ОТЗЫВ ОФИЦИАЛЬНОГО ОППОНЕНТА

на диссертационную работу Нурмухаметова Алексея Раисовича

«Применение диверсифицирующих преобразований для защиты от эксплуатации уязвимостей»,

представленную к защите на соискание учёной степени кандидата технических наук по специальности 05.13.11 — «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей»

Актуальность темы

Диссертационная работа А.Р. Нурмухаметова посвящена разработке и применению *программных* методов защиты от эксплуатации уязвимостей. Задача создания таких методов **актуальна** по нескольким причинам. Во-первых, современные программные комплексы состоят из колоссального количества кода. Такой объём кода невозможно досконально проверить на предмет отсутствия ошибок и уязвимостей. Тестирование, использование методов статического анализа и фаззинга позволяют найти некоторые ошибки на этапе разработки, но не гарантируют их полного отсутствия. На протяжении многих лет показатель числа ошибок на тысячу строк кода в программах на языках C/C++ остается неизменным. Это также подтверждается статистикой из базы данных общеизвестных уязвимостей CVE. Во-вторых, микропроцессорные архитектуры, в которые изначально заложено небезопасное исполнение программ, добавляют крайне мало средств, повышающих защиту от эксплуатации уязвимостей, а используемые программные средства не обеспечивают полной защиты. В-третьих, методы эксплуатации уязвимостей постоянно совершенствуются и требуют дополнительной адекватной программной защиты.

Структура работы

Диссертационная работа содержит 140 страниц и состоит из введения, шести глав, заключения, списка литературы из 72 наименований, и пяти приложений.

Во введении обосновывается актуальность темы исследования, определяются цель и задачи, выделяются научная новизна и основные положения работы, выносимые на защиту.

В первой главе объясняется цель работы. В ней на серии примеров рассматриваются различные методы эксплуатации уязвимостей, в частности, уязвимости повторного использования кода библиотек и использования специальных фрагментов программного кода. В ней также приводится подробный обзор существующих подходов к решаемой задаче защиты от методов эксплуатации данных видов уязвимостей, отмечаются их недостатки и рассматриваются возможные способы их преодоления. Основными способами обхода рандомизации размещения адресного пространства (ASLR) являются: подбор и перебор адресов, использование утечки информации о размещении кода программы или библиотеки в памяти процесса, использование нерандомизированных областей памяти для построения атаки, частичная перезапись указателей на код. Намечаются пути решения некоторых из этих проблем с помощью диверсифицирующих преобразований и мелкозернистой рандомизации адресного пространства.

Во второй главе описывается метод, использующий диверсифицирующие преобразования внутри компилятора для генерации большого количества уникальных образов приложения. Приводится модель атаки на диверсифицированную популяцию исполняемых файлов и её преимущества с точки зрения безопасности. Описываются диверсифицирующие преобразования, реализованные внутри компиляторной инфраструктуры GCC и LLVM. Приводится анализ влияния диверсифицирующих преобразований на производительность на наборе тестов SPEC CPU 2006, который подтверждает, что данный метод практически не снижает производительность, но снижает вероятность эксплуатации уязвимостей.

В третей главе описывается метод мелкозернистой рандомизации адресного пространства программы при запуске на исполнение, который позволяет перемешивать код на уровне функций. Мелкозернистая рандомизация состоит из двух этапов. В первую очередь во время компоновки исполняемого образа приложения происходит сохранение информации о границах функций и требующих исправления ссылках в отдельной секции исполняемого файла формата ELF. Затем, загрузчик переставляет в памяти процесса функций местами,

используя информацию об их границах из этой секции. После перестановки происходит исправление ссылок. Производится анализ влияние мелкозернистой рандомизации на время загрузки приложения, на размер исполняемых образов, на производительность и общее потребление памяти целой ОС. Отмечается, что этот метод незначительно влияет на скорость исполнения (немного замедляя отдельные программы и даже немного ускоряя другие), в допустимых пределах увеличивает время загрузки программы при запуске на исполнение, но заметно увеличивает размер кода за счет дополнительных секций и за счет дублирования кода динамических библиотек для каждого приложения.

В четвёртой главе излагается метод оценки эффективности реализованных диверсифицирующих преобразований. Он состоит из оценки количества выживших гаджетов (фрагментов повторно используемого кода), из оценки работоспособности эксплойтов на синтетически внедрённых в реальные программы уязвимостях, из оценки работоспособности эксплойтов на реальных примерах уязвимостей. Это важнейшая часть работы. Приводится статистика по количеству выживших гаджетов в зависимости от размера диверсифицированной популяции исполняемых файлов. Приводится экспериментальная статистика по эффективности противодействия атакам повторного использования кода в виде цепочек возвратного-ориентированного программирования. Результаты показывают, что вероятность успешного срабатывания эксплойта не превышает 0.3 процента и стремится к нулю с увеличением сложности эксплойта. Важным результатом является также анализ конкретных уязвимостей из базы данных NIST SVE, который подтверждает эффективность предлагаемых методов защиты путем диверсифицирующих преобразований кода.

В пятой главе производится математическая оценка вероятности функции остаться на своём месте при случайной перестановке. Выводится формула для такой вероятности, из которой следует, что такая вероятность сильно больше для функций, расположенных ближе к краю файла. Данная неоднородность распределения могла бы быть использована при конструировании атаки, поэтому была исправлена в реализации мелкозернистой рандомизации путём сдвига базы образа программы как целого.

В шестой главе приводится описание программной реализации дополнительной системы защиты от эксплуатации уязвимостей для ОС семейства Linux на основе разработанных в диссертации методов. Разработанная система применима в

2. Предложен новый метод мелкозернистой рандомизации программы при запуске, который позволяет получать уникальный образ адресного пространства процесса при каждом запуске, что осложняет атаку на приложения методами повторного использования кода, а также исключает утечку адресного пространства разделяемых библиотек через скомпрометированные процессы.
3. Предложен новый метод оценки качества реализованных преобразований, который опирается на подсчёт количества выживших гаджетов и использует экспериментальные проверки посредством внедрения синтетических уязвимостей в код реальных приложений и эксплуатации их модельными примерами атак.

Практическая значимость

Все предложенные в работе алгоритмы реализованы в промышленных операционных системах CentOS 7 и Debian 10. На них получены сертификаты о государственной регистрации программ для ЭВМ. Данные методы представляют практическую ценность и используются в промышленных ОС для усиления защиты от эксплуатации уязвимостей.

В диссертационной работе следует отметить ряд **недостатков**:

1. Описываемый в разделе 3.2.1 формат дополнительной секции расходует по 16 байт на каждую функцию и ссылку, что приводит к увеличению размера образа приложения на диске. В диссертационной работе не хватает исследования вопроса о том, можно ли упаковать данные в этой секции более плотно.
2. Реализация метода мелкозернистой рандомизации программы, повышающей ее безопасность, требует использования небезопасной самомодификации кода при его загрузке в память.
3. Все приведенные в работе примеры эксплуатации рассматриваемых уязвимостей повторного использования кода эксплуатируют известную уязвимость нарушения границ объектов в стеке.
4. В обзоре известных методов не упомянута технология KARL в OpenBSD, которая осуществляет рандомизацию ядра ОС, статически перекомпоновывая объектные файлы при каждой его перезагрузке.

рамках целой ОС, обеспечивает дополнительную защиту от атак повторного использования кода и при этом совместима с уже включенными в нее средствами защиты для достижения взаимно усиливающего эффекта.

В заключении перечисляются основные результаты работы, а также обозначаются направления дальнейшего развития рассматриваемых в работе методов мелкозернистой рандомизации.

В приложениях приводятся копии свидетельств на программные компоненты, включенные в работу, копия акта внедрения и подробные результаты сравнения производительности рассматриваемых в работе методов на пакете SPECcpu2017.

Основные результаты, полученные в ходе диссертационного исследования:

1. Разработан метод применения диверсифицирующих преобразований на уровне промежуточного представления компилятора для генерации уникальных исполняемых образов.
2. Разработан метод мелкозернистой рандомизации программ при запуске, позволяющий получать уникальные состояния адресного пространства при каждом запуске, препятствующие эксплуатации уязвимости повторного использования кода.
3. Разработан метод оценки эффективности защиты от эксплуатации рассматриваемых в работе уязвимостей.
4. Предложенные методы реализованы в качестве дополнительной системы защиты от эксплуатации уязвимостей для ОС семейства Linux.

Результаты, выносимые на защиту опубликованы в шести публикациях в рецензируемых журналах, рекомендованных ВАК, две из которых индексируются Scopus.

Научная новизна и достоверность работы

1. Диссертационная работа описывает новый подход к распространению пользователям уникальных копий исполняемых образов приложений, которые сгенерированы компилятором, снабжённым набором диверсифицирующих преобразований. Данный подход позволяет усложнить планирование одинаковой атаки на всех пользователей.

Перечисленные недостатки не влияют на общую положительную оценку работы.

Диссертационная работа А.Р. Нурмухаметова является законченным научным исследованием. Основное содержание диссертации отражено в опубликованных диссертантом статьях, доложено на научных конференциях. Практическая ценность подтверждается результатами применения разработанных методов.

Автореферат полно и правильно отражает содержание диссертационной работы.

Вывод

Диссертационная работа соответствует всем требованиям ВАК, предъявляемым к диссертациям на соискание учёной степени кандидата технических наук, а её автор, Нурмухаметова Алексей Раисович, заслуживает присуждения ему учёной степени кандидата технических наук по специальности 05.13.11 — «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей».

Официальный оппонент
кандидат технических наук, начальник отделения «Системы программирования» Публичного акционерного общества «Институт электронных управляющих машин им. И. С. Брука», Российская федерация, 119334, Москва, ул. Вавилова, д. 24,

В.Ю. Волконский

30 апреля 2021 г.