |   actions = benign: benign {malicious=0, benign=4}
|   |   actions = malicious: malicious {malicious=13, benign=0}
|   basic = malicious: malicious {malicious=4364, benign=0}
```

Рис. 8. Дерево решений для комбинированного классификатора с использованием группы признаков Basic

```
types = benign
|   actions = benign: benign {malicious=2, benign=5097}
|   actions = malicious
|   |   origin = benign: benign {malicious=1, benign=10}
|   |   origin = malicious: malicious {malicious=2, benign=0}
types = malicious
|   origin = benign
|   |   actions = benign: benign {malicious=0, benign=3}
|   |   actions = malicious: malicious {malicious=14, benign=0}
|   origin = malicious: malicious {malicious=4363, benign=1}
```

Рис. 9. Дерево решений для комбинированного классификатора без использования группы признаков Basic

3. Обсуждение результатов

Полученные результаты показывают высокую значимость группы признаков Types, определяющую дальнейшую последовательность шагов по принятию решения о степени опасности анализируемого документа. Это в значительной мере подтверждается проведенными экспериментами: в настоящее время при проведении атаки конкретного ресурса типовой программный пакет по эксплуатации уязвимостей (exploit kit), как правило, генерирует минимальный по объему документ, структурно содержащий только тот контент, который необходим для срабатывания уязвимости и начала выполнения вредоносного кода при обработке конкретным приложением. Существует и другая крайность, заключающаяся в генерации злоумышленниками документов, сходных с типовыми документами, содержащих множество внутренних данных. Наличие указанных двух групп определяет и появление различных путей анализа документа в зависимости от его содержимого.

Во всех проведенных экспериментах наибольшую точность обнаружения вредоносных документов показали методы классификации DT и k-NN. Наилучшие показатели по количеству ошибок первого рода (False Positives) продемонстрировал метод DT. Применение метода NB без использования дополнительных методов обеспечения точности на данных наборах признаков в большинстве случаев является неприемлемым.

С точки зрения показателей точности, обеспечиваемых данными из различных групп, наиболее привлекательными группами атрибутов являются (в порядке убывания точности) группы Basic, Types, Filters, Origin. В случае применения методов комбинирования классификаторов использование данных последних трех групп нивелирует значимость группы Basic (для мета-классификатора типа Stacking с использованием в качестве обобщающего метода классификации DT).

Отдельно стоят результаты, полученные для группы признаков Obfuscation. Классификаторы,

обученные на этих данных, показали более чем скромные по точности результаты и не используются (или имеют крайне незначительное влияние) в составе комбинированных схем принятия решения. Эти результаты вступают в противоречие с данными работы [8], которая для рассматриваемых исследований в некоторой степени была базовой. Проверка данных показала следующее. В некоторых случаях механизм альтернативной записи символов (пункт 7.3.5 ISO 32000 [21]) используется в безопасных документах. По этой причине сам по себе факт использования техники обфускации не является признаком вредоносности (однако наличие некоторых определенных обфусцированных строк является бесспорным признаком вредоносности документа). Для вредоносных документов свойственно использование техники обфускации конкретных имен атрибутов и их значений, однако со статистической точки зрения таких вредоносных документов крайне мало (например, из используемого нами набора, состоящего из 4382 вредоносных документов, только 7 имеют обфусцированную последовательность символов, формирующую имя метода упаковки /FlateDecode). Вследствие этого на предварительных этапах анализа данных необходимы альтернативные методы выбора значимых признаков (feature selection) для выделения подобных атрибутов.

В заключение еще раз хотелось бы коснуться вопроса точности полученных результатов и их применимости. Как было сказано выше, общее число объектов, рассмотренных при проведении экспериментов, составило 9493 (5111 безопасных и 4382 вредоносных файлов). На практике количество существующих электронных документов на несколько порядков больше. Это подчеркивает необходимость дальнейших исследований в данном направлении, в том числе проведения экспериментов в рамках предложенного подхода на более значительных по объему наборах вредоносных электронных документов, а также расширенных наборах статических признаков. Необходимы также исследования особенностей функционирования генераторов вредоносных документов, внедренных в существующие пакеты эксплуатации уязвимостей. Результаты, полученные на настоящий момент, могут быть использованы для создания программных средств оценки степени опасности электронных документов PDF.

4. Релевантные работы

Проблема анализа вредоносных документов формата Portable Document Format активно исследуется как крупными организациями, так и независимыми исследователями. Как правило, в фокусе отдельных работ находятся практические направления, касающиеся анализа новых уязвимостей, способов сокрытия вредоносного контента и программных средств автоматического разбора файлов и выявления потенциально вредоносного контента. Насколько нам известно, ранее постановка вопроса эффективности использования методов Data Mining для статического детектирования вредоносных PDF-файлов на основе структурных отклонений не осуществлялась. Однако мы считаем необходимым упомянуть ряд смежных работ, результаты которых были частично использованы для формирования начального пространства примененных в работе атрибутов или позволили углубить понимание данной предметной области в целом.

Одной из первых работ, посвященных комплексному анализу проблемы безопасности электронных документов данного формата - работа [3], в которой авторы рассматривают среду просмотра и создания документов как среду программирования. Проводится ретроспективный анализ известных на то время потенциальных уязвимостей формата и среды, и потенциальные сценарии ее злонамеренного использования.

В [12] проводится детальный разбор существующих уязвимостей на конкретных примерах и дается обзор как структурных, так и поведенческих особенностей документов, реализующих ту или иную уязвимость. В [6] рассматривается проблема безопасности электронных документов сквозь призму структурных особенностей и функциональности объектной модели документов, поддерживаемой приложениями Adobe.

В [8] представляются результаты, позволившие нам определить начальный перечень атрибутов, использованных в данной работе.

На основе набора, включающего более 250 тысяч файлов, классифицированных по степени опасности, был проведен статистический анализ наличия ряда характерных для вредоносных документов статических признаков. В ходе работы было выделено подмножество статических признаков, наиболее характерных для данного семейства документов. Помимо этого, авторы затронули вопросы статического анализа кода JavaScript, встречающегося в документах данного формата, и пояснили используемый ими общий подход к обнаружению вредоносных документов на основе формирования групп правил.

В [16] рассматривается система выявления вредоносных документов на основе динамического анализа кода JavaScript, находящегося в документах HTML и PDF. Предлагается классическая архитектура системы раннего выявления кода, выполняющего внедрение множества копий вредоносного кода в адресное пространство атакуемого процесса (так называемый Heap Spraying). Помимо описания предлагаемого подхода, авторы останавливаются на некоторых практических вопросах обработки документов и извлечения из них данных, необходимых для проведения фазы динамического анализа.

В контексте предыдущей работы необходимо отметить некоторые альтернативные популярные программные пакеты, находящиеся в открытом доступе и позволяющие производить первую фазу анализа вредоносности документов. К ним относятся пакет jsunpack [18], используемый для эмуляции кода JavaScript, внедренного в Web-документы, и пакет OPAF [24], служащий для статического разбора PDF-файлов.

Практические вопросы анализа функциональных особенностей потенциального вредоносного кода (shellcode), который может быть внедрен на этапе Heap Spraying, детально рассмотрены в [11]. Работа [9] использует JavaScript машину SpiderMonkey для лексического анализа кода, внедренного в документы формата PDF для извлечения признаков, основанных на представлении элементов кода в виде Abstract Syntax Tree (абстрактного синтаксического дерева, AST).

Отдельно следует отметить работы, посвященные поиску и выявлению вредоносных документов формата OLE, используемых программным пакетом Microsoft Office. В [10] представлен комбинированный подход к обнаружению документов данного формата как статическими, так и динамическими средствами. Особое внимание в работе уделено его структурным особенностям, рассматриваемым в контексте типов угроз, свойственных таким документам. Динамический анализ проводится в рамках использования стандартных приложений – “песочниц” (sandboxes), обеспечивающих выявление вредоносных вызовов функций операционной системы, возникающих в результае процесса Microsoft Office. В [5] проблема анализа документов OLE рассматривается как задача выявления структурных аномалий в отдельных структурах анализируемых файлов.

Заключение

Данная работа посвящена вопросу разработки подхода к обнаружению вредоносных документов с помощью методов Data Mining. Предлагаемый подход в первую очередь орентирован на выявление файловых объектов электронных документов, генерируемых существующими пакетами использования уязвимостей с целью запуска специально внедренного вредоносного кода при их обработке приложениями, установленными на стороне пользователя.

В рамках подхода абстрактный документ рассматривается как объект, имеющий ряд неотъемлемых внутренних и (или) внешних признаков, характеризующих его происхождение (Origin), типы элементов внедренного содержимого (Types), методы их обработки (Filters), использование альтернативных методов формирования структуры и внутреннего описания документа (Obfuscation), типы и характер активного содержимого документа (Actions), наличие и содержимое структур, специфичных конкретным форматам документов (XFA для документов формата PDF) и наличие структурных отклонений, не предусмотренных спецификациями формата (Basic). Каждое семейство признаков представляется набором качественных и количественных атрибутов, описывающих соответствующий ему аспект. Данные

наборы атрибутов формируют пространство атрибутов, используемых для формирования обученных классификаторов.

На примере формата Portable Document Format было показано, что наиболее значимой группой признаков, обеспечивающей наилучшую точность классификации без привлечения других групп признаков, является группа типов элементов внедренного содержимого. Эффективным является и применение групп признаков происхождения, методов обработки содержимого и информации о структурных отклонениях. В среднем, наиболее эффективным для решения данной задачи показал себя метод построения деревьев решений (decision tree), продемонстрировавший наилучшее соотношение показателя точности и количества ошибок первого рода (False Positives).

С использованием полученных данных и моделей классификации был предложен подход к улучшению качества системы выявления вредоносных документов за счет использования методов комбинирования классификаторов, обученных на непересекающихся множествах признаков (относящихся к различным семействам признаков, описанным выше). Результаты проведенных экспериментов подтверждают его эффективность. Наилучшие показатели точности получены для метода стекинга (Stacking) в случае использования в качестве входных атрибутов для классификатора второго уровня ответов классификаторов первого уровня. Применение этой комбинированной схемы классификации, дает улучшение качества классификации за счет привлечения классификаторов первого уровня, обученных на группах признаков происхождения, типов внутреннего содержимого, активного содержимого и отклонения структуры. Примечательно, что группа признаков наличия структурных отклонений, являющаяся частично зависимой от конкретной спецификации формата, может быть исключена из данной комбинированной модели классификации с незначительными потерями точности, что позволяет облегчить дальнейшие процедуры обобщения.

Анализ результатов работы также показал определенные несоответствия значений точности классификации для некоторых групп признаков работам, проведенным другими исследователями, в частности, низкие показатели точности для групп использования альтернативных методов описания структуры и структур, специфичных формату документа. Проверка данных фактов позволила выявить некоторые недостатки используемых в работе инструментальных средств и обозначила направления дальнейшего проведения работ в рамках рассматриваемой темы.

Работа выполняется при финансовой поддержке Министерства образования и науки Российской Федерации (государственный контракт 11.519.11.4008), РФФИ (проект №10-01-00826-а), программы фундаментальных исследований ОНИТ РАН, проектов Евросоюза SecFutur и MASSIF и ряда других проектов.

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. Огарок А.Л., Комашинский Д.В., Школьников Д.К., Мартыненко В.В. Виртуальные войны. Искусственный интеллект защищает от вирусов и программных закладок // Защита информации. Конфидент, № 2, 2003, С.64-69.
3. Blonce A., Filiol E., Frayssignes L. Protable Document Format (PDF) Security Analysis and Malware Threats // Presentations of Europe BlackHat 2008 Conference, 2008.
4. Cova M., Kruegel C., Vigna G. Detection and Analysis of Drive-by-Download Attacks and Malicious JavaScript Code // Proceedings of the 19th international conference on World Wide Web, 2010. P.281-290.
5. Edwards S., Baccas P. Fast Fingerprinting of OLE2 Files: Heuristics for Detection of Exploited OLE2 Files based on Specification Non-conformance // Proceeding of Virus Bulletin Annual Conference, Barcelona, October 2011. P.172-185.
6. Itabashi K. Portable Document Format Malware // Symantec Security Response Whitepapers, http://www.symantec.com /content/en/us/enterprise/media/security_response/whitepapers/ portable_document_format_malware.pdf.
7. Komashinskiy D., Kotenko I. 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 (PDP 2010). Pisa, Italy, 17-19 February, 2010. Los Alamitos, California. IEEE Computer Society. 2010. P.617-623. ISSN 1066-6192.
8. Kubec J., Sejtko J. X IS NOT ENOUGH! GRAB THE PDF BY THE TAIL! // Proceeding of Virus Bulletin Annual Conference, Barselona, October 2011. P.128-135.

9. Laskov P., Srndic N. Static Detection of Malicious JavaScript-Bearing PDF Documents // Proceedings ACSAC'11 Proceedings of the 27th Annual Computer Security Applications Conference, 2010. P.373-382.
10. Li W.-J., Stolfo S. SPARSE: A Hybrid System to Detect Malcode-Bearing Documents CU Tech. Report, Jan 2008 https://mice.cs.columbia.edu/getTechreport.php?techreportID=504.
11. Polychronakis M., Anagnostakis K., Markatos E. Comprehensive shellcode detection using runtime heuristics // Proceeding ACSAC '10 Proceedings of the 26th Annual Computer Security Applications Conference, 2010. P.287-296.
12. Rahman M. Getting Owned By Malicious PDF - Analysis // SANS Institute Reading Room Site, http://www.sans.org/reading_room/whitepapers/malicious/owned-malicious-pdf-analysis_33443.
13. Sadeghi A.-R., Davi L. Runtime Attacks: Buffer Overflow and Return-Oriented Programming // Course Secure, Trusted and Trustworthy Computing, Part 1, 2011. http://goo.gl/L8VRP.
14. Serna F. Exploits and Mitigations: EMET. http://ivanlef0u.fr/repo/expl0it/ferminjserna-exploitsmitigationsemet-100328034335-phpapp02.pdf.
15. Tyugu E. Artificial Intelligence in Cyber Defense // 2011 3rd International Conference on Cyber Conflict. C.Czosseck, E.Tyugu, T.Wingfield (Eds.) Tallinn, Estonia, 2011. P.95-105. ISBN 978-9949-9040-2-0.
16. Tzermias Z., Sykiotakis G., Polychronakis M., Markatos E. Combining Static and Dynamic Analysis for the Detection of Malicious Documents // Proceedings of the Fourth European Workshop on System Security, ACM New York, 2011.
17. Wolf J. OMG WTF PDF // Presentation for 27th Chaos Communication Congress, 2010. http://blog.fireeye.com/files/27c3_julia_wolf_omg-wtf-pdf.pdf.
18. A Generic JavaScript Unpacker. http://code.google.com/p/jsunpack-n/.
19. Acrobat Reader download page. http://get.adobe.com/reader/.
20. CVE List, CVE-2011-1983. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-1983.
21. International Organization for Standardization, Portable Document Format, ISO 32000-1:2008, http://wwwimages.adobe.com/www.adobe.com/content/dam/Adobe/en/devnet /pdf/pdfs/PDF32000_2008.pdf.
22. Microsoft Security Bulletin Summary for December 2011. http://technet.microsoft.com/en-us/security/bulletin/ms11-dec.
23. Microsoft Support, The Enhanced Mitigation Experience Toolkit. http://support.microsoft.com/kb/2458544.
24. Open PDF Analysis Framework. http://code.google.com/p/opaf/.
25. Rapid – I Rapid Miner 5. http://rapid-i.com/content/view/181/190/.
26. Security Advisory for Adobe Reader and Acrobat. http://www.adobe.com/support/security/advisories/apsa11-04.html.
27. USA National Institute of Standards and Technology, National Vulnerability Database, CVE and CCE Statistics Query Page. http://web.nvd.nist.gov/view/vuln/statistics.
28. VXHeavens. http://www.vxheavens.com.
29. Комашинский Д.В., Котенко И.В. Детектирование вредоносного программного обеспечения на основе обработки статической информации методами интеллектуального анализа данных // Управление рисками и безопасностью. Труды Института системного анализа Российской академии наук (ИСА РАН). Т.51. М.: ИСА РАН, 2009. С.65-84.
30. Комашинский Д.В., Котенко И.В., Шоров А.В. Подход к обнаружению вредоносного программного обеспечения на основе позиционно-зависимой информации // Труды СПИИРАН, Выпуск 10. СПб.: Наука, 2010. С.144-159.
31. Комашинский Д.В., Котенко И.В., Чечулин А.А. Категорирование веб-сайтов для блокирования веб страниц с неприемлемым содержимым // Системы высокой доступности, № 2, 2011. С.102-106.

Комашинский Дмитрий Владимирович. Аспирант Санкт-Петербургского института информатики и автоматизации РАН. Окончил Санкт-Петербургское высшее военное инженерное училище связи им. Ленсовета в 2000 году. Автор 20 печатных работ. Область научных интересов: методы выявления и анализа вредоносного программного обеспечения, искуственный интеллект, форенсика и анализ компьютерных преступлений. E-mail: komashinskiy@comsec.spb.ru

Котенко Игорь Витальевич. Заведующий лабораторией Санкт-Петербургского института информатики и автоматизации РАН. Доктор технических наук, профессор. Окончил Военный инженерный институт им. А.Ф.Можайского и Военную академию связи. Автор более 450 печатных работ. Область научных интересов: защита информации, искусственный интеллект, телекоммуникационные системы. E-mail: ivkote@ comsec.spb.ru