МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»
(СГУ)

Кафедра теоретических основ компьютерной безопасности и криптографии

Программно-аппаратный комплекс защищенного исполнения программного обеспечения

АВТОРЕФЕРАТ

дипломной работы

студентки 6 курса 631 группы

специальности 10.05.01 Компьютерная безопасность

факультета компьютерных наук и информационных технологий

Кирьяновой Анны Дмитриевны

Научный руководитель

к. п. н., доцент _____ А. С. Гераськин

18.01.2019 г.

Заведующий кафедрой

д. ф.-м. н., доцент _____ М. Б. Абросимов

18.01.2019 г.

Саратов 2019

ВВЕДЕНИЕ

Проблема несанкционированного использования, распространения, анализа и кражи интеллектуальной собственности, входящей в состав программного обеспечения (ПО), появилась еще в 70-х годах прошлого столетия и не теряет актуальности до сих пор, а потому вопрос защиты ПО остается острым [1].

Программный продукт, вообще говоря, является интеллектуальной собственностью, в процессе создания которой часто возникают большие материальные и трудовые затраты. Поэтому можно понять, когда у разработчика подобного продукта появляется желание защитить его от нарушения авторских прав и несанкционированного распространения [2]. Помимо этого, в самом продукте могут быть использованы и реализованы алгоритмы, являющиеся тайной, как личной, так и коммерческой или даже государственной тайной.

Защитой программного обеспечения будем называть комплекс мер, направленных на защиту программного обеспечения от несанкционированного приобретения, использования, распространения, модифицирования, изучения и воссоздания аналогов.

Несмотря на то, что рынок средств защиты является в некотором смысле перенасыщенным, пока не существует идеального продукта, все имеют те или иные слабые стороны [3]. В данной работе предлагается разработка комплекса, позволяющего решить проблемы многих других средств и представляющего из себя защищенный носитель, токен, с защищенной средой на нем, реализованной на основе виртуальной машины (ВМ), эмулирующей машину Поста.

Целью дипломной работы является создание программно-аппаратного комплекса защищенного исполнения программного обеспечения.

Для достижения цели были поставлены следующие задачи:

- Анализ литературных источников, посвященных проблеме средств защиты программного обеспечения;
- Построение теоретической модели защищенного исполнения кода и обзор ее защищенности;
- Разработка на основе построенной модели программно-аппаратного комплекса защищенного исполнения программного обеспечения.

Дипломная работа состоит из введения, 5 разделов, заключения, списка использованных источников и 3 приложений. Общий объем работы – 74 страниц, из них 42 страниц – основное содержание, включая 19 рисунков, список использованных источников из 18 наименований.

КРАТКОЕ СОДЕРЖАНИЕ

В разделе **1 «Проблемы защиты программного обеспечения»** вводятся определения защиты информации и технической защиты информации, а также даются определения кода программного обеспечения и защищенного исполнения программного обеспечения, приводятся причины защиты кода ПО и угрозы, возникающие на разных этапах жизненного цикла программы [4].

В подразделе **1.1 «Причины защиты кода программного обеспечения»** рассматриваются виды и причины защиты кода с точки зрения разработчика ПО. Защита результата выполнения программы, которая обусловливается желанием разработчика получить материальную выгоду за предоставленное решение определенной задачи. Защита кода от изучения объясняется необходимостью скрытия разработанных алгоритмов для предотвращения несанкционированного использования и воспроизведения алгоритмов, которое может повлечь создание конкурирующего продукта. Приводятся результаты исследования компании Irdeto, в котором указывается что одна компания (неназванная) из-за распространения в интернете более 350 тыс. копий потеряла потенциально более 21 млн. долларов [5].

В подразделе **1.2 «Угрозы, возникающие в отношении кода программного обеспечения»** представлена модель атакующего, реализующего одну из следующих угроз: копирования и распространения, несанкционированного использования для получения результата работы.

Рассматривается жизненный цикл программы, состоящий из трех этапов: передача ПО, хранение на устройстве пользователя, запуск и исполнение, а также угрозы, возникающие на каждом этапе [6].

В разделе **2 «Методы защиты кода программного обеспечения»** рассматриваются существующие теоретические решения и методики защиты ПО.

В подразделе **2.1 «Обфускация кода»** рассматривается один из методов защиты кода от изучения — обфускация, а также приводятся некоторые его техники [7, 8].

В подразделе **2.2 «Шифрование»** рассматриваются особенности полного и частичного шифрования кода программы [9]. Делается вывод о том, что частичное шифрование дает лучшие результаты в защите программы в сравнении с полным шифрованием.

В подразделе **2.3 «Виртуальная исполняемая»** среда дается определение данного метода, описывается его преимущество по сравнению с другими, основанное на том, что злоумышленнику для взлома необходимо знание архитектуры виртуальной машины и используемой системы команд. Без этого знания код программы (зашифрованный или открытый) не даст взломщику никакой информации, что также сильно усложняет задачу реверс инжиниринга [10].

В подразделе **2.4 «Привязка к среде исполнения»** вводятся понятия защищенной и незащищенной среды, описываются различные способы осуществления привязки программы к среде, в которой программа будет выполняться: привязка к машине пользователя на основе одной или совокупности характеристик компьютера пользователя, привязка к удаленному серверу или токену, а также привязка к виртуальной машине [11].

В разделе **3 «Разработка модели программно-аппаратного комплекса защиты программного обеспечения»** происходит выбор оптимального сочетания методов защиты и защищенной среды, описывается модель, на основе которой реализуется виртуальная машина (ВМ), а также описываются особенности шифрования кода программ, возникающие в процессе реализации данной ВМ.

В подразделе **3.1 «Выбор оптимальной комбинации методов защиты»** рассматриваются описанные в главе 2 методы защиты ПО на различных этапах

жизненного цикла программы, строится граф эволюции состояний защиты программы в ее жизненном цикле. На основе построенного графа осуществляется выбор оптимальной комбинации методов защиты, а именно перманентное частичное шифрование кода программы и привязка ее к защищенной среде исполнения.

В подразделе **3.2 «Выбор защищенной среды и особенности взаимодействия с ней»** производится выбор защищенной среды между токеном, подконтрольным сервером и виртуальной машиной. В качестве среды выбирается токен, однако обнаруживается новая угроза — угроза подачи вредоносного для токена кода. Для защиты самого токена принимается решение о создании на нем защищенной среды в виде виртуальной машины, ограниченной с точки зрения взаимодействия с ОС и аппаратной частью, но не ограниченной с точки зрения функциональности [12, 13].

В подразделе **3.3 «Модель виртуальной машины»** вводятся определения ВМ, программы ВМ. Объясняется сложность реверс инжиниринга при использовании виртуальной машины, и описывается классическая атака программы в таком случае [14]. Описывается принцип работы ПО при частичном шифровании кода: при запуске программа выполняется в обычном режиме до момента вызова защищенной функции, при обращении к зашифрованному участку тот подается на вход ВМ на токене, где происходит расшифровка и выполнение данного участка кода. Результат исполнения участка передается с токена в среду пользователя, и далее программа работает в обычном режиме до очередного вызова защищенного участка.

Бессмысленность подачи на вход ВМ вредоносного кода объясняется тем, что из-за специфики реализуемой модели — машины Поста, поскольку ВМ представляет из-за себя ленту с нулями и единицами, по которой мы можем двигаться, ставить и убирать пометки, обозревать содержимое ячейки, но не более. Такой ограниченный набор операций и обеспечивает защиту,

поскольку в худшем случае программа завершится некорректно либо не завершится никогда, при этом не будет нанесен вред ВМ или токену.

В подразделе **3.4 «Особенности защиты шифрованием кода при использовании виртуальной машины»** приводится схема шифрования кода на разрабатываемой ВМ. При этом, между CBC (Cipher Block Chaining, или режим сцепления блоков шифротекста) и ECB (Electronic Codebook, или режим электронной книги) делается выбор в пользу шифрования каждой инструкции в режиме ECB, поскольку, хоть он и обеспечивает меньшую надежность шифрования, такой режим дает более высокую скорость работы и меньшие затраты памяти, поскольку не надо держать в ней всю расшифрованные инструкции [15].

В разделе **4 «Реализация модели программно-аппаратного комплекса защиты программного обеспечения»** даются основные определения машины Поста, вводятся 6 доступных операций и перечисляются возможные исходы выполнения программы машины Поста — безрезультатная остановка, результативная и вечный процесс работы [16]. Описывается программная реализация ВМ на основе машины Поста и выбор и настройка аппаратного токена.

В подразделе **4.1 «Использование программ машины Поста для защиты программного обеспечения»** объясняется применимость машины Поста для произвольного существующего алгоритма тем, что она является Тьюринг-полной. Описывается особенность работы с типами данных на ленте машины.

В подразделе **4.2 «Программная реализация выбранной виртуальной машины»** описывается специфика входных данных, особенность ленты и работы с ней, формат инструкции, возможности компилятора. Исходный код библиотеки для работы с машиной Поста представлен в приложении А.

В подразделе **4.3 «Выбор и настройка аппаратного токена»** обосновывается выбор в качестве аппаратного токена миниатюрный портативный компьютер Orange Pi i96 и в общих чертах описывается его настройка [17].

Подраздел **4.4 «Реализация программного обеспечения для исполнения защищенных программ машины Поста на токене»** описывает реализацию ПО для исполнения программ на токене и схематично представляет протокол обмена данными клиентской и серверной частей. Исходный код интерфейса для удаленного исполнения программ машины Поста представлен в приложении Б.

Раздел **5 «Демонстрация работы разработанного программно-аппаратного комплекса»** представляет программу, разработанную для демонстрации работы программно-аппаратного комплекса.

В подразделе **5.1 «Разработка защищенного программного продукта»** приводится разработанный алгоритм шифрования, в основе которого лежат такие операции над битами, как замешивание, полуотражение и сложение по модулю 2. Алгоритм шифрования и дешифрования разбиваются на отдельные защищаемые участки, которые представляются перечисленные операциями. Для этих трех операций были написаны программы с использованием команд машине Поста, которые вызываются по мере выполнения алгоритма шифрования и дешифрования.

В подразделе **5.2 «Демонстрация работы разработанного продукта в защищенном режиме»** приведены снимки экрана, демонстрирующие работу с разработанным защищенным программно-аппаратными комплексом защищенного исполнения ПО. Исходный код разработанной защищенной программы представлен в приложении В.

ЗАКЛЮЧЕНИЕ

Защита программного обеспечения на всех этапах жизненного цикла программы является важной проблемой и не потеряет актуальности в ближайшее время. В связи с тем, что ни одно средство защиты не является на данный момент лишенным недостатков, остается потребность в продукте, обеспечивающем защиту ПО на всех этапах и от всех угроз.

В данной работе был проведен обзор существующих теоретических методов защиты программного обеспечения с точки зрения защиты его кода, рассмотрены соответствующие существующие решения и анализ их эффективности на каждом этапе его жизненного цикла программного обеспечения.

После рассмотрения была разработана минимальная эффективная модель, использующая некоторые из рассмотренных техник, при этом позволяющая выполнять поставленные задачи без потери свойств защищенности. После разработки в теоретическом виде она была реализована набором из библиотек, реализующих виртуальную машину Поста, способную выполнять зашифрованный код, компилятора для нее, выполняющего также некоторые другие вспомогательные функции, а также аппаратного токена и серверного ПО для него.

Для демонстрации возможностей использования был разработан программный продукт, использующий все доступные функции и реализованный инструментарий, реализующий простой алгоритм шифрования на основе побитового сложения. На его основе были проведены и задокументированы все шаги, необходимые для его защиты с использованием модели, библиотек, компилятора и аппаратного токена, демонстрируя большой потенциал для практического применения описанных в работе методов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Середа, С. А. Процедура разработки систем программно-технической защиты программного обеспечения [Электронный ресурс] : BugTraq. URL: https://bugtraq.ru/library/misc/protectproc.html?k=9 (дата обращения 11.10.2018). Загл. с экрана. Яз. рус;

2 Новичков, А. Анализ рынка средств защиты от копирования и взлома программных средств [Электронный ресурс] : CitForum. URL: http://citforum.ru/security/articles/analis/ (дата обращения 11.10.2018). Загл. с экрана. Яз. рус;

3 Буинцев, Д. Н. Метод защиты программных средств на основе запутывающих преобразований [Электронный ресурс] : Электронная библиотека диссертаций. URL: http://www.dissercat.com/content/metod-zashchity-programmnykh-sredstv-na-osnove-zaputyvayushchikh-preobrazovanii (дата обращения 05.01.2019). Загл. с экрана. Яз. рус;

4 ГОСТ Р 50922-2006. Защита информации. Основные термины и определения. М.: Стандартинформ, 2008;

5 Game Piracy Results in Over $21 Million in Lost Revenue During Opening Release Window [Электронный ресурс] : Irdeto. URL: https://irdeto.com/news/game-piracy-results-in-over-21-million-in-lost-revenue-during-opening-release-window/ (дата обращения 10.10.2018). Загл. с экрана. Яз. англ;

6 ГОСТ 34.601-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы стадии создания. М.: Стандартинформ, 2009;

7 Бойко, П. В. Обфускация и защита программных продуктов [Электронный ресурс] : On-line библиотека свободно доступных материалов по информационным технологиям на русском языке. URL:

http://citforum.ru/security/software/virt_proc/ (дата обращения: 08.10.2018). Загл. с экрана. Яз. рус;

8 Морфим, не отходя от кассы: Мутация кода во время компляции [Электронный ресурс] : Хакер. URL: https://xakep.ru/2010/08/07/54469 (дата обращения: 08.10.2018). Загл. с экрана. Яз. рус;

9 Cappaert, J. Self-encrypting Code to Protect Against Analysis and Tampering [Электронный ресурс] : University of Auckland Research Repository — ResearchSpace. URL: https://researchspace.auckland.ac.nz/bitstream/handle/2292/3491/TR148.pdf (Дата обращения: 26.12.2018). Загл. с экрана. Яз. англ;

10 Метод виртуального процессора в защите программного обеспечения [Электронный ресурс] : On-line библиотека свободно доступных материалов по информационным технологиям на русском языке. URL: http://citforum.ru/security/software/virt_proc/ (дата обращения: 08.10.2018). Загл. с экрана. Яз. рус;

11 Petrov, A. Methods of executable code protection [Электронный ресурс] : arXiv. URL: https://arxiv.org/pdf/1403.1694.pdf (Дата обращения: 27.12.2018). Загл. с экрана. Яз. англ;

12 Building a Secure System using TrustZone Technology [Электронный ресурс] : ARM Security Technology. URL: http://infocenter.arm.com/help/topic/com.arm.doc.prd29-genc-009492c/PRD29-GENC-009492C_trustzone_security_whitepaper.pdf (Дата обращения: 02.01.2019). Загл. с экрана. Яз. англ;

13 Sabt, M. Trusted Execution Environment: What It is, and What It is Not [Электронный ресурс] : IEEE Xplore Digital Library. URL: https://ieeexplore.ieee.org/document/7345265 (Дата обращения: 27.12.2018). Загл. с экрана. Яз. англ;

14 Huang, K. Enhance virtual-machine-based code obfuscation security through dynamic bytecode scheduling [Электронный ресурс] : ScienceDirect. URL: https://www.sciencedirect.com/science/article/pii/S0167404818300270 (Дата обращения: 21.11.2018). Загл. с экрана. Яз. англ;

15 Schneier. B. Applied Cryptography [Электронный ресурс] : Internet Archive. URL: https://archive.org/details/AppliedCryptographyBruceSchneier (Дата обращения: 04.01.2019). Загл. с экрана. Яз. англ;

16 Успенский, В. А. Машина Поста [Электронный ресурс] : Электронная библиотека «Альтернативная наука». URL: http://www.vixri.com/d/USPENSKIJ%20V.%20A.%20%20_MAShINA%20POSTA.pdf (Дата обращения: 17.11.2018). Загл. с экрана. Яз. рус;

17 Orange Pi i96 [Электронный ресурс] : Orange Pi. URL: http://www.orangepi.org/OrangePii96/ (Дата обращения: 06.01.2019). Загл. с экрана. Яз. англ;

18 Verdult, R. The (in)security of proprietary cryptography [Электронный ресурс] : Institute for Computing and Information Sciences. URL: http://www.cs.ru.nl/~rverdult/phd_thesis-roel_verdult.pdf (Дата обращения: 09.01.2019). Загл. с экрана. Яз. англ;