УДК 004.056.53
В. Радчук, Н. Шингера, канд. техн. наук
Тернопільський національний технічний університет імені Івана Пулюя, Україна

ОГЛЯД МЕТОДУ ОБФУСКАЦІЇ КОДУ, ЯК ПРОТИДІЯ ДИЗАСЕМБЛЮВАННЮ

V. Radchuk, N. Shynhera

REVIEW OF THE OBFUSCATION CODE METHODS AS A DISASSEMBLY COUNTERACTION

Зростання інформатизація суспільства формує гостру необхідність у захисті різного програмного забезпечення (ПЗ).

Розробники ПЗ все частіше погоджуються з думкою, що для успішного просування їх продукту і отримання від цього прибутку необхідно шукати і впроваджувати найбільш ефективні методи захисту програмного продукту. Якісний захист від неліцензійного використання програмного забезпечення не дозволить комусь використовувати його, не заплативши відповідну винагороду розробнику.

Хоча ефективний захист ПЗ також коштує недешево, але при його впровадженні, слід враховувати, що затрачені кошти в надлишку повертаються за рахунок продажу ліцензій.

На даний момент існує кілька методів захисту ПЗ, а саме: захист, що базується на виконанні певних програм на сервері, на встановленні справжності коду, на шифруванні коду і т.п. Серед них, безсумнівно, перспективним і порівняно новим методом є обфускація.

Як правило, для обходу захисних механізмів ПЗ зловмиснику необхідно спочатку вивчити принцип його роботи, тобто розібратися в програмному коді модулів захисту і зрозуміти, як вони взаємодіють з захищеною програмою. Якщо ускладнити вирішення даного завдання, то це дозволить зменшити ймовірність злому.

Обфускація – це приведення сирцевого тексту або виконуваного програмного коду до вигляду, який зберігає його функціональність, але ускладнює аналіз, розуміння алгоритму роботи і модифікації при декомпіляції.

«Заплутування» коду може здійснюватися на рівні алгоритму, сирцевого тексту або асемблерного тексту. Для створення «заплутаного» асемблерного тексту можуть використовуватися спеціалізовані компілятори, які використовують неочевидні або недокументовані можливості середовища виконання програми. Існують також спеціальні програми, що виробляють обфускацію, які називаються обфускаторами.

Сенс процесу, назва якого обфускація, в тому, щоб ускладнити програмний код і тим самим приховати логічні зв'язки, які могли б показати зловмиснику принцип роботи захисних механізмів ПЗ, і, отже, сприяти злому. [1]

Найчастіше обфускацію використовують у поєднанні з іншими методами захисту, що дає можливість значно підвищити рівень захисту ПЗ.

Ще одним аргументом на користь застосування у захисті ПЗ обфускації є те, що вартість її використання порівняно нижча, ніж у інших методів захисту. Це пояснює стрімкий розвиток даного методу останнім часом.

Таким чином є сенс детальніше розглянути даний метод захисту ПЗ.

Процес обфускації повинен задовольняти наступні вимоги:

- код програми після обфускації має істотно відрізнятися від вихідного, але функціонально зобов'язаний залишатися тотожним;

- вивчення програмного коду програми після обфускації повинно бути більш складним і трудомістким, ніж до неї;
- створення програми, яка змогла б відновити вигляд ПЗ до обфускації, повинно бути неможливим. [2]

Оскільки програмний код може бути представлений як у вигляді команд вихідною мовою програмування, так і у вигляді двійкового коду, отриманого в результаті компіляції, то також існує два відповідних рівні обфускації:

- нижчий – на рівні двійкового коду;
- вищий – на рівні вихідної мови програмування.

На вищому рівні проводити обфускацію істотно легше, бо відсутня залежність від архітектури процесорів. Обфускація на низькому рівні вивчена поки ще досить мало, оскільки це значно складніше.

Залежно від способу модифікації коду розрізняють такі види обфускації:

- лексична обфускація – найпростіший вид обфускації, полягає у форматуванні коду програми, зміні його структури, що веде до неможливості його прочитання;
- обфускація даних – заснована на трансформації структур даних;
- обфускація управління – заснована на заплутуванні потоку виконання програми;
- превентивна обфускація – заснована на недоліках програмних комплексів по деобфускації.

Деобфускація – процес зворотний обфускації. Він спрямований на відновлення початкового коду, який піддавали обфускації. Головні методи деобфускації:

- знаходження складних конструкцій та їх аналіз;
- співставлення отриманих зразків;
- виявлення невикористовуваних фрагментів коду;
- аналіз потоку даних;
- статичний аналіз;
- динамічний аналіз.

Статичний аналіз являє собою технології проведення аналізу програм без необхідності їх запуску. При цьому використовуються спеціальні програмні засоби, які дозволяють дізнатися необхідну інформацію про досліджувану програму. Дані, отримані на підставі статичного аналізу, вважаються неточними, оскільки базуються на прогнозі можливої поведінки програми.

Динамічний аналіз проводиться шляхом запуску програми і дослідження роботи як повністю, так і певної її частини. Дані, отримані за допомогою динамічного аналізу, вважаються більш точними, але мають низьку повторюваність. [3]

Таким чином, проведення обфускації повинно враховувати методи і засоби, які можуть бути використані зловмисником для її нейтралізації. Цього можна досягти шляхом збільшення обсягу обчислень і ускладненням алгоритму виконуваних функціональних перетворень.

Література

1. Хорошко В.А. Методы и средства защиты информации. В. А. Хорошко, А. А. Чекатков. Москва: Юниор 2003 – 504с;

2. Панов А.С. Реверсинг и защита программ от взлома / Панов А.С. – СП.: БХВ – Петербург, 2006. – 256 с.

3. Касперски К. Искусство дизассемблирования / Касперски К., Рокко Е. – СПб.: БХВ-Перербург, 2008. – 896 с.

4. Net асемблер і дизасемблер [Електронний ресурс] Режим доступу: URL: http://php-functions.ho.ua/ukr/dotnet/rphp8.html – Назва з екрану.