**Оглавление.**

1. Введение.  
2. Обзор предметной области.

2.1. Структура PE-формата.

2.2. Особенности маскировки нежелательного программного обеспечения.

2.3. Анализ и выделение характерных признаков.

2.4. Информационная энтропия.

2.5. Другие характерные признаки.//TODO: *анализ других возможных способов идентификации, например, анализ импортов*

3. Описание программного комплекса SimScanner.

3.1. Назначение программного комплекса.

3.2. Алгоритм работы.

3.3. …

4. Экспериментальная часть.

5. Заключение.

6. Список литературы.

7. Приложения.

Приложение А. Термины и определения.

Приложение Б. Исходный код программы.

**1. Введение.**

Современное вредоносное программное обеспечение (*далее ВПО*) в общем случае представляет из себя исполняемые файлы, упакованные и зашифрованные сложнополиморфными алгоритмами. В отличие от полиморфных вирусов конца 90-х годов, современное ВПО в подавляющем большинстве случаев не содержит механизм размножения и не включает генератор полиморфного кода, который использовался для их маскировки. Соответственно, их маскировочные алгоритмы недоступны для анализа. Это т.н. offline полиморфизм, при котором после анализа и детектирования очередного образца ВПО немедленно в полуавтоматическом режиме создаётся и распространяется новый образец (обычно 15-30 образцов в сутки), и server-side полиморфизм, при котором новый шифрованный образец ВПО создаётся автоматически на стороне сервера ВПО при каждой очередной инсталляции.

Алгоритмы, разработанные против традиционных полиморфных вирусов, такие как эмуляция кода или попытки воспроизвести криптографический алгоритм в антивирусе, показали свою низкую эффективность применительно к подобным угрозам. Эмуляция кода снижает производительность антивирусного ПО, и в случае сложнополиморфных вирусов требует неприемлемых временных затрат (пример – RDA.Fighter, Idea).Анализ криптографического алгоритма требует усилий высококвалифицированных специалистов, которые не в состоянии обработать файловый поток новых образцов ВПО.

В связи с эволюцией ВПО возникла необходимость разработки более рациональных методов предварительного анализа с целью детектирования новых модификаций известного ВПО, прежде всего выявления характерных для каждого семейства ВПО признаков.  
Такими характерными признаками являются распределение информационной энтропии в исследуемом образце, характерный набор импортируемых функций, некоторые особенности формирования исполняемых файлов.

В данной работе была предпринята попытка анализа характеристик информационной энтропии исполняемых файлов. Для анализа был предложен адаптивный алгоритм на основе сигмоидальных нейросетей.

**2. Обзор предметной области.**

**2.1. Структура PE-формата.**

В большинстве случаев экземпляр ВПО представляет собой исполняемый файл Windows, имеющий формат Portable Executable (Microsoft Corp., 2017). Данный довольно сложный формат частично описан в Табл. 1.

Табл. 1. *Структура исполняемого файла формата Portable Executable*

|  |  |  |  |
| --- | --- | --- | --- |
| **Поле** | **Смещение** | **Размер, байт** | **Описание** |
| EXE signature | 0 | 2 | Поле содержит сигнатуру 'MZ' (инициалы одного из разработчиков формата)... |
| PE offset | 0x3C | 2 | Смещение PE-заголовка от начала файла. Смещение должно быть выровнено по 8. |
| PE signature | [PE offset] + 0 *Далее значения смещения отсчитываются от начала PE-заголовка* | 4 | Сигнатура, по которой загрузчик исполняемых форматов ОС определяет внутренний формат файла. Для Portable Executableимеет значение 'PE\0\0'. В некоторых устаревших форматах применялись другие сигнатуры ('LE\0\0', 'LX\0\0'). |
| File Header | 4 | 20 |  |
| Optional Header | 0x18 | - |  |
| Section headers | 0x1C + размер Optional Header | - |  |

Табл. 2. *Структура таблицы File Header*

|  |  |  |  |
| --- | --- | --- | --- |
| **Поле** | **Смещение** | **Размер, байт** | **Описание** |
| Machine | 0 | 2 (WORD) | Идентификатор архитектуры |
| Number Of Sections | 2 | 2 (WORD) | Кол-во секций в таблице секций |
| Time Date Stamp | 4 | 4 (DWORD) |  |
| Pointer To Symbol Table | 8 | 4 (DWORD) |  |
| Number Of Symbols | 0x0C | 4 (DWORD) |  |
| Size Of Optional Header | 0x10 | 2 (WORD) | Размер дополнительного заголовка |
| Characteristics | 0x12 | 2 (WORD) | Флаги |

Табл. 3. *Структура таблицы Optional Header*

|  |  |  |  |
| --- | --- | --- | --- |
| **Поле** | **Смещение** | **Размер, байт** | **Описание** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

В контексте поставленной задачи представляют интерес поля данного формата, перечисленные ниже.

**2.2. Особенности маскировки нежелательного программного обеспечения.**  
Полиморфизм состоит в формировании программного кода ВПО во время исполнения, при этом часть программы, формирующая вредоносный код, аналогичным образом, должна видоизменяется при каждом новом заражении. Подробное описание технологии полиморфизма представлено в Приложеннии А.

*//TODO: перенести в приложение А*

В качестве примера полиморфизма программного кода можно представить код программы, реализующей сложение двух чисел: A и В, а в качестве изменяющихся частей – действия над переменной С, которая никак не влияет на работу и конечные функции программы. Зачастую, изменение кода достигается путём добавления операторов, не изменяющих алгоритм работы программного кода.

Для шифрования ВПО используются т.н. полиморфные генераторы — исполняемые модули, главной функцией которых является шифрование тела вируса случайным ключом и генерация соответствующего случайного расшифровщика.

Таким образом зашифрованные полиморфными генераторами копии вирусов могут иметь разную длину и не совпадать ни в одном байте. При этом полиморфный генератор не считается вирусом, так как не содержит в себе алгоритма заражения.

Полиморфные генераторы обычно распространяются в виде модуля, присоединяющегося к коду ВПО и содержащего сам генератор. Вызов функции шифрования и генерации расшифровщика производится из тела ВПО вызовом соответствующей внешней-процедуры.

Постоянное видоизменение программного кода ВПО не позволяет создать универсальную сигнатуру для данного образца ВПО. На сегодняшний день для противодействия данному методу в антивирусном программном обеспечении применяются такие технологии, как эвристический анализ и эмуляция.

Для обнаружения таких ВПО на текущий момент используются технологии антивирусных ПО, такие как эмуляция и эвристическое сканирование.

В процессе эвристического анализа производится проверка эмулируемой программы анализатором кода. Например, файл заражен полиморфным вирусом, состоящим из зашифрованного тела и расшифровщика. Эмулятор кода считывает инструкции в буфер антивируса, разбирает их на инструкции и запускает их исполнение по одной инструкции, после этого анализатор кода подсчитывает контрольную сумму и сверяет её с той, которая хранится в базе. Эмуляция будет продолжаться до тех пор, пока необходимая для подсчета контрольной суммы часть вируса не будет расшифрована. Если сигнатура совпала — программа определена как зараженная.

Т.о., эвристическое сканирование нередко используется совместно с сигнатурным сканированием для определения сложнополиморфных ВПО.

На сегодняшний день, методы эвристического анализа не могут обеспечить должную защиту от новых или отсутствующих в сигнатурном наборе вирусов, потому как используют в качестве объекта анализа сигнатуры ранее известных вирусов, а в качестве правил эвристической верификации — знаний о механизме полиморфизма сигнатур. К тому же, этот метод основывается в большей степени на эмпирических предположениях, по этой причине полностью исключить ложные срабатывания в принципе невозможно.

Именно поэтому у данного метода можно выделить ряд недостатков:

Чрезмерная подозрительность эвристического анализатора зачастую вызывает ложные срабатывания при наличии в программе фрагментов кода, выполняющего действия, свойственные некоторым вирусам.

Существует множество простых методик обмана эвристического анализатора. Перед «выпуском» вируса, разработчики зачастую, используя известные антивирусные ПО, стремятся к тому, чтобы избежать ими обнаружения их вредоносного кода.

Помимо прочего, даже если удалось верно определить ВПО, вылечить зараженные файлы, как правило, не представляется возможным.

Есть и второй метод обнаружения полиморфных ВПО: эмуляция. Этот метод используют для обнаружения т.н. 0day-угроз – еще неизвестных вредоносных программ.

Суть метода заключается в выполнении потенциально опасного приложения в эмулируемой среде. Во время эмуляции антивирусная программа отслеживает все действия исследуемого программного обеспечения, что позволяет, на основании внутренних алгоритмов антивирусных программ, оценить степень опасности исследуемого программного обеспечения.

Недостатки этого метода очевидны: высокое потребление системных ресурсов. Это негативно сказывается на производительности компьютера. Поэтому, на сегодняшний день, эмуляция не является основной технологией антивирусного программного обеспечения.

Практически все современные антивирусные средства применяют технологию эвристического анализа программного кода. Тем не менее этот метод сканирования, как было показано[...], имеет ряд существенных недостатков.

Одной из возможностей обойти некоторые из них, является использование при сканировании файлов специально обученной нейросети (*далее ИНС*). Так как ИНС обладает огромным потенциалом и обучаемостью, она является наиболее гибким вариантом, способным предотвратить ложные срабатывания и повысить общее количество обнаружения зараженных файлов.

Проблема заключается только в том, какие данные подавать на вход ИНС.

Так как зачастую полиморфные вирусы шифруются в автоматическом или полуавтоматическом режиме, они чаще всего отличаются ключами шифрования. Самих же алгоритмов шифрования существует ограниченное количество. Хоть это и не облегчает работу антивирусных ПО с текущими методами обнаружения вирусов, это дает возможность по-новому посмотреть на структуру ВПО в принципе, найти сходства и различия на основании этой структуры.

**2.3. Анализ и выявление характерных признаков.**

Таким образом, посредством анализа методов шифрования полиморфных вирусов, мною было выдвинуто предположение о том, что при использовании конкретного алгоритма шифрования, но с генерацией разных ключей, информационная энтропия программного кода остается неизменной или незначительно отличается на погрешности.

...

**2.4. Информационная энтропия.**

Информационная энтропия — мера неопределённости или непредсказуемости некоторой системы, в частности неопределённость появления какого-либо символа первичного алфавита. В последнем случае при отсутствии информационных потерь энтропия численно равна количеству информации на символ передаваемого сообщения. Подробно суть данной характеристики сообщения описана в Приложении А.

*//TODO: сравнительный график энтропии от нескольких сэмплов*

**2.5. Другие характерные признаки.**

...

**3. Описание программного комплекса SimScanner.**

**3.1. Назначение программного комплекса.**

**3.2. Алгоритм работы программного комплекса.**

...

**4. Экспериментальная часть.  
5. Заключение.**

**6. Список литературы. //?**

**7. Приложения.**

**Приложение А. Термины и определения.**

***Полиморфизм —*** *в терминах информационной безопасности маскирующее свойство нежелательного программного обеспечения, состоящее в видоизменении каждой копии исполняемого объекта до неузнаваемости.* (C., 1996)(Harrison, 2015)(Harrison, 2015) *В антивирусной терминологии (и здесь далее) такие исполняемые объекты называются* ***полиморфиками****. В самом распространённом случае алгоритм полиморфизации кода следующий:*

* *Код ВПО сжимается каким-либо упаковочным алгоритмом, один или несколько раз шифруется каким-либо параметризуемым криптографическим алгоритмом. Параметры алгоритма, такие как ключи, подбираются случайным образом. Такой обработке подвергается код, содержит полезную троянскую нагрузку; как правило, к нему же привязываются сигнатуры антивирусов. При построении расшифровщика используются синонимичные команды и командные конструкции, между которыми внедряются «мусорные» инструкции, антиэвристические и антиотладочные приёмы, детекторы виртуального окружения, многократная вложенность расшифровщиков и т.п. Отдельно следует упомянуть т. н.* ***пермутеры****: в отличие от обычных полиморфиков, пермутеры не обязательно шифруют свой код; они перестраивают его целиком. Для этих целей может использоваться встроенный дизассемблер или аналогичная байт-коду символическая схема построения кода, сама по себе тоже полиморфная (такие вирусы называются* ***метаморфиками*** *и не выделяются в отдельный класс). В классификации CARO таким вирусам присваивается VI-ой уровень полиморфизма. Это самый сложный класс ВПО, известно всего несколько его полноценных реализаций. По трудоёмкости такая разработка близка к разработке оптимизирующих компиляторов. Позднее в указанную классификацию был добавлен VII-ой уровень – полиморфики, использующие собственные виртуальные машины. В настоящий момент данная классификация признана устаревшей* (Bontchev, 2005)*.*
* *На основе параметров шифрования динамически создаётся соответствующий расшифровщик.*
* *Если образец ВПО предусматривает функции самостоятельного распространения, полиморфный генератор внедряется в исполняемый объект (этот этап чаще происходит на стадии линковки исполняемого объекта, и тогда полиморфный генератор представляет собой единое целое с основным кодом ВПО). В дальнейшем такое ВПО обращается к полиморфному генератору на стадии самокопирования.*
* *Упаковщик создаёт новый исполняемый файл, в данные которого помещает полученную криптограмму, а в исполняемой части файла размещает расшифровщик.*

***Полиморфный генератор*** *– библиотека, подключение которой придаёт всему исполняемому объекту свойство полиморфизма. Обычно распространяется в виде объектного модуля или исходного текста. Сами по себе полиморфные генераторы вирусами не являются, более того, существуют легальные коммерческие продукты, представляющие собой, по сути, полиморфные генераторы (****ASPack/ASProtect, VMProtect****). Типичный полиморфный генератор состоит из следующих программных модулей:*

* *ГСЧ и шифровальщик*
* *Генератор «мусора»*
* *Генератор расшифровщика*

*//TODO: здесь*

***Сигнатура*** *– первичный идентификационный признак ВПО, используемый с момента возникновения антивирусной отрасли. В простейшем случае это короткая байтовая последовательность, уникальная для данного образца ВПО. Антивирусы-сканеры содержат в своей поставке регулярно обновляемую базу сигнатур, и при поиске образцов ВПО сверяют содержимое каждого анализируемого файла с данной базой. Сигнатуры в полуавтоматическом режиме выбираются антивирусными аналитиками. Подбор удачной сигнатуры может оказаться нетривиальной задачей.*

***Эвристический анализ*** *– один из распространённых механизмов бессигнатурного поиска ВПО, в том числе нового, ранее неизвестного. Эвристический анализ состоит в поиске признаков, действий и программных приёмов, специфичных для ВПО, и нехарактерных для обычного программного обеспечения* (Symantec Corp., 1997)*. Распаковка, внедрение в другие исполняемые объекты (как файлы, так и запущенные процессы в памяти), необычная сетевая активность, эксплуатация уязвимостей после комплексного анализа могут служить признаками ВПО, однако в процессе эволюции значительная часть актуального ВПО утратила отличительные черты: например, механизм самовоспроизводства в настоящее время является высокотехнологичной экзотикой.*

***Эмуляция*** *- ...*

***Энтропия информационная****, или энтропия Шеннона — мера хаотичности информации, степень неопределённости одного из символов т. н. первичного алфавита. (Применительно к существующим информационным технологиям и в рамках решаемой задачи первичный алфавит представляет весь возможный набор значений одного байта* [0..0xFF]*.) Информационная энтропия отражает степень сжатия блока данных. В определении Шеннона информационная энтропия выражается формулой (I),*

|  |  |
| --- | --- |
|  | (I) |

*где b – основание энтропии, равное 2 для бинарной энтропии, S – источник энтропии (сообщение, блок данных) с первичным алфавитом* {*a*0, …, *a*n}*, в самом распространённом случае диапазон значений одного байта* [0; 0xFF]*, pi – частота i-того символа первичного алфавита в источнике энтропии S.*

*Информационная энтропия может также быть определена, как мера удельной информации на символ сообщения.* (Шеннон, 2002)

**Список литературы.**

**Current Status of the CARO Malware Naming Scheme.** [Конференция] / авт. Bontchev V. // 15th Virus Bulletin Conference. - 2005.

**Encrypted and Oligomorphic Viruses** / auth. Harrison William L.. - Los-Angeles : Center for High Assurance Computing, California State University, 5 2015. - Vol. 7440.

**Microsoft Portable Executable and Common Object File Format Specification, Revision 11 - January 23, 2017** [Статья] / авт. Microsoft Corp.. - [б.м.] : Microsoft Corp., 23 1 2017 г.. - Revision 11.

**Polymorphic & Metamorphic Viruses** [Journal] / auth. Harrison William L.. - Los-Angeles : Center for High Assurance Computing, California State University, 5 2015. - CS4440/7440.

**Understanding and Managing Polymorphic Viruses.** [Journal] / auth. C. Nachenberg. - [б.м.] : Symantec, 1996. - The Symantec Enterprise Papers : Vol. XXX.

**Understanding Heuristics: Symantec’s Bloodhound Technology.** [Journal] / auth. Symantec Corp.. - Cupertino : Synantec Corp., 1997. - Symantec White Paper Series : Vols. XXXIV, .

**Virus and Vulnerability Classification Schemes: Standards and Integration.** [Отчет] / авт. Gordon S.. - Cupertino : Symantec, 2003.

**Работы по теории информации и кибернетике.** [Книга] / авт. Шеннон К.. - Москва : Изд. иностр. лит., 2002.