УДК 004.4.056.5

Никольская К. Ю., Хлестов А. Д.

ОБФУСКАЦИЯ И МЕТОДЫ ЗАЩИТЫ ПРОГРАММНЫХ ПРОДУКТОВ

Данная статья нацелена на рассмотрение основных технических методов защиты программных продуктов. Процесс обфускации на сегодняшний день один из самых популярных и часто используемых методов защиты, поэтому он выделен в отдельный блок для более детального изучения.

Ключевые слова: *обфускация, защита программных продуктов, программных продукт.*

Hlestov A. D., Nikolskaya K. U.

OBFUSCATION AND METHODS OF PROTECTION SOFTWARE

This article focuses on the technical review of the main methods of protection software. The process of obfuscation to date one of the most popular and commonly used methods of protection, so it is in a separate block for a more detailed study.

Keywords: *obfuscation, protection of software products, software products.*

Технологии программирования прогрессируют очень быстро. Сейчас исходный код незащищённой программы можно вскрыть без особых усилий, имея некоторую подготовку. Приведу пример технологии .NET, созданной компанией Microsoft. Платформа .NET решает многие проблемы, которые в прошлом омрачали процесс разработки Windows-приложений. Теперь существует одна для всех поддерживаемых платформой языков программирования парадигма разработки приложений. Платформа .NET позволяет разрабатывать мощные, независимые от языка программирования, настольные приложения и масштабируемые (расширяемые) Web-службы, построенные на базе новой мощной полнофункциональной библиотеки классов. Однако защита .NET программ представляет значительную трудность из-за их большой открытости. Приложения для .NET не представляют сложностей для декомпиляции. Однако существует множество методов защиты программных продуктов, которые мы рассмотрим в данной статье.

Основные способы защиты. Не секрет, что любой программный продукт представляет собой некую интеллектуальную собственность. Существует два основных спосо-

ба защиты интеллектуальной собственности (программных продуктов в частности):

1) Юридический. Заключается в создании локальных правовых актов в соответствии с законодательством, которые будут регламентировать использование и защиту интеллектуальной собственности от нелегального использования.

2) Технический. Мы рассматриваем защиту программных продуктов (ПП), поэтому в данном случае технический способ реализуется путём включения в ПП определённых методов защиты, которые будут предотвращать нелегальное использование продукта.

Методы защиты программных продуктов. Опишем наиболее распространённые методы защиты, известные на данный момент.

Выполнение на сервере. Как можно заметить из названия, программа действует по принципу «клиент – сервер». Основной код программы находится на сервере, и её код выполняется на серверной стороне, однако аргументы для работы передаются программой-клиентом. Является одним из самых эффективных методов, но:

1) требует взаимодействия клиента и сервера, что подразумевает наличие сети;

2) скорость работы программы зависит от пропускной способности сети клиента и сервера.

Исходя из этого, данный метод лучше всего использовать для простых программ (сценариев).

Водяной знак (software watermark). Водяной знак – некоторый скрытый код в программном продукте, содержащий информацию о разработчике программы. Водяной знак должен соответствовать следующим требованиям:

• водяной знак должен быть хорошо скрыт в программе и разработчик должен иметь возможность извлечь скрытый код без каких-либо повреждений;

• водяной знак не должен влиять на работу программы;

• водяной знак должен нести определённую информацию о разработчике, которая позволит доказать, что её присутствие в программе неслучайное и является результатом преднамеренных действий.

Для лучшей защиты рекомендуется вставлять в программный продукт несколько водяных знаков. Недостатком такого метода является то, что злоумышленник может обнаружить в программном коде эту информацию и подвергнуть изменению.

Установка подлинности кода (tamper-proofing). Данный метод реализуется путём внедрения в программный продукт процедуры проверки целостности программы. При попытке изменения программы данная процедура делает её неработоспособной. Процедура не должна быть слишком простой, так как для взлома программы достаточно будет определить место, откуда эта процедура вызывается.

Шифрование программного кода. Данный метод защиты предусматривает зашифровывание кода программы, после чего она в зашифрованном виде поставляется конечным пользователям. Когда пользователь запускает такую программу, вначале будет запущена процедура расшифровки программы, которой потребуется ключ, с помощью которого будет расшифрована запускаемая программа.

Ключ представляет из себя последовательность символов, генерируемых в результате некоторых математических операций. К примеру, ключ может быть привязан к характеристикам компьютера пользователя. Однако это может усложнить работу с программой в случае запуска на другом компьютере.

В последнее время становится актуально использование электронных ключей. Электронный ключ представляет из себя небольшое устройство, подключаемое к одному из портов компьютера (COM, USB и т. д.). Такой метод лучше всего подходит для защиты дорогостоящих ПП. Также он не ограничивает работу на определённом компьютере, как в случае, описанном выше.

Обфускация. Для обхода защиты программы в большинстве случаев требуется изучение программного кода. Этот процесс называется «реверсивная (обратная) инженерия». Что же из себя представляет обфускация? Обфускация (от лат. obfuscare — затенять, затемнять; и англ. obfuscate — делать неочевидным, запутанным, сбивать с толку) — приведение исходного текста программы к виду, сохраняющему его функциональность, но затрудняющему его анализ, понимание алгоритмов работы. То есть, обфускация усложняет процесс реверсивной инженерии. Данный метод не является идеальным и его рекомендуется использовать вместе с другими методами защиты. Это очень сильно повысит общий уровень защищённости программного продукта.

Некоторый процесс трансформации программного кода будет считаться процессом обфускации, если он удовлетворяет следующим требованиям:

• код программы будет существенно отличаться от оригинала, но при этом функционирование программы не изменится;

• изучение трансформированного кода (реверсивная инженерия) будет более сложным и трудоёмким, чем изучение исходного кода;

• при каждом процессе трансформации результирующий код будет различным;

• создание программы обратной трансформации кода будет неэффективно.

Принято выделять следующие уровни процесса обфускации:

• низший уровень, когда процесс обфускации осуществляется над ассемблерным кодом программы или непосредственно над двоичным файлом программы, хранящим машинный код;

• высший уровень, когда процесс обфускации осуществляется над исходным кодом программы, написанном на языке высокого уровня.

Наиболее популярным на данный момент является высший уровень процесса обфускации. На низком уровне должны быть учтены особенности работы процессоров, так как программа после обфускации будет корректно работать на одной архитектуре и некорректно на другой.

Оценка процесса обфускации:

• устойчивость – показывает уровень сложности изучения кода программы, прошедшей процесс обфускации;

• эластичность – показывает защищённость программного кода от применения деобфускаторов (программ для обратного преобразования кода);

• стоимость преобразования – показывает, насколько больше системных ресурсов требует преобразованная программа в сравнении с оригиналом.

Виды обфускации. Существуют различные способы преобразования программ, следовательно, данный процесс подразделяется по видам (способам) такого преобразования. Ниже рассмотрим некоторые из них.

Лексическая обфускация. Наиболее простая, заключается в изменении исходного кода программы для приведения его к нечитабельному виду. Включает в себя: удаление комментариев или изменение их на дезинформирующие; удаление отступов и пробелов; замена имён идентификаторов (имён переменных, функций, процедур и т. д.) на длинные наборы символов, сложных для визуального восприятия; изменение расположения блоков программы.

Обфускация данных. Данный тип обфускации связан с изменением структур данных. Является более сложной, чем лексическая, однако наиболее используемой. Этот вид обфускации делится на 3 группы:

1) Обфускация хранения. Заключается в трансформации хранилищ данных, а также самих типов данных (например, создание и использование необычных типов данных, изменение представления существующих и т. д.);

2) Обфускация соединения. Один из важных этапов в процессе реверсивной инженерии программ, основан на изучении структур данных. Поэтому важно постараться в процессе обфускации усложнить представление используемых программой структур данных. Например, при использовании обфускации соединения это достигается благодаря соединению независимых данных или разделению зависимых;

3) Обфускация переупорядочивания. Заключается в изменении последовательности объявления переменных, внутреннего расположения хранилищ данных, а также переупорядочивании методов, массивов, определенных полей в структурах и т. д.

Превентивная обфускация. Данный вид обфускации предназначен для предотвращения успешного применения деобфускаторов к коду программного продукта. Нацелен на использование недостатков часто используемых программных средств деобфускации.

Никольская Ксения Юрьевна, преподаватель кафедры «Безопасность информационных систем» ЮУрГУ, г. Челябинск. E-mail: bambucha13@mail.ru

Хлестов А. Д., студент ЮУрГУ, г. Челябинск

Nikolskaya Ksenia, Lecturer, Department of Information Systems Security SUSU, Chelyabinsk. E-mail: bambucha13@mail.ru

Hlestov A. D., student SUSU, Chelyabinsk