Подход к повышению эффективности обфускации кода с использованием смешанной булевой арифметики

М. В. Соколов, email: SaSa_t@mail.ru[1]
В. О. Даренских, email: darenskih.vadim@yandex.ru1
А. А. Дорофеев, email: andrei3267@gmail.com[1]
Я. С. Тодояков, email: todoyakovs@mail.ru[1]

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

Аннотация. *В данной работе рассматривается подход к повышению эффективности обфускации кода добавлением синтаксической сложности с использованием смешанной булевой арифметики, а также представлен общий подход для синтеза разнообразных и формально проверенных смешанных булево-арифметических выражений произвольной сложности.*

Ключевые слова: *смешанная булева арифметика, обфускация, деобфускация, синтаксическая сложность.*

Введение

Обфускация программного обеспечения - важнейшая технология защиты интеллектуальной собственности. Несмотря на свою важность, современные подходы к обфускации уязвимы для множества атак автоматической деобфускации, таких как символическое выполнение, анализ "испорченности" или синтез программы [1-6].

Опора на синтаксическую сложность в современных схемах обфускации и широкий арсенал передовых методов деобфускации спровоцировали дальнейшие исследования по созданию более устойчивых схем, направленных на препятствование этим автоматическим анализам. Были выдвинуты предложения по затруднению анализа "испорченности" [7, 8] и неэффективности символьного выполнения [1, 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 булевы 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.

1. Смешанная булева арифметика

Смешанная булева арифметика (MBA) описывает кодирование выражений синтаксически сложным способом. Цель состоит в том, чтобы скрыть основную семантику в синтаксически сложных конструкциях. Алгебра MBA соединяет арифметические операции (например, сложение) с побитовыми операциями (например, логическими операциями или сдвигом битов). Полученные выражения обычно трудно упростить символически [10-14], поскольку для каждого выражения существует бесконечное число синтаксических представлений. В общем случае, задача сокращения выражений MBA - известных как арифметические кодировки - до эквивалентных, но более простых выражений является трудной.

Пример1: функции

$$f(x, y, c) := x + y \tag{1}$$

$$g(x, y, c) := (x \oplus y) + 2 * (x \vee y) \tag{2}$$

семантически эквивалентны. Обе функции осуществляют одну и ту же основную семантику, но g использует синтаксически более сложное представление, так называемый MBA.

2. Добавление синтаксической сложности

Предполагая объединенную семантику с использованием различных кодировок ключей, злоумышленник все еще может различать кодировку ключа и семантику ядра для данной функции f, поскольку $e_i(k)$ оперирует только ключом, в то время как семантика ядра использует x, y и c. В то же время злоумышленник, знающий k, может использовать символическое выполнение для упрощения f до семантики ядра, связанной с известным k, путем арифметического обнуления операций, не вносящих вклад в результат. Один из подходов, позволяющий предотвратить атаку такого типа, заключается в увеличении синтаксической сложности выражения, путем добавления формул смешанной булевой арифметики (MBA) к кодировкам ключей, а также к основной семантике.

Символьное выполнение этих синтаксически сложных формул не позволяет получить осмысленных выражений. Несмотря на то, что современные механизмы символьного выполнения содержат правила

упрощения для основных арифметических тождеств и законов (например, для коммутативности операторов). В общем случае упрощение такого выражения до его синтаксически наименьшего эквивалентного выражения является трудной задачей.

Пример 2: Для

$$f(x, y, c, k) := e_0(k)(x + y) + e_1(k)(x - y) \quad (3)$$

мы можем заменить $x + y$ на $(x \oplus y) + 2 \cdot (x \wedge y)$, $x - y$ на $x + \neg y + 1$, и заменить мультипликацию $e_1(k)(x - y)$ правилом $(a \wedge b) \cdot (a \vee b) + (a \wedge \neg b) \cdot (\neg a \wedge b)$ для $a \cdot b$, в результате чего получим конечную функцию

$$f(x, y, c, k) := e_0(k)((x \oplus y) + 2 \cdot (x \wedge y)) + (e_1(k) \wedge (x + \neg y + 1)) \cdot (e_1(k) \vee$$

$$\vee (x + \neg y + 1)) + (e_1(k) \wedge \neg (x + \neg y + 1)) \cdot (\neg e_1(k) \wedge (x + \neg y + 1))$$

3. Синтез MBA

Для затруднения символьного выполнения и сопоставления шаблонов, MBA используется для всех компонентов выражения, включая основную семантику и кодировку ключей. Поскольку жестко закодированные правила не обеспечивают должного разнообразия, необходимо заранее вычислять большие классы семантически эквивалентных синтаксических выражений, объединяя их с помощью рекурсивной, случайной перезаписи выражений.

4. Преобразование выражений

В рамках исследования был создан большой набор разнообразных классов эквивалентности, используемых для замены синтаксически простых выражений более сложными. Тривиальный подход, заменяющий выражения MBA из классов эквивалентности, ограничен наибольшей глубиной, найденной в соответствующем классе. Для преодоления этого ограничения, был предложен рекурсивный подход к переписыванию выражений. Данный подход позволяет выстраивать выражения произвольной синтаксической глубины.

Для произвольного основного выражения e, выбирается случайное подвыражение, содержащееся в e, затем производится его проверка на принадлежность к определенному классу эквивалентности. При положительном результате проверки, выбранное подвыражение заменяется произвольным членом соответствующего класса эквивалентности. Предлагаемый подход предполагает рекурсивное повторение вышеописанных операций для случайно определенной верхней границы n.

Пример 4: Пусть основное выражение, требующее увеличения синтаксической сложности, имеет вид:

$e := (x + y) + z$

$n = 2$ – верхняя граница.

На первом этапе, согласно описанному подходу, случайным образом выбрано подвыражение: $x + y$, содержащееся в e. Заменив данное подвыражение членом того же класса эквивалентности, например: $(x \oplus y) + 2 \cdot (x \wedge y)$, выражение для e примет вид: $e := ((x \oplus y) + 2 \cdot (x \wedge y)) + z$.

На втором этапе, выбрав в качестве подвыражения – $(x \oplus y)$ и подобрав ему семантически эквивалентный член $(x \vee y) - (x \wedge y)$ запишем последнее обфусцированное выражение для e: $e := (((x \vee y) - (x \wedge y)) + 2 \cdot (x \wedge y)) + z$.

По результатам проведенных эмперических исследований, стоит отметить, что в силу малой длины случайно выбранных подвыражений, результирующее рекурсивное преобразование выражений носит локальный характер. Это приводит к тому, что операции верхнего уровня (в примере – сложение с z) зачастую игнорируются. С каждой итерацией число операций нижних уровней растет, соответственно, вероятность случайного выбора операции верхнего уровня с каждой итерацией уменьшается [14]. Для устранения данного недостатка, достаточно выбирать в первых итерациях цикла операции верхнего уровня.

Заключение

Рассмотренный подход к повышению эффективности обфускации кода добавлением синтаксической сложности с использованием смешанной булевой арифметики позволяет скрыть основную семантику в синтаксически сложных конструкциях, существенно повышая эффективность обфускации кода, а представленный подход для синтеза разнообразных и формально проверенных смешанных булево-арифметических выражений произвольной сложности дополнительно усложняет процесс 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.