# Способ повышения эффективности обфускации компьютерного кода при деструктивном воздействии на основе синтеза программ

В. О. Даренских, email: darenskih.vadim@yandex.ru[1]
А. А. Дорофеев, email: andrei3267@gmail.com[1]
Я. С. Тодояков, email: todoyakovs@mail.ru[1]
М. В. Соколов, email: SaSa_t@mail.ru[1]

[1] Краснодарское высшее военное училище им. С. М. Штеменко

***Аннотация.*** *В данной работе рассматривается способ повышения эффективности обфускации компьютерного кода при деструктивном воздействии на основе синтеза программ, а также предложен подход, устойчивый к атакам на основе синтеза программ, и определены пределы синтеза в эмпирическом исследовании.*

***Ключевые слова:*** *обфускация, деобфускация, синтаксис, программный синтез, ГОСТ Р 34.11-2012.*

## Введение

Опора на синтаксическую сложность в современных схемах обфускации и широкий арсенал передовых методов деобфускации вызвали интерес к дальнейшим исследованиям по созданию более устойчивых схем, направленных на препятствование этим автоматическим анализам. Были выдвинуты предложения по затруднению анализа "испорченности" [1-4] и неэффективности символьного выполнения [5-9]. Например, последнее может быть достигнуто путем инициирования исследования путей для механизма символьного выполнения путем искусственного увеличения числа путей для анализа. Появились и другие перспективные схемы обфускации, включая смешанные булево-арифметические (MBA) выражения, которые предлагают модель для сложного кодирования произвольных арифметических формул. Выражения представлены в области, которая нелегко поддается упрощению, эффективно скрывая фактические семантические операции. Обычно автоматизированные подходы к деобфускации MBA основаны на символическом упрощении; эти методы опираются на определенные предположения о структуре выражения, что делает их непригодными для упрощения таких выражений в общем случае.

Методы деобфускации на основе синтеза программ являются основными методами для автоматического анализа обфусцированного кода.

## 1. Синтез программ

В отличие от других методов, основанных на синтаксическом анализе обфусцированного кода, подходы, основанные на программном синтезе, функционируют на семантическом уровне. В рамках таких методов код рассматривается, как черный ящик, а его восстановление основано на анализе данных ввода-вывода. В таких подходах как SYNTIA и XYNTIA реализован стохастический алгоритм, с помощью которого производится поиск выражения, обладающего эквивалентным поведением [10]. Другие подходы, например, QSYNTH, основаны на перечислительном синтезе: они вычисляют большие таблицы поиска выражений, которые используются для упрощения частей выражения, уменьшая его общую комплексность. При деобфускации кода эти подходы используются для упрощения синтаксически сложных конструкций или для изучения семантики обработчиков ВМ.

Пример 1: Рассмотрим функцию

$$f(x, y, c) := (x \oplus y) + 2 \cdot (x \wedge y) \quad (1)$$

Для того, чтобы узнать основную семантику функции f, были сгенерированы случайные входы.

$$f(2,2,2) = 4,\ f(10,13,10) = 23,\ f(16,3,0) = 19 \quad (2)$$

В результате синтеза программы, очевидно, что функция g имеет вид $g(x, y, c) := x + y$ и обладает эквивалентным поведением входа-выхода. Стоит отметить, что параметр $c$ в данном случае не имеет значения.

Зачастую супероператоры содержат в себе различные семантики, представленные в виде отдельных входов/выходов. Это позволяет злоумышленнику синтезировать каждую семантику независимо, воздействуя на каждый вход по отдельности..

## 2. Пределы синтеза программ

В рамках исследования было произведено оценивание зависимости успешности синтеза программ от семантической сложности. Сгенерировано 10 000 случайных выражений для каждой семантической глубины от 1 до 20 и измерено количество успешно синтезированных. При моделировании искомой функции f, была использована грамматика SYNTIA [11-14] для генерации случайных выражений в зависимости от трех переменных. Основываясь на рекомендациях экспертов, был задан

вектор конфигурации SYNTIA равный (1.5, 50000, 20, 0) и используемый для синтеза каждого выражения.

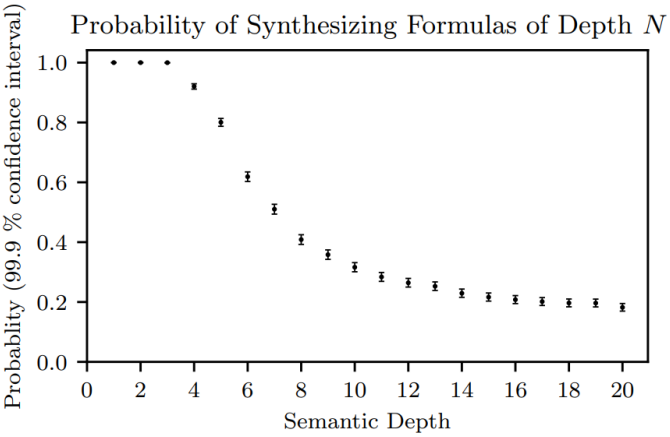


*Рис. 1.* Вероятность синтеза выражений для формул глубины N.

Исходя из Рис.1 можно сделать вывод, что простые выражения могут быть синтезированы довольно легко; при семантической глубине 7, только ~50% может быть синтезировано. При большей семантической глубине синтез выражений становится все менее вероятным.

**Заключение**

Таким образом, в данной статье описан способ повышения эффективности обфускации компьютерного кода при деструктивном воздействии на основе синтеза программ, а также предложен подход, устойчивый к атакам на основе синтеза программ, и определены пределы синтеза в эмпирическом исследовании. деобфускации.

**Список литературы**

1. Дидрих, В. Е. Методика оценки изменений показателей свойств исходных текстов программного обеспечения после обфускации / В. Е. Дидрих [и др.] // Информация и безопасность. – 2014. – Т. 17. – № 2. – С. 288-291.

2. Казарин, И. С. Обзор сетевых атак на информационные системы / И. С. Казарин, Е. М. Михайлова // В сборнике: Интеллектуальный потенциал XXI века: ступени познания. Сборник материалов XXXIX Молодежной международной научно-практической конференции. Под общей редакцией С.С. Чернова. – 2017. – С. 140-148.

3. Диченко, С. А. Реализация двоичных псевдослучайных последовательностей линейными числовыми полиномами / С. А. Диченко, А. К. Вишневский, О. А. Финько // Известия ЮФУ. Технические науки. – 2011. – № 12 (125). – С. 130-140.

4. Диченко, С. А. Контроль и восстановление целостности данных в защищенных информационно-аналитических системах / С. А. Диченко, О. А. Финько // Труды Военно-космической академии имени А.Ф.Можайского. – 2021. – № 676. – С. 36-49.

5. Диченко, С. А. Модель контроля целостности многомерных массивов данных / С. А. Диченко // Проблемы информационной безопасности. Компьютерные системы. – 2021. – № 2 (46). – С. 97-103.

6. Сухов, А. М. Математическая модель процесса функционирования подсистемы реагирования системы обнаружения, предупреждения и ликвидации последствий компьютерных атак / А. М. Сухов [и др.] // Проблемы информационной безопасности. Компьютерные системы. – 2019. – № 2. – С. 56-64.

7. Диченко, С. А. Контроль и восстановление целостности многомерных массивов данных посредством криптокодовых конструкций / С. А. Диченко, О. А. Финько // Программирование. – 2021. – № 6. – С. 3-15.

8. Сухов, А. М. Индуктивный подход для оценки защищенности информационно-телекоммуникационной сети / А. М. Сухов [и др.] // Защита информации. Инсайд. – 2017. – № 6 (78). – С. 68-73.

9. Диченко, С. А. Разработка алгоритма контроля и обеспечения целостности данных при их хранении в центрах обработки данных / С. А. Диченко [и др.] // Сб. науч. статей VIII Междунар. молод. научнопр. конф. с элементами науч. шк. - Омск: Омский ГТУ, 2018. - С. 40-43.

10. Сачков, И. К. Автоматизация противодействия бот-атакам / И. К. Сачков, А. Н. Назаров // T-Comm: Телекоммуникации и транспорт. – 2014. – Т. 8. – № 6. – С. 5-9.

11. Диченко, С. А. Концептуальная модель обеспечения целостности информации в современных системах хранения данных / С. А. Диченко // Информатика: проблемы, методология, технологии. Сборник материалов XIX международной научно-методической конференции. Под ред. Д. Н. Борисова. – Воронеж, 2019. - С. 697-701.

12. Дементьев, В. Е. Понятийный аппарат протокольной защиты информационно-телекоммуникационной сети / В. Е. Дементьев, А. В. Дементьева, Д. А. Маняшин // В сборнике: Актуальные проблемы инфотелекоммуникаций в науке и образовании. Сборник научных статей. – 2016. – С. 70-74.

13. Сухов, А. М. Математическая модель процесса функционирования подсистемы реагирования системы обнаружения, предупреждения и ликвидации последствий компьютерных атак / А. М. Сухов [и др.] // Проблемы информационной безопасности. Компьютерные системы. – 2019. – № 2. – С. 56-64.

14. Варновский, Н. П. Современное состояние исследований в области обфускации программ: определения стойкости обфускации / Н. П. Варновский [и др.] // Труды Института системного программирования РАН. – 2014. – Т. 26. – № 3. – С. 167-198.