УДК 004.056.53

ИСПОЛЬЗОВАНИЕ МЕТОДОВ МЕТАПРОГРАММИРОВАНИЯ ЯЗЫКА C++ ПРИ РАЗРАБОТКЕ АЛГОРИТМОВ ОБФУСКАЦИИ

Галкин Д.А. Университет ИТМО
Научный руководитель – к.т.н., с.н.с., доцент Канжелев Ю.А.
Университет ИТМО

Работа выполнена в рамках темы ВКР «Разработка алгоритма обфускации программ на языке C++ с использованием метапрограммирования»

В работе рассмотрены методы применения метапрограммирования C++ для повышения защищенности бинарных файлов посредством применения алгоритмов обфускации. Освещены преимущества данного метода и проведена оценка эффективности.
Ключевые слова: обфускация, C++, метапрограммирование, конфиденциальность, программное обеспечение, реверс-инжиниринг

Введение

Необходимость защиты авторского права, интеллектуальной собственности, а также целостности и конфиденциальности в целом, является одной из важнейших задач при обеспечении информационной безопасности и затрагивает сферы коммерческой, научной и военной деятельностей. Программное обеспечение (ПО) подвергается различного рода угрозам, например, реверс-инжинирингу. Задача алгоритмов обфускации состоит в повышении устойчивости ПО против такого рода атак. Несмотря на актуальность проблемы, существующие обфускаторы обеспечивают недостаточный уровень защищенности и дороги в использовании.

Таким образом целью данной работы является разработка алгоритма обфускации с использованием метапрограммирования, способного обеспечить надлежащий уровень защиты ПО.

Метапрограммирование C++

Метапрограммирование можно описать как «программирование программы». Другими словами, программист может написать код, который будет выполнен компилятором для генерации нового кода, реализующего функциональность, которая действительно требуется. Более того, метапрограммирование можно считать полноценным подъязыком, ведь оно является полным по Тьюрингу и похоже на функциональные языки программирования.

Основу метапрограммирования C++ составляют следующие возможности языка:

1. Шаблоны.
2. Принцип SFINAE (неудавшаяся подстановка – не ошибка).
3. Вычисления во время компиляции.

Методы обфускации кода

Задача алгоритмов обфускации состоит в манипулировании кодом таким образом, чтобы он стал более трудным для анализа человеком и автоматизированными инструментами, сохраняя при этом исходную функциональность. Выделяют следующие методы обфускации кода:

1. Лексические преобразования.
2. Трансформация потока управления.
3. Обфускация данных.
4. Превентивная обфускация.
5. Обфускация с использованием самомодифицирующегося кода.

Анализ ПО и оценка эффективности обфускаторов

Методы анализа программ нацелены на получение информации о поведении и структуре программы. Зачастую анализ может предоставить лишь приблизительную оценку, поэтому основная задача исследователя ПО заключается в извлечении максимальных

сведений из ограниченного набора ресурсов. Выделяют три основных подхода к проведению анализа программного обеспечения:

1. Статический.
2. Динамический.
3. Синтаксический.

В целях оценки эффективности алгоритмов обфускации должны быть установлены метрики, которые способны измерить сложность и защищенность результирующего ПО и, следовательно, эффективность разработанного алгоритма. Показатели оценки эффективности алгоритмов обфускации были сформированы на основе работы Колберга, а именно:

1. Действенность.
2. Устойчивость.
3. Стоимость.

Описание предлагаемого алгоритма обфускации и оценка эффективности

В основу разработанного алгоритма обфускации легли методы обфускации данных и трансформации потока управления. Лексические преобразования кода не были задействованы не только в силу их низкой эффективности, но и вследствие того, что цель работы заключается в повышении защищенности бинарного исполняемого файла и подразумевается, что злоумышленник не будет располагать исходным кодом программы. Кроме того, следует отметить, что методы превентивной обфускации и самомодифицирующегося кода также не были включены в текущую версию алгоритма в силу значительной сложности реализации. Реализация этих мер защиты может стать мотивацией для будущих исследований.

Разработанный алгоритм предоставляет возможность генерации бинарных файлов с уникальной сигнатурой, что позволяет предотвратить поиск шаблона обфускации и усложняет разработку деобфускатора.

Краткое описание алгоритма:

1. Построение графа потока управления по исходному коду программы.
2. Обфускация данных: замена встроенных типов криптографическими аналогами.
3. Трансформация потока управления: введение непрозрачных предикатов, «мёртвого» и недостижимого кода.

Применение и анализ эффективности разработанного алгоритма обфускации

Применение обфусцирующих преобразований вызывает незначительное увеличение размера потребляемой оперативной памяти, увеличение времени выполнения программы прямо пропорциональное увеличению уровню обфускации. Кроме того, в результате применения разработанного алгоритма обфускации происходит заметный рост размера исполняемого файла. Тем не менее если брать в расчёт современное ПО, размер которого может достигать нескольких гигабайт, увеличение размера критического кода исполняемого файла на несколько мегабайт представляется допустимым.

Заключение

В ходе работы были рассмотрены методы применения метапрограммирования C++ для повышения защищенности бинарных файлов посредством применения алгоритмов обфускации. Была выполнена программная реализация и проведена оценка эффективности рассмотренных методов. Применение разработанного алгоритма для обфускации ПО позволит обеспечить необходимый уровень защиты от пиратства и несанкционированного доступа.