УДК 004.056.5
Зайко Т.А.[1], Козлов В.В.[2]
[1] канд. техн. наук, доц. НУ «Запорізька Політехніка»
[2] студ. гр. КНТ-137 НУ «Запорізька Політехніка»

ОБФУСКАЦІЯ ЯК ЗАСІБ ЗАХИСТУ ВИХІДНОГО КОДУ ПРОГРАМИ

Сьогодні інформацію розглядають як один з основних ресурсів розвитку суспільства, а інформаційні системи і технології як засіб підвищення продуктивності та ефективності роботи людини. Тому інформація та інформаційні технології є найціннішим і найдорожчим ресурсом, який потребує захисту.

Найбільш серйозна проблема при захисті програмного забезпечення – це протидія різним засобам статичного і динамічного аналізу коду, які використовуються для реверсінга.

Реверсінг, реверс-інжинірінг або зворотна розробка – це дослідження деякого пристрою або програми, а також документації на них з метою зрозуміти принцип і алгоритм його роботи. Під час зворотної розробкидослідник вивчає програму з закритим вихідним кодом, аналізуєїї структуру та принцип роботиі, можливо, вносить якісь зміни [1].

Мета розробника захисту полягає в тому, щоб якомога сильніше ускладнити і затягнути роботу по реверсінгу.

Для цього використовують процес засмічення коду або обфускації. Обфускація – це приведення виконуваного коду або вихідного тексту програми до виду, який зберігає її функціональність, але ускладнює розуміння і аналіз алгоритмів його роботи.

Обфускацію використовують для того, щоб ускладнити

процесдекомпіляції і вивчення програми [2], запобігти обхід систем перевірки ліцензій та зменшити розмір працюючого коду, а отже і прискорити його роботу.

Існує три основних технології, які використовуються в залежності від мови програмування і способу розповсюдження програми.

Перша технологія – це технологія, що використовується на рівні вихідних текстів програми. На JavaScript, VBScript та подібних скрипт-мовах вихідний текст завжди доступний користувачу. В цьому випадку форматуванням тексту і заміною імен змінних можна зробити вихідний текст програми менш читабельним. Найпопулярніший варіант на даний момент –це перейменування в недруковані символи або використання коротких незрозумілих ідентифікаторів. Також одним з підходів є створення великої кількості overload методів з одним ім'ям, які мали до обфускації різні імена, і ніяк не були пов'язані.

Друга технологія – це технологія, яка використовується на рівні проміжного або байт-коду. Java і мови платформи .NET компілюють вихідний код в байт-код, що містить достатньо інформації для того, щоб відновити вихідний код. Для таких мов використовується принцип зміни потоків управління. На цьому етапі змінюються інструкції та їх порядок в коді, вводяться умовні оператори, а методи розбивають на блоки.

Третя технологія – це технологія, яка використовується на рівні машинного коду. Мови програмування, такі як C++ та Pascal, компілюють вихідний код в машинний. В цьому випадку обфускація застосовується в критичних до безпеки, але не критичних до швидкості місцях, шляхом вставки недіючих конструкцій.

Представлені технології прийнято використовувати як в сукупності, так і окремо [3]. Все залежить від бажаного ступеня захисту програми і мети його розробки. Найчастіше всі дії зводяться до того, що в програму вносяться спеціальні функції, які виконують складні дії, звертаються до накопичувачів даних, але нічого не змінюють.

Обфускація докучає хакерам, перешкоджаючи реконструкції алгоритмів і швидкому злому захистів, проте і створюєпроблеми в антивірусній індустрії. Щоб зламати програму, аналізувати її алгоритм необов'язково. А ось виявити шкідливий код (він же malware) без цього вже не вдасться.Також до основних недоліків використання обфускації можна віднести ще втрату гнучкості коду і труднощі його налагодження.

Підводячи підсумки можна сказати, що обфускація допомагає зробити розподілену систему безпечнішою, але не варто обмежуватися тільки нею. Обфускація – це безпека через неясність. Жоден з існуючих алгоритмів обфускації не гарантує складності декомпіляції і не забезпечує безпеки на рівні сучасних криптографічних схем. Тому її слід проводити в комплексі з

іншими методами шифрування і захисту даних.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Панов А.С. Реверсинг и защита программ от взлома / А.С. Панов. – Спб.: БХВ-Петербург, 2006. – 256 с.

2. Brunton F. Obfuscation: A User's Guide for Privacy and Protest / F. Brunton, H. Nissenbaum. – Massachusetts.: Press Cambridge, 2015. – 107 p.

3. Оголюк А.А. Защита приложений от модификации. Дополнительные материалы / А.А. Оголюк. – Спб.: СпбГУ ИТМО, 2014. – 122 с.