

# Защита программ от исследования

Константин Пасечников, руководитель департамента технических проектов компании "StarForce Technologies"



**В ОСНОВЕ** каждого программного продукта лежит интеллектуальная собственность его разработчиков. Ведь на создание программы ушло много часов работы программистов, тестировщиков и службы маркетинга. Хочется, чтобы при коммерческом релизе уникальный продукт не был изучен, скопирован и модифицирован конкурентами, а для

этого программы необходимо защищать от статического и динамического анализа. Под статическим анализом понимается исследование исполняемого кода приложения без запуска программы (дизассемблирование). Под динамическим – изучение алгоритмов работы программы при запуске приложения (инструкции проходятся пошагово). Именно на предотвращение этих видов анализа и направлена защита программ от исследования.

Для того чтобы модернизировать исследуемую программу или сделать на основе про-

граммы конкурентов новый продукт, нужно досконально изучить исполняемый код приложения, то есть провести реверс-инжиниринг. Чаще всего для этого используются специальные инструменты, позволяющие разобрать программу "до винтиков", – дизассемблеры и отладчики. Первые используются для того, чтобы преобразовать исходный машинный код программы в удобочитаемый код на низком языке программирования – ассемблере. Отладчики же информируют о каждом действии в недрах компьютера и позволяют отследить каждый шаг программы таким образом, чтобы можно было понять суть всех операций. В основе реверс-инжиниринга лежит принцип "Лучше там, где нас нет". На практике этот принцип представляет собой самый интересный тип программного обеспечения, так как исходный код программы неизвестен. Причины для исследования чужой программы может быть много. Другое дело, что иногда требуется защитить собственную программу от исследования, чтобы никто не смог ее

повторить, изменить или дополнить в кратчайшие сроки.

## Запутывание кода

Наиболее часто встречающийся метод защиты ПО от исследования – обфускация, или запутывание кода. Под этим термином подразумевается приведение исполняемого кода к виду, сохраняющему функциональность программы, но затрудняющему анализ и понимание алгоритмов работы. Другими словами, запутывание так изменяет программу, что ее обратное преобразование будет экономически невыгодным (а физически очень трудновыполнимым). В основном этот способ защиты приложения используется для защиты программ от воссоздания исходного кода (декомпиляции) и незаконного использования, нарушения авторских прав программистов. Основная функция защиты программного обеспечения заключается не только в том, чтобы приложение нельзя было незаконно использовать, копировать или модифицировать, но и в том, чтобы не дать хакеру возможности изучить эту программу, применив излюбленный метод – пошаговый режим отладки. Тут уже может помочь нетипичное расположение стека (область памяти, где хранятся данные программы), его размер или варианты применения. Ведь при анализе с помощью специальных программ хакер может отбросить ненужный кусок кода или данных, в результате функционирование программы может оказаться невозможным или неправильным.

С помощью запутывания можно перемешать в программе куски кода или действия так, что логика работы становится совершенно непонятной. Кроме того, при запуты-



вании могут вставляться новые куски неисполняемого (неиспользуемого) кода, а существующие блоки кода могут быть модифицированы таким образом, чтобы они использовались в нескольких частях программы одновременно.

**Методы защиты от исследования**

Иногда при защите программного продукта используется архивация данных программы, которые находятся в стеке. Для усложнения работы взломщиков в исполняемый код встраиваются "пустышки", которые выполняют некоторую сложную и на первый взгляд важную работу, но на самом деле не имеют никакого отношения к логике работы. Иногда используется такой метод, как "общая переменная", когда одна и та же переменная в разных частях алгоритма может употребляться для разных нужд (в разных функциях). Другой похожий метод – "разделяемая переменная" (для усложнения исследования программы одну переменную

заменяют функцией от набора других переменных). В последнем случае при изменении одной из переменных, входящих в набор, меняется значение функции, а следовательно, и искомым переменной. Обычно в набор включаются константы, другие переменные, адреса памяти, а также контрольные суммы отдельных блоков кода. Таким образом, меняя одну произвольную инструкцию в одном из блоков кода, можно повлиять на функциональность совершенно других частей программы. Также используется шифрование содержимого файлов данных, при котором защищенные файлы переносятся в защищенный контейнер. Особенностью защищенного контейнера является то, что к нему можно обращаться только из защищенного приложения.

В некоторых случаях применяется метод самогенерируемого кода, когда массив данных может быть сам по себе исполняемым кодом или смысловым текстом, но после некоторых операций он стано-

вится участком программы, выполняющим важные функции. Также используется полиморфный код, который при исполнении может изменять сам себя. Шифрование же позволяет изменить исполняемый код до полной неузнаваемости.

Используя защиту программного обеспечения от исследований, нужно иметь в виду, что лучше не только запутывать отдельные части кода, но и защищать всю программу целиком. Ведь защиту исполняемого кода можно дополнять другими видами защиты, например, при распространении приложения на дисках или через Интернет и т.п. Кроме того, защита кода должна быть максимально незаметной, замаскированной, запутывание не должно иметь регулярную структуру (иначе можно будет изучить алгоритм запутывания и разработать программу, выполняющую обратные преобразования).

Используя защиту программного обеспечения от исследований, нужно иметь в виду, что лучше не только запутывать отдельные части кода, но и защищать всю программу целиком. Ведь защиту исполняемого кода можно дополнять другими видами защиты, например, при распространении приложения на дисках или через Интернет и т.п. Кроме того, защита кода должна быть максимально незаметной, замаскированной, запутывание не должно иметь регулярную структуру (иначе можно будет изучить алгоритм запутывания и разработать программу, выполняющую обратные преобразования).

**Ваше мнение и вопросы присылайте по адресу [infosec@groteck.ru](mailto:infosec@groteck.ru)**

Какую воду вы пьёте?



Какой трафик потребляет ваша компания?

**eSafe** комплексное решение для очистки корпоративного трафика от вредоносного контента на уровне шлюза

EXTREME CAPACITY. MAXIMUM SECURITY.

- Полная очистка почтового и Web-трафика от всех видов вредоносного и нежелательного контента (на проводной скорости).
- URL-фильтрация по 62 категориям с использованием баз лидера отрасли – компании IBM.
- Избирательное блокирование Интернет-приложений (программ для мгновенного обмена сообщениями (IM), клиентов файлообменных сетей (P2P), Skype, программ удаленного управления и т. д.) по сигнатурам протоколов.
- Аудит состояния систем обеспечения контент-безопасности.

**eSafe уже используют:** Федеральный Депозитный банк, ВТБ, банк «Возрождение», МЕТРОБАНК, Метрокомбанк и многие другие.

«С внедрением eSafe у нас появился надежный инструмент для определения используемых приложений, выявления опасных сайтов, оценки характера передаваемой и получаемой информации. Это позволяет пресечь потенциально опасные действия со стороны сотрудников, исправить недочеты в корпоративной политике безопасности, а также выявить ее нарушения»

Дмитрий Васильевский,  
Заместитель начальника Службы информационной безопасности – начальник отдела анализа и сопровождения систем обеспечения информационной безопасности  
Банк «Возрождение»

**Aladdin**  
SECURITY SOLUTIONS

e-mail: [aladdin@aladdin.ru](mailto:aladdin@aladdin.ru)  
tel.: +7 (495) 223-0001

Узнайте больше: [www.aladdin.ru](http://www.aladdin.ru)