Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

УДК 004.056.5

На правах рукописи

ЕРОХОВЕЦ
Наталия Юрьевна

АЛГОРИТМ ЗАЩИТЫ ИСПОЛНЯЕМОГО ПРОГРАММНОГО КОДА ОТ ДИНАМИЧЕСКОГО И СТАТИЧЕСКОГО АНАЛИЗА

АВТОРЕФЕРАТ
диссертации на соискание степени
магистра технических наук

по специальности 1-38 80 04 – Технология приборостроения

Минск 2018

Работа выполнена на кафедре проектирования информационно-компьютерных систем учреждения образования «Белорусский государственный университет информатики и радиоэлектроники»

Научный руководитель: **ШАТАЛОВА Виктория Викторовна,**
кандидат технических наук, доцент, доцент кафедры проектирования информационно-компьютерных систем учреждения образования «Белорусский государственный университет информатики и радиоэлектроники»

Рецензент: **ПОЛОЗКОВ Юрий Владимирович,**
кандидат технических наук, доцент, заведующий кафедрой «Программное обеспечение вычислительной техники и автоматизированных систем» Белорусского национального технического университета

Защита диссертации состоится «27» июня 2018 г. года в $10^{\underline{00}}$ часов на заседании Государственной экзаменационной комиссии по защите магистерских диссертаций в учреждении образования «Белорусский государственный университет информатики и радиоэлектроники» по адресу: 220013, Минск, ул. П. Бровки, 6, копр. 1, ауд. 415, тел. 293-20-80, e-mail: kafpiks@bsuir.by

С диссертацией можно ознакомиться в библиотеке учреждения образования «Белорусский государственный университет информатики и радиоэлектроники».

ВВЕДЕНИЕ

Защита программного обеспечения от неавторизованного использования, модификации и копирования является важнейшей задачей современных информационно-вычислительных систем. Лицензионные соглашения между поставщиком программного продукта и пользователем обычно ограничивают конечного пользователя возможностью использования результатов выполнения программ, и не передают права на алгоритмы, используемые в программном продукте. При несанкционированном доступе к исполняемым кодам программного обеспечения возможен анализ и реконструкция этих алгоритмов третьими лицами. В результате этого становится возможным нарушение прав интеллектуальной собственности, кража технологий и несанкционированная модификация программного обеспечения. Нынешний уровень коммерческих систем защиты недостаточен для надежной защиты прикладных программ ответственного назначения. Большинство коммерческих систем защиты довольно быстро после ввода их в эксплуатацию взламываются злоумышленниками.

Учитывая широкое распространение технологий виртуализации и облачных вычислений, в рамках которых прикладное программное обеспечение часто исполняется в недоверенной вычислительной среде, особую актуальность приобретают исследования и разработки, направленные на создание новых методов защиты программ от незаконного использования и обратного проектирования применяемых в них алгоритмов.

Разработка эффективных методов противодействия неавторизованному использованию программных кодов является основой создания надежного барьера на пути распространения нелицензионной программной продукции.

Применяемые в настоящее время методы защиты от нелицензионного использования и копирования программного обеспечения можно разделить на организационно-инфраструктурные и функциональные. Первые направлены на формирование доверенной вычислительной среды, тогда как вторые - на блокирование действий, разрушающих существующие средства защиты программ от копирования и обратного проектирования. Функциональные методы защиты используют различные алгоритмы и методы преобразования исполняемого кода прикладных программ к виду, затрудняющему анализ алгоритмов, положенных в основу их функционирования с помощью технологии реинжиниринга и средств анализа данных. Одним из известных методов функциональных защиты программ является использование запутывающих преобразований (обфускации) исполняемого кода. После таких преобразований программный код может исполняться только на виртуальных машинах (ВМ), оснащенных специальным процессором с набором предназначенных для этого команд и средствами доступа к области оперативной памяти, в которой хранится защищенный программный код и данные. Использование ВМ с процессором, исполняющим обфусцированный код (ПИОК), повышает доверие пользователей к среде исполнения программ, а также обеспечивает за-

щиту программных систем от реинжиниринга или нелицензионного использования.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы исследования

Компьютерное пиратство и незаконное использование программ наносит большой вред экономике страны, особенно ее высокотехнологичному сектору. Согласно оценке специалистов потери от использования нелицензионного программного обеспечения постоянно растут, что свидетельствует о необходимости повышения эффективности методов его защиты.

Использование ВМ с процессором, исполняющим обфусцированный код (ПИОК), повышает доверие пользователей к среде исполнения программ, а также обеспечивает защиту программных систем от реинжиниринга или нелицензионного использования.

Степень разработанности проблемы

Исследование алгоритма защиты программного кода от динамического и статического анализа проводилось на основе работ российских и зарубежных ученых, в том числе В.А. Захарова, П.Д. Зегжда, В.П. Бойко, В.С. Заборовского, Н.Н. Кузюрина, А.В. Шокурова, Р.И. Подловченко, В.Л. Щербина, Н.П. Варновского, С.С. Гайсаряна, В.П. Иванникова, А.И. Аветисяна, К. Коллберга, К. Сомборсона, Д. Викстрома, Д. Лоу, Б. Барака, Д. Бергстрома, Ф.де Клю.

В работах указанных авторов отмечается необходимость создания алгоритмов защита программного обеспечения, которые обладают высокой стойкостью к исследованию, организованным с использованием инструментальных средств обратного проектирования. С учетом вышеизложенного, работа посвящена разработке алгоритма защиты программного обеспечения с использованием виртуальных машин. Особенностью предлагаемых решений является масштабируемость по отношению к объемам защищаемых программных кодов, и эффективная адаптпция к различным операционным средам исполнения программы.

Цель и задачи исследования

Целью исследования является разработка алгоритма защиты исполняемого программного кода от компьютерных атак обратного проектирования на основе динамического и статического анализа данных.

Поставленная цель работы определяет **следующие основные задачи**:

1. Разработать модель угроз, связанных с использованием технологий обратного проектирования исполняемого программного кода, основанных на динамическом анализе (отладке) и статическом исследовании исполняемого кода.

2. Разработать алгоритм защиты от компьютерных атак обратного проектирования прикладного программного обеспечения, основанный на заме-

щении выбранного критически важного сегмента кода защищенным программным объектом.

3. Разработать алгоритм защиты исполняемого кода, основанный на запутывающих преобразованиях, которые используют изменение потока управления и внесения избыточности.

Область исследования

Содержание диссертации соответствует образовательному стандарту высшего образования второй ступени (магистратуры) ОСВО 1-38 80 04-2012 специальности 1-38 80 04 «Технология приборостроения».

Теоретическая и методологическая основа исследования

В основу диссертации легли работы белорусских и зарубежных ученых в области создания алгоритмов защиты программного кода от динамического и статического анализа, а также анализ технических нормативных правовых актов по рассматриваемой тематике.

Информационная база исследования сформирована на основе литературы, открытой информации, технических нормативно-правовых актов, сведений из электронных ресурсов, а также материалов научных конференций и семинаров.

Научная новизна

Научная новизна и значимость полученных результатов работы заключается в разработке алгоритма защиты при помощи запутывающих преобразований программного кода, подвергающегося компьютерным атакам обратного проектирования

Теоретическая значимость работы заключается в детальном анализе процессов воздействия динамического и статического анализа на программный код

Практическая значимость диссертации состоит в разработанном алгоритме защиты программного обеспечения от исследования и реинжиниринга, который позволит повысить качество защиты программного кода.

Основные положения, выносимые на защиту

1. Модель угроз обратного проектирования машинного кода при условии доступа к исполняемому программному коду.

2. Алгоритм создания вычислительных машин с процессором, исполняющим обфусцированный код с псевдослучайной архитектурой, исполняющих защищенный от обратного проектирования код программ.

3. Алгоритм защиты исполняемого кода, основанный на запутывающих преобразованиях, которые используют изменение потока управления и внесения избыточности.

Апробация диссертации и информация об использовании ее результатов

Результаты исследований, вошедшие в диссертацию, докладывались и обсуждались на LXXIV Международной научно-практической конференции «Молодой исследователь: вызовы и перспективы» (г. Москва, Российская Федерация, 2018 г.), LXVI Международной научно-практической конференции «Научное сообщество студентов XXI столетия. «ТЕХНИЧЕСКИЕ НАУКИ» (г. Новосибирск, Российская Федерация, 2018 г.), LXXV Международной научно-практической конференции «Молодой исследователь: вызовы и перспективы» (г. Москва, Российская Федерация, 2018 г.).

Публикации

Изложенные в диссертации основные положения и выводы опубликованы в 4 печатных работах. В их числе 1 статья в журнале, 3 статьи в сборниках материалов научных конференций.

Общий объем публикаций по теме диссертации составляет 16 страниц.

Структура и объем работы

Диссертация состоит из введения, общей характеристики работы, трех глав, заключения, библиографического списка и приложений.

В первой главе приведен анализ современных подходов к созданию средств защиты прикладного программного обеспечения от компьютерных атак, организованных с целью обратного проектирования, несанкционированного копирования и модификации исполняемых кодов.

Во второй главе разработаны теоретические основы реализации предложенного алгоритма защиты с помощью формирования байт-кода, исполняемого с помощью виртуальных машин с псевдослучайной архитектурой ПИОК, и генерации интерпретатора ПИОК при помощи автоматных моделей. Представлена практическая реализация разработанного алгоритма запутывающих преобразований.

В третьей главе производится исследование разработанного алгоритма защиты исполняемого программного кода на основе запутывающих преобразований, а также проводится анализ эффективности разработанного метода при противодействии атакам, основанным на технологиях обратного проектирования.

В приложении представлены публикации автора и акт внедрения.

Общий объем диссертационной работы составляет 109 страниц. Из них 81 страница основного текста, 17 иллюстраций на 13 страницах, 29 таблиц на 20 страницах, библиографический список из 54 наименований на 4 страницах, список собственных публикаций соискателя из 4 наименований на 1 странице, 4 приложений на 28 страницах.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во **введении** рассмотрено современное состояние проблемы защиты программного кода от обратного проектирования, указаны основные направления исследований, проводимых по данной тематике, а также описано обоснование актуальности темы.

В **общей характеристике работы** показана актуальность проводимых исследований, степень разработанности проблемы, сформулированы цель и задачи диссертации, обозначена область исследований, научная (теоретическая и практическая) значимость исследований, а также апробация работы.

В первой главе приведен анализ современных подходов к созданию средств защиты прикладного программного обеспечения от компьютерных атак, организованных с целью обратного проектирования, несанкционированного копирования и модификации исполняемых кодов.

Указываются преимущества использования методов, обеспечивающих защиту от различных средств статического и динамического анализа исполняемого кода с помощью запутывающих преобразований. Важной характеристикой таких преобразований является оценка количества информации по Колмогорову, которая характеризует длину программы реализующей целевой алгоритм. Показано, что сложность обратного проектирования таких алгоритмов с помощью кода исполняемой программы, которая модифицирована в процессе применения запутывающих преобразований, количественно выражается как относительная мера увеличения числа используемых машинных команд. Анализируется возможность применения метода защиты программного кода, основанного на использовании запутывающих преобразований применительно только части исполняемой программы.

Анализируются недостатки существующих методов обфускации и показано, что для их преодоления можно использовать процедуры поэтапного преобразования исполняемого кода, инвариантом которого является семантика целевого алгоритма.

В главе делается вывод о перспективности, актуальности и практической значимости применения метода поэтапного преобразования, а также выдвигаются требования к программным и пользовательским интерфейсам. В конце главы приведена постановка задачи исследования, которая базируется на анализе исполняемого кода с помощью инструментальной системы трансформации и оптимизации программ *LLVM (Low Level Virtual Machine)*, формировании списка всех доступных для защиты подпрограмм, находящихся в объектном файле, анализа набора инструкций, используемых подпрограммой, а также выбора критического блока машинных инструкций, которые трансформируется в байт-код команд ПИОК. При этом замещение защищаемого исполняемого кода происходит на уровне вызова удаленной подпрограммы интерпретатором ПИОК с параметрами, идентичными тем, которые использовались в исходном исполняемом коде с добавлением адреса точки входа в байт-код защищенной подпрограммы.

Во второй главе разработаны теоретические основы реализации предложенного метода защиты с помощью формирования байт-кода, исполняемого с помощью виртуальных машин с псевдослучайной архитектурой ПИОК, и генерации интерпретатора ПИОК при помощи автоматных моделей, основанных на сетях Петри.

Предложена модель угроз обратного проектирования исполняемого кода при помощи анализа и отладки. Она состоит трех компонентов <*S, O, A*>, где *S* – это субъект атаки, *O* – объект атаки, *A* – атакующее действие.

К субъектам атаки *S* относят:

− множество автоматизированных средств отладки кода (*s*): отладчики, песочницы;

− множество полуавтоматических средств анализа кода (*ss*): дизассемблеры, анализаторы графа достижимости;

− множество интеллектуальных средств анализа (*is*): хакер, семантическая сеть, генетические алгоритмы.

Объекты атаки *O* включают в себя следующие компоненты:

− алгоритм, который реализует исполняемый код (*m*). Обычно он недоступен для атакующего воздействия, так как кодируется программистом в код на языке высокого уровня (ЯВУ) и потому не предается в недоверенную вычислительную среду;

− код на ЯВУ (*t*). Обычно он также недоступен для атакующего воздействия, так как транслируется в машинный код инструментального процессора компилятором и потому не предается в недоверенную вычислительную среду;

− машинный код (*p*). Непосредственно передается в недоверенную вычислительную среду и, потому, непосредственно доступен для атаки.

Атакующее действие *A* состоят из следующих операций:

− операция (*o*) по сбору данных о переменных, адресах областей памяти исполнения кода и их порядка выполнения;

− преобразование (*c*) кода к текстовому формату;

− оперативное восстановление (*r*) алгоритма.

В результате модель угроз обратного проектирования может быть редуцирована до кортежа <*S, p, A*>. В дальнейшем объектом рассмотрения являются компьютерные атаки на исполняемый машинный код, для защиты от которых предложен новый вид запутывающих преобразований, формирующих байт-код для ПИОК с псевдослучайной архитектурой.

Систему команд ПИОК предложено формировать так, чтобы каждая инструкция имела: символическое имя, код (уникальное число, однозначно характеризующее действие) и набор аргументов-операндов, которые параметризируют действие. В результате набор машинных команд ПИОК можно сгенерировать таким образом, чтобы количество информации по Колмогорову в защищаемой сегменте исполняемого кода увеличилось: $Q_p > Q_u$, где Q_p– количество информации в защищенном коде, а Q_u – количество информации в незащищенном коде, а сама структура команд имела псевдослучайный характер.

Эффект увеличения количества информации по Колмогорову достигается за счет таких факторов как: добавление обфусцирующих вычислений, изменение набора исполняемых инструкций и включение в защищаемый сегмент кода интерпретатора ПИОК, который становится неотъемлемой частью исполняемого алгоритма (рисунок 1).

Последовательность команд ПИОК, реализующая целевой алгоритм в дальнейшем рассматривается как виртуальная программа, в которой обфускации подвержен как целевой алгоритм, так и исполняемый байт-код.

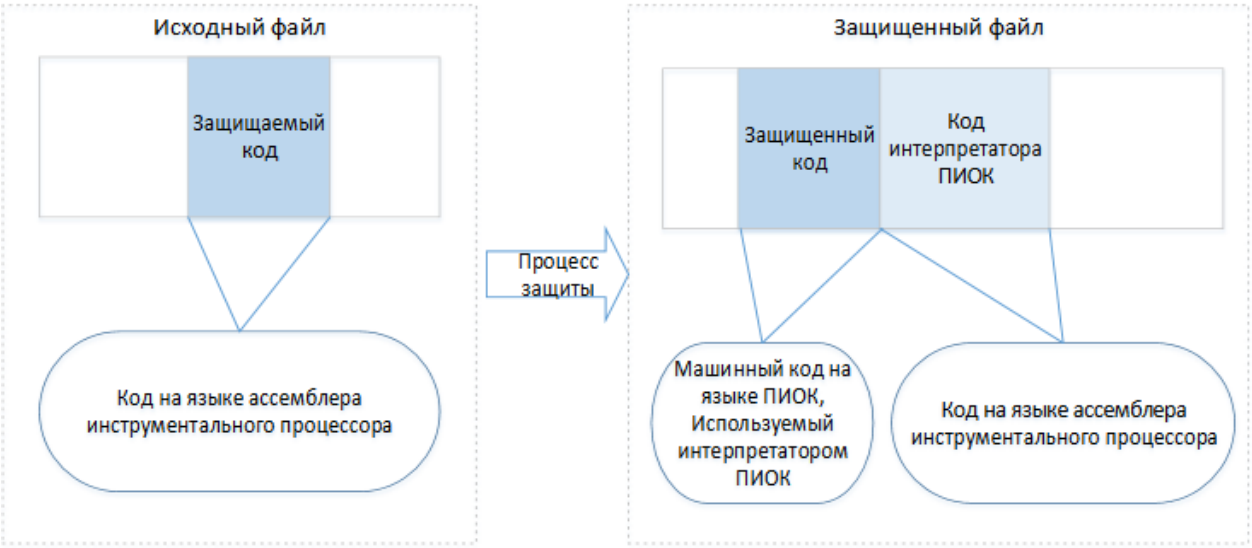

Рисунок 1 – Пример незащищенного (исходного) и защищенного программного кода

С учетом того, что для анализа защищаемого кода используется инструментальная система *LLVM*, для реализации разработанного алгоритма обфускации предложено использовать архитектуру ПИОК программным счетчиком. Таким образом, в процессе защиты кода для его реализации используется ПИОК, обладающий случайным для данной реализации набором машинных команд, что требует создания специализированного программного обеспечения для каждого конкретного ПИОК, усложняя процесс обратного проектирования целевых алгоритмов. Так как определенный блок инструкций ассемблера инструментального процессора может быть трансформирован в различные блоки инструкций для каждой виртуальной машины, то преобразование, основанное на предложенном методе обфускации, затрудняет процесс дизассемблирования программ, повышая доверенность среды выполнения команд виртуальной программы.

Преимуществом предложенного подхода к защите исполняемого программного кода при помощи обфускации по отношению с существующим аналогам, используемым в промышленных решениях, является то, что в зависимости от архитектуры ПИОК, операнды опкода могут быть значениями регистров, значениями стека, ссылками на ячейки памяти или непосредственными значениями.

Алгоритм защиты исполняемого кода с использованием запутывающих преобразований на основе сетей Петри может применяться к каждому конкретному базовому блоку защищаемой программы в отдельности. Реализация алгоритма защиты исполняемого кода с использованием обфускации при помощи сетей Петри представлена на рисунке 2.

Рисунок 2 – Алгоритм защиты программного кода при помощи сети Петри

На первом этапе исходный код необходимо преобразовать. Для этого машинный код извлекается из объектных файлов. Затем извлеченный код обрабатывается с помощью дизассемблера. Дизассемблированные инструкции переводятся на язык *Low Level Virtual Machine* ассемблера (*LLVM IR*). Ссылки на переменные не меняются, а для ячеек регистров процессора и регистра флагов, используются статические переменные. При помощи стандартных оптимизаций *LLVM* упрощается полученный код.

Второй этап включает в себя выбор набора защищаемых значений. Защищаемыми значениями могут быть константы, адреса констант и значения ячеек памяти, которые являются начальными для базового блока. Так же необходимо выбрать число раундов сети Петри, при этом оно не должно сильно превышать число защищаемых значений. Для увеличения стойкости

запутывающих преобразований часть пометок можно выбрать случайным образом.

На третьем этапе необходимо выбрать одну из заранее придуманных сетей Петри. Для поддержки базовых блоков с большим количеством защищаемых значений, при помощи операции масштабирования, сеть Петри приводится к адекватному размеру.

На четвертом этапе определяются начальные пометки сети Петри при помощи решения системы диофантовых уравнений в кольце вычетов по длине машинного слова инструментального процессора.

После выполнения четвертого этапа имеется базовый блок со вставленными инструкциями расчёта раундов сети Петри между инструкциями защищаемого алгоритма, заданный в виде *LLVM IR*. Для генерации кода создается фиктивный модуль компиляции, который содержит только защищаемый код, полученный после третьего этапа. Процесс получения выходного файла зависит от того, находилась ли используемая функция в отдельной секции объектного файла или нет. Если функция находилась в отдельной секции, то из исходного объектного файла удаляется секция с этой функцией, а в конец файла дописывается секция с защищенной функцией. Если используемая функция не находилась в отдельной секции объектного файла, то тело оригинальной функции заменяется на псевдослучайную последовательность байтов, ссылки в ее прежней секции на удаленную часть преобразуются в символы со сгенерированными именами, а с использованной функцией дописывается в новую секцию, в конец.

В третьей главе производится исследование разработанного алгоритма защиты исполняемого программного кода на основе запутывающих преобразований, реализуемых с помощью вложенной виртуализации и применении сетей Петри, а также проводится анализ эффективности разработанного метода при противодействии атакам, основанным на технологиях обратного проектирования.

Исследования проводятся на основе разработанного во второй главе подхода к построению ПИОК, содержащий набор инструкций достаточный для выполнения целевого алгоритма, реализованного с помощью исполняемого кода инструментального процессора

Приводятся оценки, указывающие на количественные и качественные отличия между представлениями байт-кода исходного и защищенного исполняемых файлов, что затрудняет процесс анализа алгоритма, команд и операндов.

При формировании оценок, характеризующих алгоритмическую сложность решения задач обратного проектирования учитывается то, что при реализации предложенного алгоритма защиты исполняемого кода генерируется виртуальная машина с псевдослучайной архитектурой, а эффективных программных инструментов для динамического анализа таких исполняемых байт-кодов не существует.

Показано, что применяемый метод существенно повышает алгоритмическую сложность процесса обратного проектирования при использовании

процессорных плагинов, что делает их применение критически зависимым от особенностей псевдослучайной ПИОК. В случае использования ПИОК, одним из методов обратного проектирования является частичный анализ машинного кода интерпретатора ПИОК с последующим частотным анализом опкодов ПИОК. Поэтому, кроме увеличения меры количества информации по Коломогорову, эффективность предложенного метода рассматривается и с помощью частотного анализа защищенного байт-кода (таблица 1).

Таблица 1 – Частотный анализ защищенного байт-кода

Код операции	Относительная частота (защищенный код)	Относительная частота (незащищенный код)
00	0,26	0,01
CF	0,14	0,01
44	0,14	0,01
45	0,08	0,02
A9	0,06	0,00

Показано, что частота появления опкода, соответствующего определенной команде, в длинных текстах разных программ не изменяется, что создает дополнительный уровень защищенности, затрудняя применение статистических методов отладки в том числе на основе анализа парных корреляций.

ЗАКЛЮЧЕНИЕ

Основные научные результаты диссертации

1. Разработана модель угроз, связанных с использованием технологий обратного проектирования исполняемого программного кода, основанных на динамическом анализе и статическом исследовании исполняемого кода. Разработанная модель состоит из субъекта атаки, машинного кода, который непосредственно передается в недоверенную вычислительную среду и доступен для атаки, и атакующего действия. На основе данной модели был разработан алгорим защиты исполняемого кода от исследования.

2. Разработан алгоритм защиты от компьютерных атак обртного проектирования прикладного программного обеспечения, основанный на замещении выбранного критически важного сегмента кода защищенным программным объектом. Данный алгоритм преобразует код исходной программы в байт-код виртуальной машины и далее выполняет преобразованный код. Виртуализированные части кода исполняются интерпритатором виртуальной машины без трансляции в оригинальный машинный код. В результате этого обратное проектирование требует изучения архитектуры виртуальной машины, что требует больших временных затрат и увеличивает надежность програмного обеспеспечения.

3. Разработан алгоритм защиты исполняемого кода, основанный на запутывающих преобразованиях, которые используют изменение потока

управления и внесения избыточности. Разработанный алгоритм алгоритм защищает ассемблерный код при генерации кода целевой платформы из кода в формате промежуточного представления LLVM. Защита осуществляется при помощи вставки мусорного кода, в результате чего происходит преобразование исполняемого кода программы в вид, сохраняющий ее функциональность, но затрудняющий анализ, понимание алгоритмов работы и модификацию при декомпиляции. Так же для защиты исполняемого кода в алгоритм включено метаморфирование кода, и вставка инструкций для создания полиморфного кода. Этот подход еще больше усложняет процесс динамического анализа.

Рекомендации по практическому использованию результатов

Полученные результаты внедрены в учебный процесс на кафедре проектироваания информационно−компьютерных систем учреждения образования “Белорусский государственный университет информатики и радиоэлектроники в учебный курс “Компьютерные сети в электронных системах безопасности”.

СПИСОК ПУБЛИКАЦИЙ СОИСКАТЕЛЯ

Статьи в журналах

1. Ероховец Н.Ю. Алгоритм защиты исполняемого кода при помощи виртуальных машин / Н.Ю. Ероховец // Студенческий: электронный научный журнал/ Новосибирск, Российская Федерация – 2018. – С. 49-51

Тезисы конференций

2. Ероховец Н.Ю. Анализ методов защиты программного кода от обратного проектирования / Н.Ю. Ероховец // Молодой исследователь: вызовы и перспективы: сб. ст. по материалам LXXIV Международной научно-практической конференции «Молодой исследователь: вызовы и перспективы». – № 21(74). – М., Изд. «Интернаука», 2018. – С. 35-39.

3. Ероховец Н.Ю. Алгоритм запутывания машинного кода для защиты от исследования / Н.Ю. Ероховец // Молодой исследователь: вызовы и перспективы: сб. ст. по материалам LXXV Международной научно-практической конференции «Молодой исследователь: вызовы и перспективы». – № 22(75). – М., Изд. «Интернаука», 2018. – С. 330-333

4. Ероховец Н.Ю. Алгоритм защиты программного кода при помощи запутывающих преобразований / Н.Ю. Ероховец // Научное сообщество студентов XXI столетия. ТЕХНИЧЕСКИЕ НАУКИ: сб. ст. по материалам LXVI Международной научно-практической конференции, Новосибирск, Российская Федерация – 2018. – С. 79-83

РЭЗЮМЭ

Ерахавец Наталля Юр’еўна

Алгарытм абароны праграмнага кода ад дынамічнага і статычнага аналізу

Ключавыя словы: віртуальная машына, абфускацыя.

Мэта працы: Распрацоўка алгарытму абароны выкананага праграмнага кода ад кампутарных нападаў зваротнага праектавання на аснове дынамічнага і статычнага аналізу дадзеных.

Атрыманыя вынікі і іх навізна: выкананы аналіз сучасных падыходаў да стварэння сродкаў абароны прыкладнога праграмнага забеспячэння ад кампутарных нападаў, арганізаваных з мэтай зваротнага праектавання, несанкцыянаванага капіявання і мадыфікацыі выкананых кодаў. Выяўлена, што ў цяперашні час у айчынных і замежных крыніцах недастаткова асветлены пытанне абароны праграмнага забеспячэння ад зваротнага праектавання на аснове віртуальных машын; распрацаваны тэарэтычныя асновы рэалізацыі прапанаванага алгарытму абароны з дапамогай фарміравання байт-кода, выкананага з дапамогай віртуальных машын з псеўдавыпадковых архітэктурай ПВОК, і генерацыі інтэрпрэтатара ПВОК пры дапамозе аўтаматных мадэляў. Прадстаўлена практычная рэалізацыя распрацаванага алгарытму заблытвае пераўтварэнняў; выраблена вы ка-прытрымліванне распрацаванага алгарытму абароны выкананага праграмнага кода на аснове заблытвае пераўтварэнняў, а таксама праводзіцца аналіз эфектыўнасці распрацаванага метаду пры супрацьдзеянні нападам, асновы бела-ванным на тэхналогіях зваротнага праектавання.

Ступень выкарыстання: вынікі ўкаранёны ў навучальны працэс на кафедры праектавання інфармацыйна-камп'ютэрных сістэм ўстановы адукацыі «Беларускі дзяржаўны універсітэт інфарматыкі і радыёэлектронікі» у навучальны курс «Кампутарныя сеткі ў электронных сістэмах бяспекі»

Вобласць ужывання: абарона праграмнага забеспячэння.

РЕЗЮМЕ

Ероховец Наталия Юрьевна

Алгоритм защиты программного кода от динамического и статического анализа

***Ключевые слова*:** виртуальная машина, обфускация.

***Цель работы*:** разработка алгоритма защиты исполняемого программного кода от компьютерных атак обратного проектирования на основе динамического и статического анализа данных.

***Полученные результаты и их новизна*:** выполнен анализ современных подходов к созданию средств защиты прикладного программного обеспечения от компьютерных атак, организованных с целью обратного проектирования, несанкционированного копирования и модификации исполняемых кодов. Выявлено, что в настоящее время в отечественных и зарубежных источниках недостаточно освещен вопрос защиты программного обеспечения от обратного проектирования на основе виртуальных машин; разработаны теоретические основы реализации предложенного алгоритма защиты с помощью формирования байт-кода, исполняемого с помощью виртуальных машин с псевдослучайной архитектурой ПИОК, и генерации интерпретатора ПИОК при помощи автоматных моделей. Представлена практическая реализация разработанного алгоритма запутывающих преобразований; произведено исследование разработанного алгоритма защиты исполняемого программного кода на основе запутывающих преобразований, а также проводится анализ эффективности разработанного метода при противодействии атакам, основанным на технологиях обратного проектирования.

***Степень использования*:** результаты внедрены в учебный процесс на кафедре проектирования информационно-компьютерных систем учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» в учебный курс «Компьютерные сети в электронных системах безопасности».

***Область применения*:** защита программного обеспечения.

SUMMARY

Erohovets Natalia Yurevna

Algorithm for protection of program code from dynamic and static analysis

***Keywords*:** virtual machine, obfuscation.

***The object of study*:** The purpose of the work: development of algorithm for protection of executable program code from computer attacks of reverse design based on dynamic and static data analysis.

***The results and novelty*:** the analysis of modern approaches to creation of means of protection of applied software against computer attacks organized for the purpose of reverse engineering, unauthorized copying and modification of executable codes is performed. It has been revealed that at the present time in domestic and foreign sources the issue of software protection against reverse engineering based on virtual machines is not sufficiently covered; The theoretical foundations for implementing the proposed protection algorithm with the help of bytecode generation executed with virtual machines with pseudo-random architecture of PEDCs and the generation of the PEOC interpreter using automatic models are developed. The practical implementation of the developed algorithm for confusing transformations is presented; The research of the developed algorithm of protection of executable program code on the basis of confusing transformations is carried out, as well as the analysis of the efficiency of the developed method is carried out when counteracting attacks based on reverse engineering technologies.

***Degree of use*:** the results are implemented in the educational process at the department of designing information and computer systems of the educational institution "Belarusian State University of Informatics and Radioelectronics" in the training course "Computer networks in electronic security systems".

Sphere of application: software protection.