Титульный лист

# Введение

В наши дни информационные технологии являются основной движущей силой прогресса и захватывают все большие области современной жизни. Информационные технологии – это совокупность программно-технических средств вычислительной техники и способов их применения для обработки информации в различных областях. В связи с таким широким их распространением, серьезно встает вопрос защиты обрабатываемых и хранимых данных. Ведь если раньше, вы были уверены в том, что никто не получит доступ к вашим документам, лежащим в сейфе, без ключа или пароля. Теперь же большая часть информации хранится в электронном виде и необходимы уже другие способы защиты.

Этот принцип действует и для разрабатываемых программ. Если раньше, программы были по большей части низкоуровневыми и хранились в виде машинных кодов конкретной архитектуры, то с течением времени появляется все больше различных высокоуровневых инструментов для облегчения разработки, развиваются и усложняются сами языки программирования.

Чтобы снять с разработчика обязанности прямой работы с памятью, контроля типов, а также сделать программу более гибкой для различных платформ и языков, приложения выполняются специальной исполняющей средой (виртуальной машиной). Примеры таких виртуальных машин: JVM (виртуальная машина Java), CLR (общеязыковая исполняющая среда), Dalvik Virtual Machine (часть мобильной платформы Android). Эти средства значительно упрощают разработку, но они разработаны так, что требуют трансляции приложения в промежуточный байт-код (промежуточный язык, язык высокоуровневого ассемблера), который затем в момент выполнения компилируется в машинный ход JIT-компиляторами (компиляция «на лету»). Вследствие чего, приложения хранятся на диске в виде инструкций промежуточного языка, который легко декомпилируется обратно в исходные коды с помощью различных инструментов, таких как .NET Reflector, FernFlower (декомпилятор Java) и другие.

Соответственно существуют простые способы обхода лицензирования, получения исходных кодов и модификации под свои нужды незащищенных приложений написанных на языках .Net (и других подобных языков программирования, таких как Java). При этом для них не существует абсолютной защиты, как и не существует такой защиты для низкоуровневых приложений (например Win-32, написанных на С++), отсюда исходит основной принцип защиты программных продуктов:

*Защита приложения лишь увеличивает время его взлома и в идеале, защита должна окупить себя, т.е. время, затраченное на защиту приложения должно быть сопоставимо со временем, затраченным на взлом этой защиты.*

Существуют несколько основных технологий или методик защиты .Net приложений. Их можно разделить на две части: те, которые работают с кодом программы и те, которые работают с памятью.

Для сокрытия кода используются различные обфускаторы и упаковщики. Обфускаторы используют для защиты запутывание кода (control flow), переименование методов и классов, шифрование ресурсов, добавление невалидных конструкций и др. Упаковщики модифицируют сборку, и на выходе получается winapi-приложение, содержащие в себе код сборки и модуль распаковки. Упаковщики и обфускаторы могут применяться одновременно.

В дополнение к защите кода может применяться защита данных, хранящихся в памяти во время выполнения программы. Здесь возможно множества решений, которые зависят от конкретного приложения и его архитектуры.

Так же существует еще один метод защиты – модификация кода во время выполнения приложения. С помощью перехвата вызовов функций (hooking) можно изменять поведение программы, что дает возможность более сложной защиты приложений.

Данная работа направлена на рассмотрение существующих решений, выявления преимуществ и недостатков и разработку нового решения, основанного одновременно на нескольких методах и технологиях, как защиты, так и упрощения ее внедрения в готовый проект.