УДК 004.416.6
Павелко О.О.[1], Сташук Д.А.[1], Зайко Т.А.[2]
[1] студ. гр. КНТ-228 НУ «Запорізька політехніка»
[2] канд. техн. наук, доц НУ «Запорізька політехніка»

МАТЕМАТИЧНІ ТА ПРОГРАМНІ МЕТОДИ ОБФУСКАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Захисту програмного коду від декомпіляції у світі приділяється багато уваги, оскільки вихідні тексти програм є комерційною таємницею.

Для захисту вихідних текстів програм використовують обфускацію, яка повністю зберігаючи функціональність програми, ускладнює її аналіз та можливість модифікації [1].

Математичні методи обфускації є одними із сучасних напрямів розвитку криптографії, тому вони тісно пов'язані з їх математичними описами. Такий опис методу обфускації формулюється наступним чином: за обфсукатор розуміється алгоритм, який отримує на вході програму P, і перетворює її в програму O(P) [2].

При цьому задовольняються такі вимоги, як функціональність, ефективність та стійкість [3].

Під функціональністю мається на увазі те, що програми P і O(P) обчислюють одну й ту саму функцію, тобто є еквівалентними. Витрати на перехід від P до O(P) незначні, тобто збільшення обсягу програми і часу її виконання невеликі, що робить її ефективною. Якщо отримана програма O(P) важка для розуміння, то вона є стійкою.

Недоліками методів математичної обфускації є те, що алгоритм не дає сторонньому з'ясувати, що робить код, але й не дає розробнику налагоджувати його, крім того жоден з існуючих алгоритмів не гарантує абсолютної безпеки.

Програмні методи вимірюються декількома параметрами, що характеризують ефективність їх застосування, а саме: дані (роблять елементи

коду схожими на те, чим вони не є), потік коду (виставляють логіку програми абсурдною або навіть недетермінованою), структура формату (застосовують різне форматування даних, перейменування ідентифікаторів, видалення коментарів) [4].

Перейменування ідентифікаторів – заміна імен у програмі на послідовності символів, що не несуть смислового навантаження [4]. Приклад ідентифікатору : oo01iooo11o001oioii1ol00oioillol00o101ol. Цей метод має такі переваги: втрата структурності, зміна типу змінних, заплутування логіки. Незважаючи на переваги, дана технологія не позбавлена таких недоліків: структурність легко відновлюється після деобфускації, змінені імена змінних можна замінити на читані [4].

Додавання "мертвого" коду. «Мертвий» код виконується під час роботи програми, але впливає на її результат. У цьому методі визначені такі переваги, як: повна втрата структурності та гнучкості без можливості відновлення. Однак після обфускування код стає залежним від платформи та деякі деобфускатори здатні привести код до початкового стану [4].

Додавання надлишкового коду. Цей метод є простим у реалізації, але забезпечує надійний захист від зловмисника. Ідея полягає в тому, що вихідний код програми засмічується не важливою інформацією, непотрібними викликами або методами та вставляються помилкові пояснення подальшого коду. Цьому методу притаманні такі переваги: втрата структурності коду, заплутування логіки, збільшення обсягу [4].

Також було визначено недоліки даної обфускації: вставка програмного сміття може збільшити час виконання програми та її легко деобфусцувати (якщо використовується лише вставка коментарів);

Отже, для забезпечення надійного захисту вихідного тексту програми, необхідно комбінувати розглянуті методи обфускації, такі як: перейменування ідентифікаторів, додавання надлишкового та "мертвого" коду. Важливо, щоб результат обфускації не впливав на час виконання програми та був стійким до деобфускації.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Collberg C. Manufacturing cheap, resilient, i stealthy opaque constructs / C. Collberg, C. Thomborson, D. Low // Proc. Symp. Principles of Programming Languages (POPL'98), Jan. 1998.

2. Collberg C. Taxonomy of Obfuscating Transformations [Electronic resource] / C. Collberg, C. Thomborson, D. Low. – Access mode: http://www.cs D.arizona.edu/collberg/Research/Publications/CollbergThomborso nLow97a/index.html, вільний.

3. Чернов А.В. Исследование и разработка этодологии аскирования програм: дис. канд. физ.-мат. наук. Моск. держ. ун-т им. М. В. Ломоносова. – М., 2003. – 133 с.

4. Чернов А.В. Анализ запутывающих преобразований программ / А.В Чернов // Библиотека аналитической информации: дис. канд. физ.-мат. наук. Моск. держ. ун-т им. М.В. Ломоносова. – М., 2005.