УДК 004.4
О.П. Ясній, докт. техн. наук, проф., В.І. Карплюк
Тернопільський національний технічний університет імені Івана Пулюя, Україна

МЕТОДИ ОБФУСКАЦІЇ ПРОГРАМНОГО КОДУ В КОМП'ЮТЕРНИХ СИСТЕМАХ

O.P. Yasniy, Dr. Sc., Prof., V.I. Karplyuk

OBFUSCATION METHODS OF SOFTWARE CODE PROTECTION IN COMPUTER SYSTEMS

Програміст витрачає багато зусиль, налагоджуючи код, однак то тільки мала частина роботи. Після того, як проект створено, налагоджено та розгорнуто, потрібно запобігти спробам сторонньої особи скопіювати вихідний код і скористатися ним.

Сьогодні зловмисники мають велику кількість програм для злому – від простих шкідливих програм до складних інструментів реверсної інженерії. Дизасемблери, декомпілятори та інші інструменти дозволяють хакерам отримувати доступ та аналізувати вихідний код програми. Очевидно, що за допомогою такої інформації хакери можуть зловживати програмним забезпеченням різними способами: витягувати конфіденційну інформацію, додавати шкідливий код, наносити різного роду збитки клієнтам або ж цілим проектам, клонувати програмні продукти.

У результаті обфускації вихідний код навмисно ускладнюють для того, щоб запобігти реверсній інженерії. При цьому функціональність обфускованого коду еквівалентна вихідному.

Мета обфускації – заплутати програмний код і усунути більшість логічних зв'язків у ньому, тобто перетворити код так, щоб його було важко вивчити і модифікувати стороннім особам. Обфускацію здійснюють наступними методами:

- лексична обфускація (перейменування імен змінних та методів);
- обфускація даних (маскування структур даних під такі, якими вони не є);
- обфускація графа потоку керування (заміна виконуваної логіки недетермінованою та додавання випадкового зайвого заплутаного коду).

Проаналізувавши існуючі методи обфускації програмного коду, написано удосконалений обфускатор для мови JavaScript, котру застосовують у веб-проектах.

Розроблено три унікальних режими роботи обфускатора, кожен із яких обфускує код у різних частках того чи іншого етапу:

- зміна розміру графу потоку керування функції (клонування базових блоків, зміна структури циклів);
- руйнування вихідної структури графу потоку керування функції (додавання до логіки зайвих зв’язків, розбиття блоків на менші, створення нових блоків для наступного етапу);
- генерація додаткового коду (заповнення новоутворених порожніх блоків інструкціями, що ніяк не впливають на виконання основної логіки, додавання мертвого коду в інші блоки програми);
- поєднання мертвого коду із основною програмою за допомогою зайвих логічних зв’язків та математичних тотожностей і нерівностей.

Обфускувавши важливі частини вихідного коду запропонованим інструментом, отриманий код стає значно складнішим для реверсної інженерії. Зловмиснику знадобиться набагато більше часу, щоб зрозуміти, що саме замасковано у коді.