УДК 681.31

О.С. Волковский, К.А. Каращенко

ПРИМЕНЕНИЕ ЭВРИСТИЧЕСКОГО АЛГОРИТМА В АНТИВИРУСНЫХ ПРИЛОЖЕНИЯХ ИДЕНТИФИКАЦИИ ВИРУСНЫХ КОНСТРУКЦИЙ В WEB-СТРАНИЦАХ

Аннотация. Рассмотрен эвристический алгоритм поиска потенциально опасных команд скриптовых языков, внедренных в web-страницы. Показано, что существующие методики определения вирусных вставок не совершенны. Создана модель эвристико-логической системы для оценки опасности web-страниц. Проведены исследования и доказана научная ценность предлагаемого метода.

Введение

Одним из наиболее опасных путей, по которым вирусы попадают на пользовательский компьютер, является Internet. Главной особенностью этого проникновения является его прозрачность для пользователя в большинстве случаев. При этом он просто открывает в браузере web-страницу, и в этот момент активизируются встроенные в нее коды, написанные на скриптовых языках программирования, таких как Visual Basic Script, JavaScript, WSH, PHP, ActiveX[1].

Постановка задачи

Действующий с 1999 года стандарт HTML 4.01 предполагает возможность вставки исполняемого кода. Возможности перечисленных выше языков программирования с точки зрения нанесения вреда ОС пользователя достаточно широки. Не обязательно внедрять весь код вируса в страницу, достаточно определить тип ОС, скопировать вирусный файл на жесткий диск и запустить его на выполнение. В этом случае идентификация стандартными средствами антивирусного ПО не представляется возможной[3].

Современное антивирусное ПО использует следующие методы обнаружения вторжений:

1. Метод соответствия определению вирусов в словаре;
2. Метод обнаружения странного поведения программ;
3. Метод обнаружения при помощи эмуляции;

4. Метод «Белого списка».

Все вышеперечисленные методы ориентированы на идеологию сравнения подозрительного файла с шаблонами известных типов вирусов. Недостатки такого подхода обусловлены возможностью модификации вирусных программ. Достаточно изобрести новый вирус, полностью идентичный разработанному ранее, однако использующий другие команды и методы для достижения своих целей, и антивирус не сможет заблокировать его как опасную программу, так как его не будет в списке его сигнатур. Кроме того, обновление сигнатур не всегда настроено у пользователей антивируса и еще менее охотно они дают отчеты о новых типов вирусов, поразивших их компьютеры[4,5].

Другим, потенциально возможным методом борьбы с сетевыми вирусами являются брандмауэры и фаерволы. Однако качественные продукты этой категории достаточно дорогостоящие и требуют больших усилий квалифицированных специалистов для грамотной настройки, а у свободно распространяемых - ошибка идентификации потенциально опасных страниц настолько велика, что зачастую утомляет пользователя своими бессмысленными запретами[2].

В данной статье предлагается принципиально новый метод обнаружения вирусных конструкций в интернет-страницах, который построен на основании эвристического анализа команд скриптовых языков.

Обзор алгоритма работы эвристического анализа

Целью исследований является создание метода, который позволил бы с максимальной точностью отнести web-страницу к категории опасных и настоятельно не рекомендуемых к просмотру, либо, в противном случае – безопасных для системы.

Важной особенностью рассматриваемого подхода является концепция структуры сигнатур скриптовых языков. Подключаемые файлы сигнатур содержат по 2 файла для каждого языка – безопасных и небезопасных команд(exclude и include) со своими коэффициентами опасности, что рассматривается как поиск определенной команды в exclude, а на следующем шаге - include в условных блоках алгоритма (Рис. 1). Преимущество такого подхода обусловлено возможностью добавления новых команд в файл сигнатуры вследствии появления новых версий языка

программирования. При выпуске нового скриптового языка достаточно незначительно модифицировать приложение, создав дополнительную пару сигнатур и обеспечить возможность проверки на эту сигнатуру. Следует отметить, что скорость развития существующих языков и добавления новых к числу интегрируемых в HTML несоизмеримо меньше скорости создания новых версий вирусов.

На содержание потенциально опасных команд анализируются также и участки кода, которые содержатся между установленными тегами включения этого языка в текст HTML. Например, команды языка php не отображаются браузером, если они находятся между тегами

(<? ;?>), (<script language="php"> ; </script>), (<% ; %>).

Пары этих тегов обозначаются в алгоритме как (rightDelimetr; leftDelimetr), что отображает проверка «Последняя пара (rdel, ldel)?» (Рис. 1). Каждый язык имеет несколько таких пар, за исключением ActiveX и WSH. Для них проверка на наличие команд из списка сигнатур осуществляется путем сканирования всего кода JavaScript и VBS.

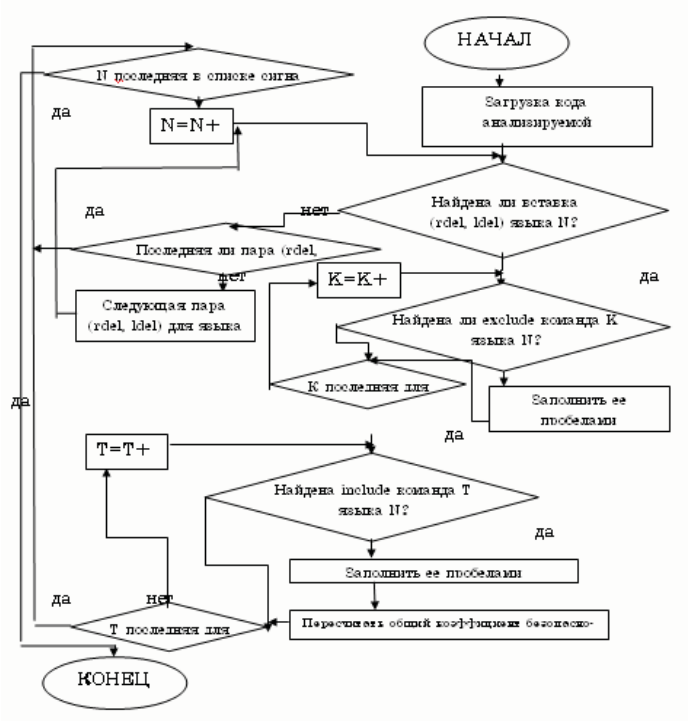

Рисунок 1 – Алгоритм работы программы нахождения потенциально опасных команд методом эвристического анализа

Все потенциально опасные команды скриптовых языков условно можно разделить на группы:

-WWFD (work with file or directory) - команды работы с файлом или папкой в текстовом режиме: писать в файл (1), читать из файла (2), изменять позицию нахождения в файле или работа с указателем на файл(3);

- Sys (System) - команды работы с файлом или папкой с точки зрения ОС: копировать, удалить, открыть файл или папку (1), получить системную информацию о файле или папке (2), создать / удалить указатель на файл, все действия с путем к файлу, получить имя файла или папки (3);

- Right - попытка автоматически изменить права доступа от имени администратора (суперпользователя для Unix-систем), или просто делать такие изменения в работе системы и ее настройках, на которые имеет права только главный пользователь: добавить права на изменения файла или папки, изменить содержимое конфигурационных файлов, изменить главного пользователя файла или папки (1), модификация системного времени и даты, добавить пользователя системы (2), изменить стартовую страницу, добавить в закладки, изменить политику безопасности, и другие настройки браузера (3);

- CNSE (Create new system element) – создать новый системный элемент: создать системный диск, файл или папку (1), переразметить, получить сведения относительно дискового пространства (2), получить информацию о дисках (3);

- Crypth - попытка кодировать / декодировать команды: выполнить команды, предварительно преобразовав из символьных кодов (1), декодировать (2), кодировать (3);

- Post - автоматические попытки отсылать письма от имени пользователя: получить список всех адресов из адресной книги, отослать письмо (1), добавить прикрепленный документ (2), работа с получателями (3);

- SysLink - ссылки на системные ресурсы: *.bat, *.com, *.tar, *.exe-файлы, системные сервисы (1), стандартные системные файлы и папки (2), диски (3);

- MD (Most dangerous) – дополнительная группа команд, которой присвоены коэффициенты $D = 0.99$, $\upsilon = 0.64$: добавить /

удалить сервис, перезагрузить систему, запустить его на исполнение, внести изменения в системный реестр.

Каждой команде из списка сигнатур присваивается свой коэффициент. Команды всех скриптовых языков из сигнатур были разделены на группы команд по функциональным характеристикам. К одной группе относятся однотипные команды в различных языках. Внутри каждой группы команды распределяются на 3 уровня опасности:

1 - команда может нанести серьезный ущерб системе, а выполнение внесет важные изменения в ее работу;

2 - выполнение команды потенциально опасно для ПК, однако не несет необратимых операций;

3 - команда безопасна для системы. Ее присутствие скорее служит иден-тификатором нахождения других вредоносных команд в коде.

Наибольший коэффициент опасности в каждой группе получили те команды, которые чаще всего используются хакерами в вирусных программах. Коэффициенты функциональных групп команд D и частота их появления υ в конструкторах вирусов в виде отношения числа команд группы к общему числу сгенерированных страниц приведены в Табл.1.

Таблица 1

Распределение коэффициентов опасности команд по группам и уровням

Группа	1		2		3	
	D1	υ1	D2	υ2	D3	Y3
WWFD	0.85	>50=1	0.75	23/50=0.46	0.2	8/50=0.16
Sys	0.9	>50=1	0.75	46/50=0.92	0.4	39/50=0.78
Right	0.95	41/50=0.82	0.85	12/50=0.24	0.75	28/50=0.56
CNSE	0.95	>50=1	0.8	9/50=0.18	0.75	13/50=0.26
Crypth	0.9	>50=1	0.85	>50=1	0.4	18/50=0.36
Post	0.85	24/50=0.48	0.8	3/50=0.06	0.55	16/50=0.32
SysLink	0.96	>50=1	0.85	>50=1	0.75	14/50=0.28

Помимо поиска сигнатур опасных команд скриптовых языков осуществляется проверка на содержание в коде языка команд обращения к системным дискам, файлам и папкам,

конфигурационным файлам в системных директориях. Нахождение длинных последовательностей закодированных 16-ричных символов, последовательностей со знаками %, которые использует JavaScript для сокрытия сути шифрованного текста от антивирусного ПО и просмотрщиков кода страниц через АСКИ-коды символов команд [4,5]. Для них приняты коэффициенты $D = 0.6$, $\upsilon = 1$.

Результаты исследований

Были проведены эксперименты по поиску потенциально опасных включений в тестовых web-страницах (Табл. 2). На основе предложенного алгоритма реализованы методы расчета общего коэффициента опасности страницы - арифметического среднего и наивного классификатора Байеса.

Проведенные исследования показали, что метод арифметического среднего более чувствителен к потенциально опасным включениям, чем метод Байеса. Страницы, инфицированные конструкторами вирусов, с большей достоверностью определяются на основе эвристического метода, чем известными антивирусами. При анализе двух других групп страниц антивирус Касперского не обнаруживает ни одной инфицированной, Avira – 5. Возможно, эти страницы действительно не содержат опасных включений, однако имеют возможность их сохранения на ПК, выполняя команды создания файлов без ведома пользователя, получая информацию о настройках его системы, протоколах обмена данными и пр.

Таблица 2

Сравнение оценок результатов сканирования web-страниц с использованием эвристического алгоритма и известных антивирусов

Группа web-страниц	Эвристический алгоритм, метод ар. среднего	Эвристический алгоритм, метод Байеса	Антивирус Касперского	Avira AntiVir
Инфицированные конструкторами	24/25=0,96	18/25=0,72	18/25=0,72	14/25=0,56
Черные списки	4/25=0,16	4/25=0,16	0	5/25=0,2
Лидеры посещаемости	19/50=0,38	17/50=0,34	0	0

Выводы

Предложенный эвристический метод в результате исследований показал хорошие результаты определения потенциально опасных команд в web-страницах. Основными преимуществами метода следует считать возможность быстрой адаптации к новым опасным включениям, точность идентификации, а также свободный выбор метода дальнейшего определения полного коэффициента опасности страницы, который рассчитывается на основе коэффициентов найденных команд для каждой из сигнатур.

ЛИТЕРАТУРА

1. Фленов М.Е. «Web-сервер глазами хакера» – Санкт Петербург «БХВ-Петербург», 2007г.
2. Скембрей Дж., Шема М. «Секреты хакеров. Безопасность Web-приложений - готовые решения», Москва, Издательский дом «Вильямс», 2003г. -384 с.
3. Джонс К., Шема М., Джонсон Б. «Антихакер. Средства защиты компьютерных сетей. Справочник профессионала», «Издательский дом Вильямс», 2003 г. -458 с.
4. Ludwig M. «The Giant Black Book of Computer Viruses», American Eagle Publications, Inc. 1998- 490 p.
5. Ludwig M. «The Little Black Book of Computer Viruses», American Eagle Publications, Inc. 1993- 384 p.
6. Фленов М.Е. «Linux глазами хаккера" – Санкт Петербург «БХВ-Петербург», 2005г.