УДК 004.66 + 004.056 Вестник УрФО № 3(25) / 2017, с. 12–18

Бухарев Д. А., Вагин С. В., Соколов А. Н.

ОБНАРУЖЕНИЕ ВРЕДОНОСНОГО МОБИЛЬНОГО КОДА НА ОСНОВЕ ЭВРИСТИЧЕСКОГО АНАЛИЗА PDF-ФАЙЛОВ

Описаны методы классификации вредоносных PDF файлов и способы обнаружения мобильных кодов путем статистического анализа. PDF формат файлов позволяет встроить вредоносный контент любого типа вне зависимости от используемого браузера и операционной системы. По этой причине количество угроз этого типа непрерывно растет. Проанализирована выборка из 110000 PDF файлов, часть из которых была заражена. По результатам анализа предложены эффективные способы обнаружения вредоносных мобильных кодов.

Ключевые слова: *Формат PDF, мобильный код, JavaScript, вредоносная программа.*

Bukharev D. A., Vagin S. V., Sokolov A. N.

DETECTION OF MALICIOUS MOBILE CODE BASED ON HEURISTIC ANALYSIS OF PDF FILES

Methods of classification of malicious PDF files and ways of detecting mobile codes by the means of statistical analysis described. PDF format allows us to embed malicious content of any type regardless of browser or operation system used. For this reason, number of threats of this type is increasing constantly. Corpus of 110000 PDF files analyzed, some of them were infected. On the basis of the analysis, effective methods of detecting malicious mobile codes suggested.

Keywords: *PDF format, mobile code, JavaScript, malicious software.*

PDF формат существует с 1993 года [1]. До 2004 года он не использовался злоумышленниками как контейнер для распространения вредоносных программ в отличие от форматов, использующих макросы (MS Word, MS Excell). Поэтому во второй половине 1990-х годов PDF формат рекомендовался исследователями как наиболее надежный формат текстовых документов.

С появлением вредоносного ПО (червя) Peachy [2] отношение к PDF формату начало меняться, начались исследования безопасности формата. Peach является классической вредоносной скрипт-программой, написанной на языке Visual Basic Script (VBS). Кроме этого, существует возможность внедрения в PDF-файлы программного кода, написанного на языке JavaScript и другого мобильного кода [3]. С появлением другого червя – Yourde [4], который массово заражал PDF файлы, исследователи в области компьютерной безопасности PDF формату стали уделять большее внимание.

С момента появления PDF формат претерпел множество изменений (с частотой 1 раз в 18 месяцев [5]), позволяющих разработ-

чикам лучше контролировать процесс просмотра. Однако при проведении синтаксического анализа PDF файлов могут возникать проблемы эффективности анализа их структуры (рис. 1).

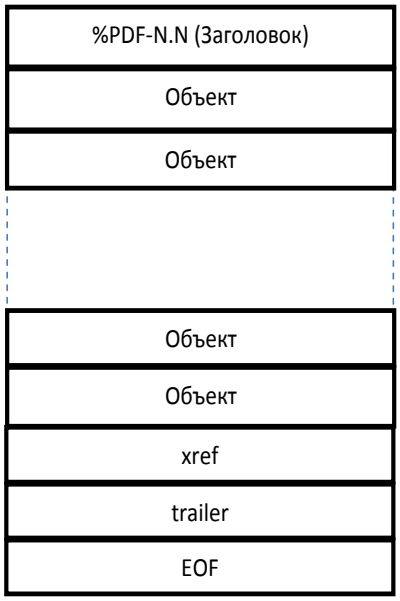

Рис. 1. Структура PDF файла

Заголовок PDF файла обычно находится в его начале, но это необязательное требование. Далее в файле PDF обычно находятся объекты, которые начинаются со строки «N R obj», где N – номер объекта, R – номер ревизии (обычно 0), объект завершается маркером «endobj». Внутри объекта может содержаться сжатый поток данных, который обозначается маркерами начала и конца потока. Таблица перекрестных ссылок (xref) содержит список объектов, их положение и состояние в файле. В файле может находиться несколько xref таблиц. В новом (немодифицированном) PDF файле будет содержаться лишь одна таблица xref, аналогичная приведенной (табл. 1).

Первая запись сообщает, что в таблице содержатся ссылки на 5 объектов, начиная с объекта под номером 0. Нулевой объект – это специальный объект, который имеет смещение в 0 байт, номер ревизии 65535 (-1), а также не используется (флаг f). В остальной части этой таблицы указано, что в файле содержатся четыре объекта со смещениями 10, 120, 189 и 408 байт. Все объекты заданы с номером ревизии 0 и находятся в использовании (флаг n). Такой способ определения объектов может означать, что все объекты уникальны, однако это не обязательно.

Объект типа trailer говорит анализатору как считывать таблицы перекрестных ссылок. В этом объекте обязательно содержится информация о количестве всех объектов в документе (ключ /Size), а также ссылка на корневой объект (ключ /Root). После объекта trailer следуют метка «startxref», величина смещения последней таблицы перекрестных ссылок, а затем метка %%EOF, которая означает «конец файла». Таблица перекрестных ссылок (xref) позволяет разработчикам PDF-анализаторов быстро создавать таблицы позиций объектов в отличие от простого перебора самого файла (что не всегда эффективно).

В рамках исследования была произведена выборка из 110000 PDF файлов, взятых из базы данных SophosLabs с 2014 года. Анализ этих файлов с помощью сканеров Kaspersky, McAfee, Microsoft и Sophos показал, что примерно 70000 из них являются подозрительными или содержат вредоносный мобильный код.

Результаты сканирования файлов введены в базу данных SQLite для последующего их анализа. Собранные данные не являются полными, так как процедура анализа таблицы xref оказалась трудоемкой с точки зрения временных затрат.

За последние несколько лет количество

Таблица 1

Таблица перекрестных ссылок (xref)

	Смещение, байт	**Номер ревизии**	**Флаг использования**
Первая запись	0	5	–
Объект 0	0	65535	f
Объект 1	10	0	n
Объект 2	120	0	n
Объект 3	189	0	n
Объект 4	408	0	n

вредоносных образцов PDF файлов значительно увеличилось. Для иллюстрации были запрошены базы данных SophosLabs по времени поступления PDF файлов, помеченных как вредоносные путем статического сканирования. Динамика роста количества зараженных файлов приведена на рис. 2.

Представленный график отображает натуральный логарифм общего числа вредоносных образцов в зависимости от времени. Во второй части графика, начиная с 2015 года, кривая принимает линейный характер. Если экстраполировать прямую часть представленного графика влево до оси x, то получим дату начала появления зараженных PDF файлов – март 2003 года (появление червя Yourde).

Чтобы показать, что размер PDF файла не является критерием наличия вредоносного мобильного кода, рассмотрим гистограмму из 60000 файлов, имеющих наименьший размер (рис. 3). Файлы сгруппированы в блоки по 512 байт (0-512 байт, 512-1024 байт и т. д.).

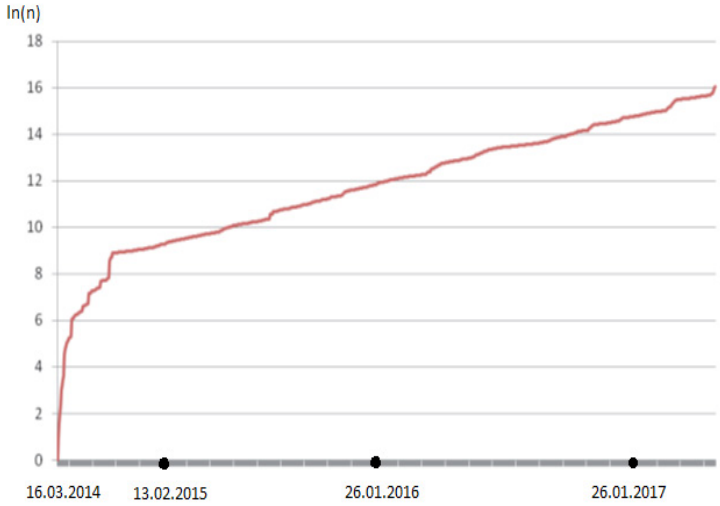

Рис. 2. Динамика роста вредоносных образцов PDF файлов (n – количество вредоносных образцов)

График показывает распределение файлов размером менее 20 КБ: красная полоса – это все файлы (54.05% выборки), а зеленая – только те, которые помечены как вредоносные (83.46%).

Объект, содержащий код на языке JavaScript, может косвенно ссылаться на дру-

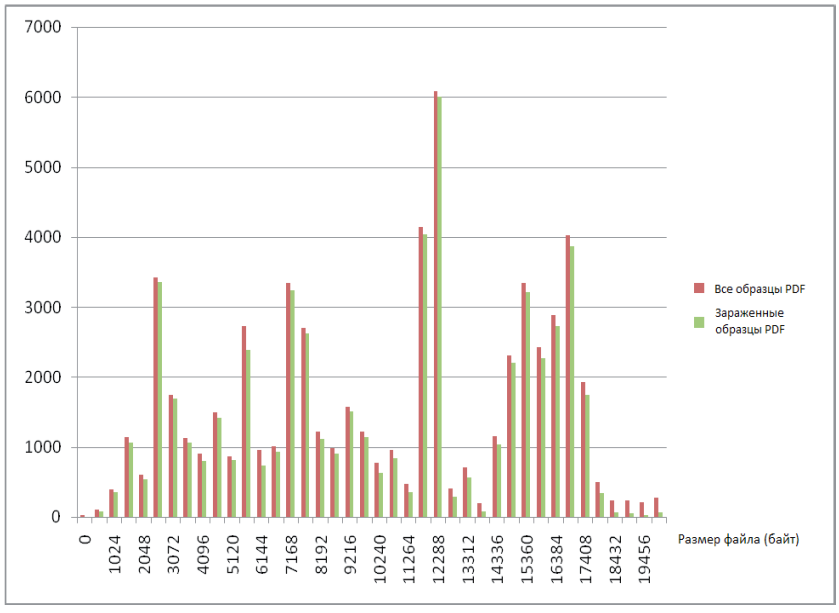

Рис.3. Распределение файлов размером менее 20 КБ

гой объект, содержащий тег /JavaScript. Это означает, что статическое сканирование тега завершится неудачей. Из результатов анализа выборки файлов на наличие тега /JavaScript, представленных в табл. 2, видно, что подавляющее большинство зараженных PDF файлов в отличие от обычных файлов содержат в себе код JavaScript. Таким образом, поиск в файле тега /JavaScript – хороший способ обнаружения вредоносных файлов. Однако, встраивание кода JavaScript – это не единственный способ заражения PDF файлов.

Таблица 2

Наличие тега/JavaScript в файлах

Всего	Вредоносные		Незараженные	
	К-во	%	К-во	%
64616	63523	98,3	1093	1,7

Во вредоносном файле число объектов и потоков может не совпадать с описанным в специальных объектах файла. Сопоставления меток «N R obj» – «endobj» и «stream» – «endstream» должно быть достаточно, чтобы получить истинное число объектов и потоков. В табл. 3 показано, что при выборке из 110000 файлов 10321 файлов имеют несоответствующие объекты, а 2296 – несоответствующие потоки. Процент несовпадений меток во вредоносных файлах значительно выше (84,1% и 69%), чем у незараженных файлов (15,9% и 31%).

Таблица 3

Несовпадающие объекты и потоки в файлах

	Несоответствующие объекты		Несоответствующие потоки	
	К-во	%	К-во	%
Всего файлов	10321	100	2296	100
Вредоносные	8685	84,1	1585	69,0
Незараженные	1636	15,9	711	31,0

В табл. 4 представлена статистика файлов, содержащих уникальные объекты, в выборке из 110000 PDF файлов. Незараженные файлы содержат больше неуникальных объектов по сравнению с вредоносными.

Анализ таблиц перекрестных ссылок (xref) может указывать на то, какой файл может быть заражен: таблица перекрестных ссылок у вредоносных файлов будет иметь недопустимый формат чаще, чем у незараженных файлов (табл. 5).

Таблица 4

Количество уникальных объектов в файлах

	Уникальные объекты		Неуникальные объекты	
	К-во	%	К-во	%
Всего файлов	104749	100	7746	100
Вредоносные	64425	61,7	3078	39,7
Незараженные	40324	38,3	4668	60,3

Таблица 5

Статистика нарушений таблицы перекрестных ссылок (xref)

Типы нарушений таблицы xref	PDF без startxref		Недопустимый xref		xref не сканируется	
	К-во	%	К-во	%	К-во	%
Всего файлов	5506	100	6691	100	91530	100
Вредоносные	5373	97,6	6123	91,5	56114	61,3
Незараженные	133	2,4	568	8,5	35416	38,7

Злоумышленник может закодировать вредоносный контент внутри объекта с помощью фильтров (табл. 6).

Фильтры №№ 1 – 5 обычно используются для всех типов данных, фильтры №№ 6 – 8 – для изображений и носителей, фильтры №№ 9 – 10 используются Adobe для защиты данных. Статистика распределения фильтров внутри выборки из 110000 файлов показаны в табл. 7.

Формат PDF позволяет записывать в файл символы, отличные от ASCII и не запрещает использование их шестнадцатеричного представления. Например, последовательность символов обфусцированного тега /Filter выглядит как:

Таблица 6

Типы кодирования внутри фильтров

№ п/п	Тип кодирования	Имя в фильтре
1.	Flate	FlateDecode
2.	LZW	LZWDecode
3.	ASCII85	ASCII85Decode
4.	ASCIIHex	ASCIIHexDecode
5.	RunLength	RunLengthDecode
6.	JBIG2 J	JBIG2Decode
7.	RichMedia	RichMedia
8.	DCT	DCTDecode
9.	Crypt	Crypt
10.	Encrypt	Encrypt

держать в себе другие теги, которые используются для запуска мобильного кода:

• OpenAction – определяет, что должна делать программа для просмотра PDF файлов при открытии документа;

• Launch [6] – позволяет запускать медиа объекты и скрипты при открытии PDF файла;

• «AA» (additional action) – определяет действия в ответ на «событие триггера» внутри объектов;

• Acroform – интерактивная форма, которая используется вредоносными PDF-файлами, где она переопределяет объект как исполняемый JavaScript код.

Год от года количество обнаруженных заражений PDF файлов растет. Сложность формата файла и его способность запускать мобильный код означает, что эта тенденция, ско-

Таблица 7

Статистика распределения фильтров в файлах

Фильтр	Всего	Вредоносные		Незараженные	
		К-во	%	К-во	%
FlateDecode	92277	52342	56,7	39935	43,3
LZWDecode	5907	319	5,0	5588	95,0
ASCII85Decode	6174	841	13,6	5333	86,4
ASCIIHexDecode	6451	6404	99,3	47	0,7
RunLengthDecode	371	159	42,9	212	57,1
JBIG2Decode	689	525	76,2	164	23,8
RichMedia	178	178	100	0	0
DCTDecode	20666	1404	6,8	19262	93,2
Crypt	234	28	11,9	206	88,1
Encrypt	5478	614	11,2	4864	88,8

Filter [/A#53#43#49#49#48#65#78Dec#6f#64e /#4c#5aW#44#65c#6fde/#41S#43I#4985Decod#65/ R#75n#4ce#6e#67#74#68D#65#63o#64e/FlateDecode]

Обфусцированный тег декодируется средствами просмотра PDF файлов в последовательность символов:

Filter [/ASCIIHexDecode /LZWDecode / ASCII85Decode

RunLengthDecode /FlateDecode]

Этот пример демонстрирует, что значение тега /Filter может быть обфусцировано с целью скрытия в нем фильтров, которые могут использоваться во вредоносных файлах. Для поиска обфусцированных тегов можно использовать регулярное выражение:

Filter\s+\[\s+[^\]]+#

Вредоносные PDF файлы могут также сорее всего, будет развиваться. Полученная статистика позволяет сделать заключение о признаках заражения PDF файла. Чтобы файл отнести к подозрительным, необходимо выполнение одного из условий:

• наличие тега /JavaScript;

• несовпадение меток потоков и объектов в файле;

• содержит больше неуникальных объектов;

• содержит любой из видов нарушений таблицы перекрестных ссылок (xref)

• содержит фильтры FlateDecode и ASCIIHexDecode.

Таким образом, для ограничения возможностей запуска вредоносного мобильного кода, содержащегося в PDF файле, рекомен-

дуется при использовании средств просмотра:

• запускать только подписанный JavaScript код (и другие функции);

• запускать только подписанные встроенные файлы;

• отключать JavaScript в плагинах браузера;

• реализовывать строгий режим синтаксического анализа, например, чтобы не допустить запутывания фильтров.

Статья выполнена при поддержке Правительства РФ (Постановление №211 от 16.03.2013 г.), соглашение № 02.A03.21.0011.

Примечания

1. История формата PDF (сайт компании Adobe) [Электронный ресурс] //http://www.adobe.com/pdf/about/history/. (Дата обращения: 13.10.2017).

2. История вредоносного контента встраиваемого в PDF файлы (сайт компании McAfee) [Электронный ресурс] //http://vil.nai.com/vil/content/v_99179.htm. (дата обращения: 13.10.2017).

3. Сети промышленной коммуникации. Безопасность сетей и систем. Часть 3-3. Требования к системной безопасности и уровни безопасности [Текст]: ГОСТ Р МЭК 62443-3-3 – 2016. – Введ. 2017–04–01. – М. : Стандартинформ, 2016.

4. Анализ вредоносного контента в PDF файлах (сайт компании Sophos) [Электронный ресурс] // http://www.sophos.com/security/analyses/viruses-andspyware/jsyourdea.html#table4. (дата обращения: 13.10.2017).

5. История формата PDF. Блог Леонарда Орра [Электронный ресурс] // http://www.acrobatusers.com/blogs/leonardr/history-of-pdfopenness/.(дата обращения: 15.10.2017).

6. Запуск вредоносного контента из PDF файлов (сайт компании Sophos) [Электронный ресурс] // http://www.sophos.com/blogs/sophoslabs/?p=9301. (дата обращения: 13.10.2017).

References

1. Istoriya formata PDF (sayt kompanii Adobe) [Elektronnyy resurs] //http://www.adobe.com/pdf/about/history/. (Data obrashcheniya: 13.10.2017).

2. Istoriya vredonosnogo kontenta vstraivayemogo v PDF fayly (sayt kompanii McAfee) [Elektronnyy resurs] //http://vil.nai.com/vil/content/v_99179.htm. (data obrashcheniya: 13.10.2017).

3. Seti promyshlennoy kommunikatsii. Bezopasnost' setey i sistem. Chast' 3-3. Trebovaniya k sistemnoy bezopasnosti i urovni bezopasnosti [Tekst]: GOST R MEK 62443-3-3 – 2016. – Vved. 2017–04–01. – M. : Standartinform, 2016.

4. Analiz vredonosnogo kontenta v PDF faylakh (sayt kompanii Sophos) [Elektronnyy resurs] //http://www.sophos.com/security/analyses/viruses-andspyware/

jsyourdea.html#table4. (data obrashcheniya: 13.10.2017).

5. Istoriya formata PDF. Blog Leonarda Orra [Elektronnyy resurs] // http://www.acrobatusers.com/blogs/leonardr/history-of-pdfopenness/.(data obrashcheniya: 15.10.2017).

6. Zapusk vredonosnogo kontenta iz PDF faylov (sayt kompanii Sophos) [Elektronnyy resurs] // http://www.sophos.com/blogs/sophoslabs/?p=9301. (data obrashcheniya: 13.10.2017).

БУХАРЕВ Дмитрий Александрович, студент кафедры защиты информации высшей школы электроники и компьютерных наук ФГАОУ ВО «Южно-Уральский государственный университет (национальный исследовательский университет)». Россия, 454080, г. Челябинск, проспект Ленина, д 76. E-mail: needleinspace@gmail.com

ВАГИН Сергей Владиславович, студент кафедры защиты информации высшей школы электроники и компьютерных наук ФГАОУ ВО «Южно-Уральский государственный университет (национальный исследовательский университет)». Россия, 454080, г. Челябинск, проспект Ленина, д 76. E-mail: vaginserg.96@mail.ru

СОКОЛОВ Александр Николаевич, кандидат технических наук, доцент, заведующий кафедрой защиты информации высшей школы электроники и компьютерных наук ФГАОУ ВО «Южно-Уральский государственный университет (национальный исследовательский университет)». Россия, 454080, г. Челябинск, проспект Ленина, д 76. E-mail: ANSokolov@inbox.ru

BUKHAREV Dmitrii, student of the department of information security of the school of electrical engineering and computer science in FSAEI HE «South Ural State University (national research university) ». 76, Lenin prospect, Chelyabinsk, Russia, 454080. E-mail: needleinspace@gmail.com

VAGIN Sergey, student of the department of information security of the school of electrical engineering and computer science in FSAEI HE «South Ural State University (national research university) ». 76, Lenin prospect, Chelyabinsk, Russia, 454080. E-mail: vaginserg.96@mail.ru

SOKOLOV Alexander, Candidate of Engineering Science, Docent, Head of the department of information security of the school of electrical engineering and computer science in FSAEI HE «South Ural State University (national research university) ». 76, Lenin prospect, Chelyabinsk, Russia, 454080. E-mail: ANSokolov@inbox.ru