УДК 004.056.53

РАЗРАБОТКА МЕТОДОВ И АЛГОРИТМОВ ЗАПУТЫВАЮЩЕГО ПРЕОБРАЗОВАНИЯ ДЛЯ ИСПОЛНЯЕМЫХ ФАЙЛОВ

Магистрант Мазько К.А.
Белорусский государственный университет информатики и радиоэлектроники

С развитием индустрии создания и распространения программного обеспечения (ПО) обостряются проблемы его несанкционированного использования, поэтому одним из актуальных вопросов для производителей ПО является его защита от незаконного копирования; присвоения алгоритмов или их частей для использования в конкурирующих продуктах; несанкционированной модификации.

Объектом исследования является программное обеспечение. Предметом исследования являются методы технической защиты программного обеспечения.

Целью работы является разработка эффективных методов защиты ПО от обратного проектирования и несанкционированного исследования.

Для достижения поставленной цели решены следующие задачи:

– проведен анализ существующих методов защиты ПО от угроз несанкционированного исследования и обратного проектирования;

– разработаны методы запутывающего преобразования, затрудняющие статический анализ исполняемых файлов платформы Win32: перемешивание машинных команд и настройка новых переходов между ними («спагетти-код»); добавление конструкций, затрудняющих дизассемблирование; удаление важных для исполнения программы данных и их расчет в процессе выполнения с помощью математических алгоритмов; локальное шифрование отдельных блоков машинного кода;

– разработаны методы запутывающего преобразования, затрудняющие динамический анализ (отладку) исполняемых файлов платформы Win32: защита от отладчиков с изменением потока выполнения программы; добавление дополнительных блоков машинных команд, которые за счет сложности собственного анализа увеличивают сложность всего запутанного кода.

Результатом работы стало программное средство, реализующее разработанные алгоритмы для исполняемых файлов операционных систем семейства Win32 (процессорная архитектура IA-32). В связи с сильным влиянием запутывающего преобразования на объем и скорость исполняемого кода, степень запутанности может гибко настраиваться для отдельных участков кода. Реализация алгоритмов дала возможность оценить их влияние на защищенность реального ПО и сделать положительные выводы о возможности применения. В дальнейшем планируется реализация алгоритмов для других аппаратных платформ.