УДК 303.7
Медведєв С.Р.[1], Зайко Т.А.[2]
[1]студ. гр. КНТ-127 НУ «Запорізька Політехніка»
[2]канд. техн. наук, доц. НУ «Запорізька Політехніка»

УСКЛАДНЕННЯ ЛОГІКИ ПРОГРАМИ ЯК РЕАЛІЗАЦІЯ ЗАХИСТУ ВІД ДИЗАСЕМБЛЮВАННЯ

На сьогоднішній день, взлом і піратство програмного забезпечення є проблемою для багатьох початківців і навіть для досвідчених розробників і програмістів. Зловмисники в різних технологічних сферах діяльності всіляко намагаються викрасти інформацію як у "сирих" або нових програмних продуктів, так і у тих продуктів, які вже давно на ринку. Як приклад, можу навести багаточисленні взломи відеоігор від різних хакерських угруповань. Взлом і піратство становлять загрозу для розробників як у фінансовому плані, так і в плані довіри. Одним з видів взлому є процес дизасемблювання.

Більшість програм поширюються в вигляді виконуваного файлу. Для взлому програми з метою піратства необхідно перш за все перевести машинний код в асемблерне уявлення. Такий процес носить назву дизасемблювання. Якщо вдасться зробити нерозв'язною задачу зловмисника вже на етапі дослідження програми, шанси на збереження інтелектуальної власності стануть стовідсотковими.

Процес дизасемблювання дуже ризиковий процес з більшістю втрат, тому що він однонаправлений. При повторному асемблюванні відновленого тексту немає гарантії отримати той же самий код, тобто отримана програма, швидше за все, відмовиться працювати. Будь-яка спроба модифікації дизасемблювання тексту розвалить програму остаточно. Справа в тому, що асемблер замінює всі мітки на реальні зміщення, інакше кажучи - на константи. При внесенні змін до програми необхідно скорегувати всі посилання на мітки.

Першим і універсальним методом протидії дизасемблювання програми є шифрування. Вочевидь, що дизасемблювання зашифрованого коду марно. Рекомендується використовувати шифрування з відкритим ключем (алгоритм

RSA, шифр Єль-Гамаля і ін.). В цьому випадку можлива вдала спроба розшифрувати код і зрозуміти логіку роботи захисного механізму не дозволить вносити зміни в захищений код, так як для повноцінної подальшої роботи програми ці зміни необхідно впровадити в код в зашифрованому вигляді. А зловмиснику доступний лише ключ для розшифровки. Можлива атака в даному випадку - знаходження «секретного» ключа за допомогою трудомістких математичних обчислень в залежності від використовуваного алгоритму шифрування. Для захисту програм від дизасемблювання, не рекомендується використання симетричних криптографічних алгоритмів.

Другі методи, які непогано зарекомендували себе на практиці, використовують разом з шифруванням архівування програмного коду. До переваг даних методів відносять і зменшення розміру виконуваного файлу. Найбільш відомими архіваторами на сьогоднішній день є WinRAR та 7-Zip. Однак слід враховувати, що алгоритми роботи цих архіваторів відомі багатьом зловмисникам.

Треті методи, які засновані на динамічній зміні коду програми в процесі виконання, теж широко поширені на практиці,. Суть цих методів зводиться до отримання істинних здійсненних команд на етапі виконання програми шляхом деякого перетворення початкових кодів. Найчастіше цей спосіб захисту називають самостійно-згенерованим кодом.

Четвертий метод, більш простий, ніж перераховані вище, адже його реалізацію можна почати ще на етапі розробки коду програми. Називається він "обманом дизасемблера". Метод передбачає такий стиль програмування, який викликає порушення правильної роботи стандартного дизасемблера за рахунок нестандартних прийомів використання окремих команд або ж порушення загальноприйнятих угод.

На сьогоднішній день, існує чимало програм дизасемблерів. Наприклад: PE-Explorer, FlexHex, Radare2, PE-bear, Hiew, IDA тощо. Але найкращими представниками даної функціональності є IDA Pro Disassembler та вільний багатоплатформений фреймворк Radare2.

IDA є комерційним продуктом, ціни якого коливаються в межах від 400 до 4000 тисяч доларів. Існує також безкоштовна версія, але в ній вирізаний майже весь контент та немає підтримки модулів.

Дизасемблер IDA Pro має консольну і графічну версії. Підтримує велику кількість форматів виконуваних файлів. Однією з відмінних рис IDA Pro є можливість дизасемблювання байт-коду віртуальних машин Java і .NET. Також підтримує макроси, плагіни і скрипти, а останні версії містять інтегрований відладчик.

Фреймворк Radare2 є безкоштовним кросплатформним продуктом, який спочатку розроблявся одною людиною, а після - цілою командою.

Radare2 доступний у вигляді бібліотек і утиліт, в склад яких входять: власний асемблер/дизасемблер, утиліта для роботи з різними виконуваними файлами, утиліта для отримання хеш-значень даних, декомпілятор тощо. Також, на його архітектурі був оснований інтерактивний дизасемблер Cutter.

Підсумовуючи наведене, можна сказати, що за захист від дизасемблювання відповідають методи ускладнення логіки програм, а вся відповідальність при їх використанні чи невикористанні, при розробці програм, відповідально лежить на плечах розробників програмного забезпечення.