МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра теоретических основ компьютерной безопасности и криптографии

Автоматизированный поиск новых вредоносных программ

АВТОРЕФЕРАТ
дипломной работы

студента 6 курса 631 группы
специальности 10.05.01 Компьютерная безопасность
факультета компьютерных наук и информационных технологий
Луконина Ильи Игоревича

Научный руководитель
доцент _____ И. Ю. Юрин
22.01.2022 г.

Заведующий кафедрой
д. ф.-м. н., доцент _____ М. Б. Абросимов
22.01.2022 г.

Саратов 2022

ВВЕДЕНИЕ

Некоторые из самых распространенных мифов в компьютерной сфере связаны с вирусами и антивирусными технологиями. Широко распространено мнение, что антивирусное программное обеспечение может обнаруживать только определенные известные вирусы существует с первых дней исследования вирусов. Тогда это было не совсем так. Некоторые из первых антивирусных программ не были предназначены для обнаружения конкретных вирусов, а, скорее, для обнаружения или блокирования вирусоподобного поведение или подозрительных изменений в файлах. И сейчас это определенно неправда.

Коммерческие антивирусные системы дополняют сканирование сигнатур множеством более общих подходов, которые часто объединяются под вывеской эвристического анализа. Более того, большинство современных антивирусных продуктов способны обнаруживать широкий спектр вредоносных программ, а не только вирусы. Они могут сочетаться с другими технологиями безопасности, такими как обнаружение спама и фишинговых сообщений.

Цель дипломной работы – уменьшить некоторую путаницу вокруг работы антивирусных технологий и прояснить, чего можно ожидать от антивирусной защиты, в частности, с помощью эвристического анализа.

Для достижения поставленной цели требуется решить следующие задачи:
1) изучить необходимые положения антивирусных технологий;
2) подробно рассмотреть один из алгоритмов эвристического анализа программного кода.

Дипломная работа состоит из введения, 5 разделов, заключения, списка использованных источников и 1 приложения. Общий объем работы – 61 страница, из них страница 41 – основное содержание, включая 12 рисунков и 0 таблиц, список использованных источников из 20 наименований.

КРАТКОЕ СОДЕРЖАНИЕ

Первый раздел дипломной работы «Необходимые положения антивирусных технологий» содержит описание основных понятий, связанные с антивирусными технологиями.

Вредоносное ПО – это любое программное обеспечение, намеренно разработанное для нарушения работы компьютера, сервера, клиента или компьютерной сети, утечки личной информации, получения несанкционированного доступа к информации или системам, лишения доступа пользователей к информации или которое неосознанно нарушает безопасность и конфиденциальность компьютера пользователя.

Компьютерный вирус – вид вредоносных программ, способных внедряться в код других программ, системные области памяти, загрузочные секторы и распространять свои копии по разнообразным каналам связи.

Антивирусное ПО или *антивирус* – это компьютерная программа, используемая для предотвращения, обнаружения и удаления вредоносных программ.

Хотя существует множество определений вируса, большинство исследователей вредоносных программ принимают как компьютерную программу, которая может заражать другие компьютерные программы, модифицируя их таким образом, чтобы включать в себя (возможно, эволюционировавшую) свою копию. Это определение охватывает многие типы вирусов, в том числе:

- Инфекторы загрузочного сектора или сектора раздела;
- Файловые инфекторы (вирусы-паразиты);
- Многокомпонентные вирусы;
- Макровирусы и скриптовые вирусы.

Индустрия антивирусов никогда не приходила к единому мнению относительно того, являются ли *черви*, особым случаем вируса, но в любом случае антивирусное программное обеспечение обычно их обнаруживает. Существует по крайней мере столько же определений червя, сколько и вирусов,

но большинство исследователей антивирусного ПО определяют червя как программу, которая реплицируется не паразитически, то есть не прикрепляется к файлу хоста. Массовые почтовые программы можно охарактеризовать как особый вид червей. Большинство антивирусных компаний описывают этот тип вредоносных программ, передаваемых по электронной почте, как червя, но некоторые почтовые программы и программы массовой рассылки имеют характеристики вируса.

Самым известным нерепликативным вредоносным ПО является *троянский конь* (или для краткости троян). Троян – это программа, которая утверждает, что выполняет некоторую желаемую или необходимую функцию и может даже делать это, но также выполняет некоторые функции или функции, которые человек, запускающий программу, не ожидал и не хотел бы. Это охватывает ряд специализированных вредоносных программ, в том числе:

- Дропперы
- Кейлоггеры
- Деструктивные трояны
- Загрузчики
- Шпионское ПО
- Рекламное ПО
- Руткиты и стелскиты
- Зомби (боты, трояны удаленного доступа, DDoS-агенты)

Второй раздел работы посвящен методам обнаружения вредоносных программ: сигнатурный, поведенческий и эвристический.

Методы обнаружения вредоносных программ по сути делятся на разные категории с разных точек зрения. В этой работе выделено три категории методов обнаружения вредоносных программ, показанных на рисунке 1.

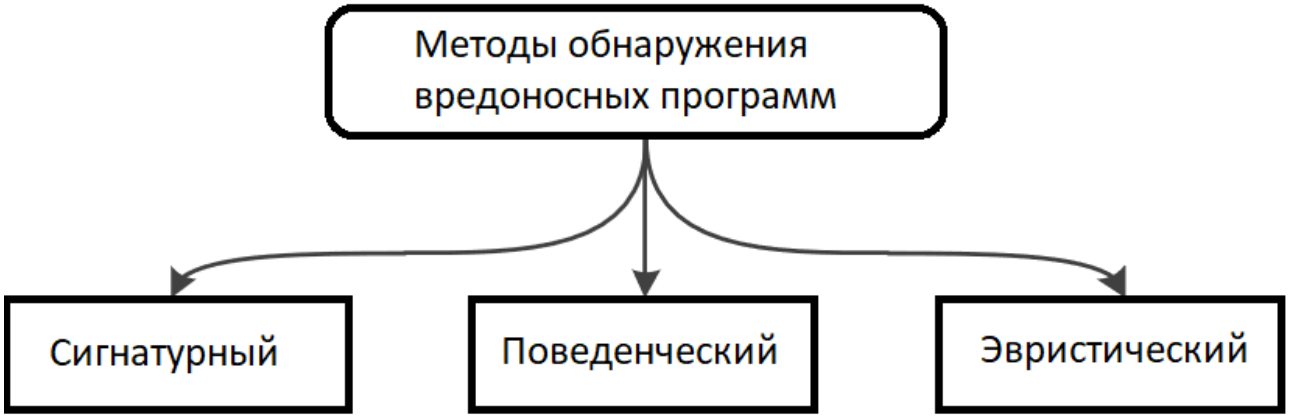

Рисунок 1 – Методы обнаружения вредоносных программ

Для начала рассмотрим *сигнатурный метод*. В настоящее время сопоставление с образцом является наиболее распространенным методом обнаружения вредоносных программ, а обнаружение на основе *сигнатур* – наиболее популярным методом в этой области. Сигнатура – это уникальная функция для каждого файла, что-то вроде отпечатка пальца исполняемого файла. Сигнатурные методы используют шаблоны, извлеченные из различных вредоносных программ, для их идентификации и являются более эффективными и быстрыми, чем любые другие методы. Эти сигнатуры часто извлекаются с особой чувствительностью, чтобы быть уникальными, поэтому методы обнаружения, использующие эту сигнатуру, имеют небольшую частоту ошибок.

Причина, по которой этот прием используется в большинстве коммерческих антивирусов – это небольшая частота ошибок. Эти методы не могут обнаружить неизвестные варианты вредоносных программ, а также требуют большого количества рабочей силы, времени и денег для извлечения уникальных сигнатур. Это основные недостатки этих методов. Кроме того, невозможность противостоять вредоносным программам, которые видоизменяют свои коды при каждом заражении, такие как полиморфные и метаморфические – еще один недостаток.

Методы обнаружения вредоносных программ на основе *поведения* наблюдают за поведением программы, чтобы определить, является ли она вредоносной. Поскольку методы, основанные на поведении, наблюдают за тем, что делает исполняемый файл, они не подвержены недостаткам методов,

основанных на сигнатуре. Проще говоря, детектор на основе поведения определяет, является ли программа вредоносной, проверяя то, что она делает, а не то, что говорит. В этих методах собраны программы с одинаковым поведением. Таким образом, с помощью одной сигнатуры поведения можно идентифицировать различные образцы вредоносного ПО.

Основным преимуществом методов обнаружения вредоносных программ на основе поведения является способность обнаруживать типы вредоносных программ, которые методы сигнатурной базы не могут обнаружить, например, неизвестные и полиморфные варианты вредоносных программ. С другой стороны, отсутствие многообещающего *коэффициента ложных срабатываний* (False Positive Ratio), а также большое время сканирования являются основными недостатками этих методов обнаружения вредоносных программ, основанных на поведении.

В третьем разделе работы подробно рассмотрены эвристические методы обнаружения вредоносных программ и основные функции, которые используют эти методы.

Эвристический относится к действию или процессу поиска или открытия. *Эвристический анализ* – метод обнаружения вредоносных программ, при котором антивирусная программа контролирует все действия, выполняемые проверяемой программой. В ходе эвристического анализа отслеживаются потенциально опасные действия, характерные для вирусов и вредоносных программ других типов. Словарь Вебстера определяет его как помощь в обучении, открытии или решении проблем с помощью экспериментальных методов и особенно методов проб и ошибок или (опять же, в контексте вычислений) относящихся к исследовательским методам решения проблем, в которых используется самооценка – методы обучения (например, оценка обратной связи) для повышения производительности.

Эвристический анализ использует основанный на правилах подход к диагностике потенциально опасного файла.

Поскольку механизм анализатора работает через свою базу правил, проверяя сообщение на соответствие критериям, указывающим на возможное вредоносное ПО, он присваивает баллы при обнаружении совпадения. Если оценка соответствует или превышает пороговую оценку, файл помечается как подозрительный (или потенциально вредоносный или запоминающийся) и обрабатывается соответствующим образом.

Некоторые функции эвристического метода, исследуемые на предмет обнаружения вредоносных программ, изображенные на рисунке 2, будут описаны ниже.

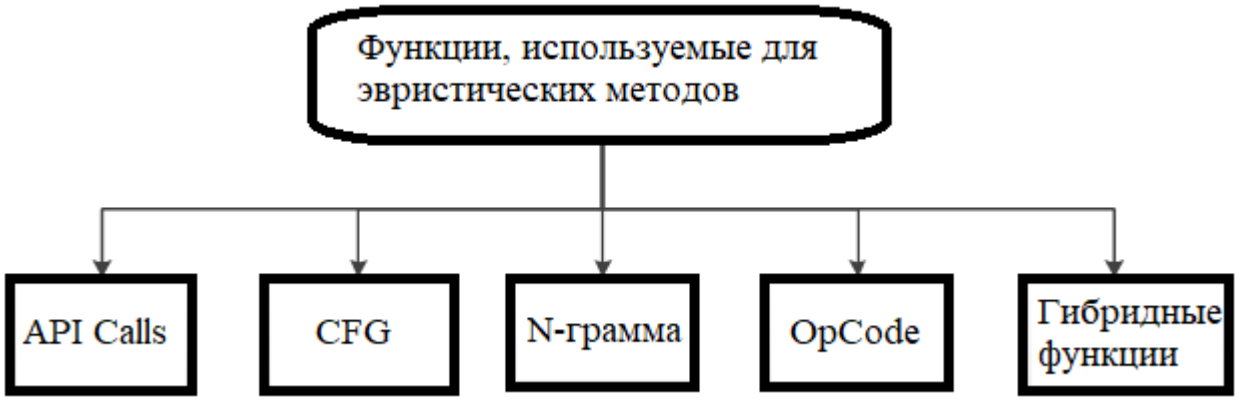

Рисунок 2 – Функции, используемые для эвристических методов

API Calls (API вызовы). Почти все программы используют вызовы интерфейса прикладного программирования (API) для отправки своих запросов в операционную систему. Последовательности вызовов API – это один из наиболее привлекательных способов отражения поведения фрагмента кода, такого как вредоносное ПО.

OpCode (сокращение от Operational Code) – это часть инструкции машинного языка, которая идентифицирует операцию, которая должна быть выполнена. Более конкретно, программа определяется как серия упорядоченных инструкций по сборке. *Инструкция* – это пара, состоящая из операционного кода и операнда или списка операндов.

N-граммы – это все подстроки большей строки длиной N . Например, строку «ВИРУС» можно разделить на несколько 3-граммов: «ВИР», «ИРУ», «РУС» и так далее. За последнее десятилетие было проведено несколько

исследований по обнаружению неизвестного вредоносного ПО на основе его двоичного кода.

Граф потока управления (CFG) – это граф, который представляет поток управления программами, широко используется при анализе программного обеспечения и изучается в течение многих лет. CFG – это ориентированный граф, где каждый узел представляет оператор программы, а каждое ребро представляет поток управления между операторами. Выражения могут быть присваиваниями, операторами копирования, ветвями.

С кодом выполняется ряд операций нормализации после дизассемблирования исполняемой программы для уменьшения эффектов методов мутации и выявления потоковых связей между доброкачественным и вредоносным кодом. Затем генерируется соответствующий CFG для программы. CFG сравнивают с CFG нормализованной вредоносной программы, чтобы узнать, содержит ли CFG подграф, изоморфный CFG нормализованного. Таким образом, проблема обнаружения вредоносных программ заменяется проблемой изоморфизма подграфов.

В четвертом разделе описан формат PE-файла. Рассмотрены основные заголовки и разделы, их структуры и наиболее уязвимые места.

Формат Portable Executable (PE) – это формат файлов, используемый в 32-разрядных и 64-разрядных версиях операционных систем Microsoft Windows для исполняемых файлов, объектного кода и библиотеки динамической компоновки (DLL).

На рисунке 3 показана структура PE-файла, состоящая из нескольких заголовков и разделов. PE-файл определяет для загрузчика Windows, как отображать файл в памяти. Он состоит из заголовка PE-файла, таблицы разделов (заголовков разделов), за которыми следуют данные разделов.

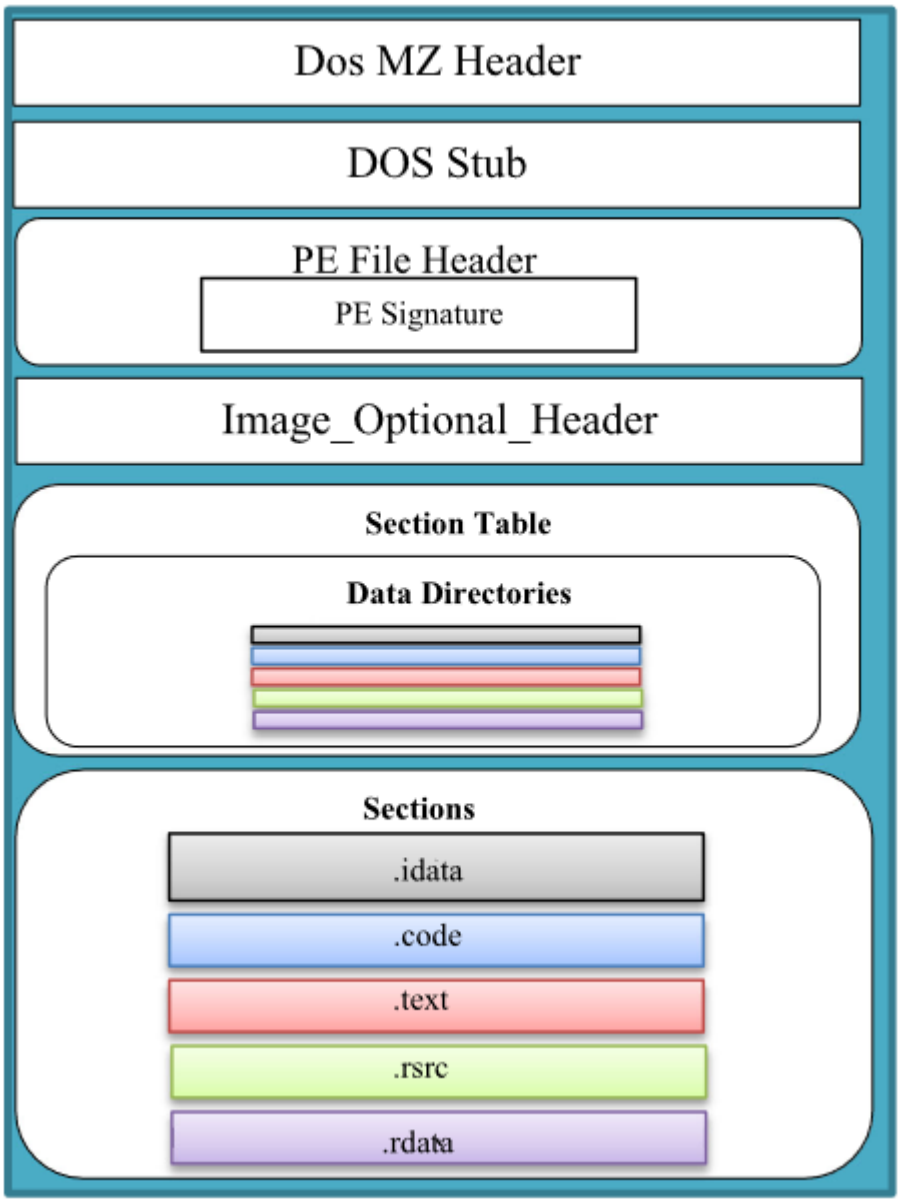

Рисунок 3 – Формат PE-файла

Пятый раздел посвящен программной реализации эвристического метода, в результате которой написана программа на языке программирования C++ – сканер PE-файлов, основанный на эвристическом методе. Главным методом по обнаружению вредоносного ПО является алгоритм, основанный на CFG. Также для увеличения количества распознавания вредоносных программ в сканере проверяется на изменения DOS заглушка (т.к. вирусы часто помечают ее, чтобы не происходило повторное инфицирование файла) и добавлена проверка последовательности OpCode, которая специфична для вредоносного ПО. Реализация данной программы представлена в Приложении А.

Работа программы делится на несколько этапов. После запуска сканирования, вызывается функция *scanDef*, в которой происходит поиск всех PE-файлов, находящихся в выбранном каталоге. Далее для каждого найденного файла вызывается функция *checkFile*, в которой он тестируется основными алгоритмами проверки файла: проверка DOS заглушки файла, проверка файла через граф потока управления и проверка через OpCode последовательность. За каждый непройденный тест повышается уровень угрозы файла.

DOS заглушка проверяется на искажения путем сравнивания ее с заглушкой, которая должна встречаться в большинстве обычных программ.

Алгоритм проверки файла через CFG делится на следующие шаги:

1. Берется .text секция файла и преобразовывается из машинного кода в текст программы на языке ассемблера.
2. Вызывается функция *generateCFG*, в которая проходится по ассемблерному коду и генерирует вершины графа, заполняя их необходимой информацией.
3. Далее в функции *makeEdges* соединяются вершины ребрами, используя адреса инструкций, полученные на шаге 2.
4. На последнем шаге запускается функция *findSubGraph* и в графе потока управления происходит поиск подграфа, который перенаправляет поток выполнения программы на стартовую вершину (т.е. вершину с дугой, направленную на начальную вершину). Если такой подграф есть, то уровень угрозы PE-файла повышается.

Для составления последовательности OpCode используются команды push, ret и адрес смещения на начало секции .text. Происходит это через запуск функции *opCodeCreate*. Далее для каждой секции PE-файла происходит поиск составленной последовательности. При нахождении такой последовательности повышается уровень угрозы файла и выписывается название секции, в которой была найдена последовательность и ее смещение в этой секции.

Подозрительные файлы записываются в контейнер *m_infFiles* и выводятся в область со списком подозрительных файлов. В зависимости от уровня угрозы, в списке они подсвечиваются красным, желтым или зеленым цветом.

На рисунке 4 представлено окно программы, после сканирования каталога. Программа вывела подозрительные файлы, среди которых находится тестовый инфицированный файл.

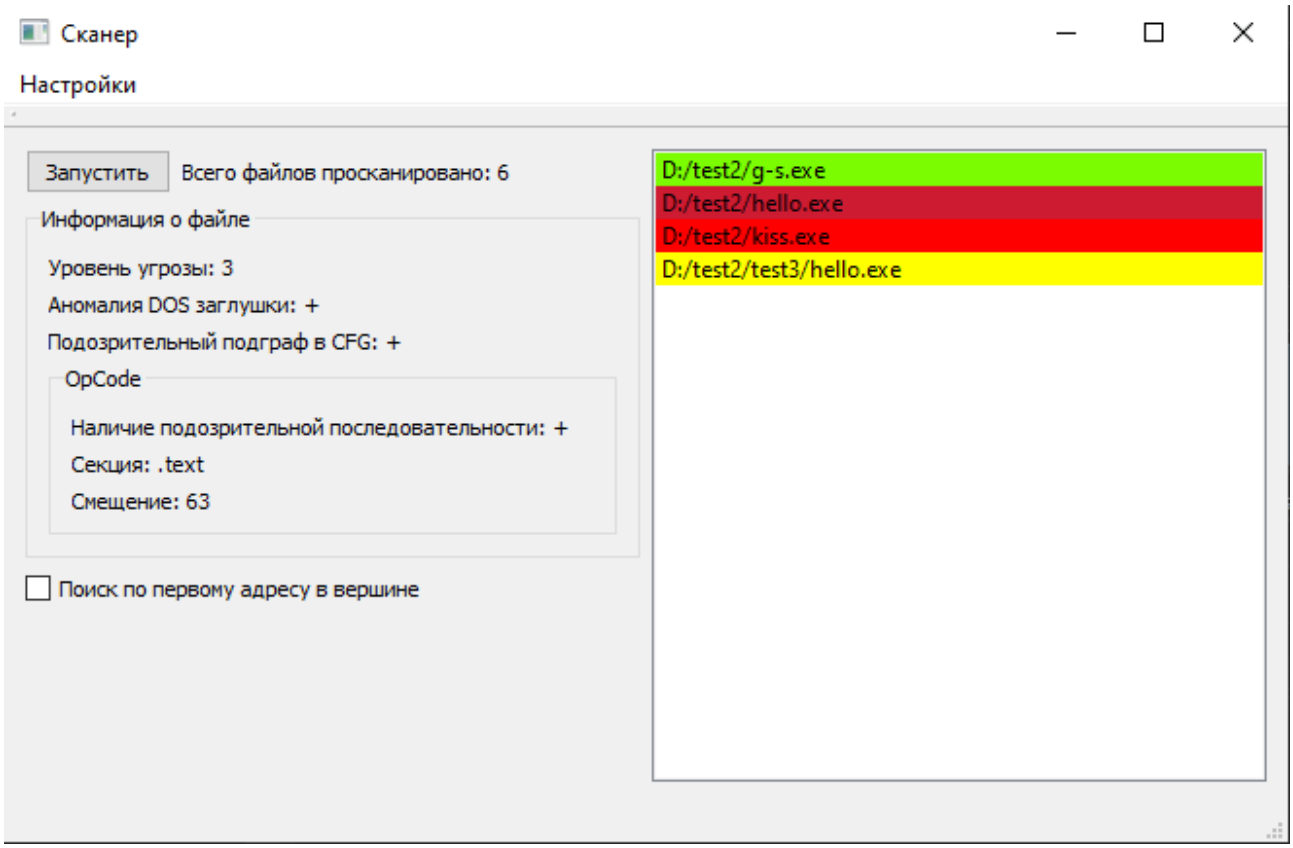

Рисунок 4 – Окно программы Сканер после сканирования

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы были рассмотрены основные положения антивирусных технологий и различные методы эвристического анализа, такие как:

- метод, основанный на API Calls,
- метод, основанный на CFG,
- метод, основанный на N-грамме,
- метод, основанный на OpCode.

В практической части работы реализована программа, позволяющая с помощью изученных, в частности с помощью CFG, распознавать вредоносные программы.

Данный метод довольно затратный по ресурсам компьютера, так как использует в основе графы и приходится оптимизировать программу при большом количестве вершин, но за счет этого можно обнаруживать новые вредоносные программы.

Данная программа может использоваться как антивирусный сканер и для проверки подозрительных PE-файлов.

Таким образом, все поставленные задачи полностью выполнены, цель достигнута.