Высшее профессиональное образование
БАКАЛАВРИАТ

В. Г. ПРОСКУРИН

ЗАЩИТА ПРОГРАММ И ДАННЫХ

Допущено
Учебно-методическим объединением по образованию в области информационной безопасности в качестве учебного пособия для студентов высших учебных заведений, обучающихся по направлению подготовки 090900 «Информационная безопасность» (бакалавр) и специальностям 090301 «Компьютерная безопасность», 090303 «Информационная безопасность автоматизированных систем»

2-е издание, стереотипное

ACADEMIA

Москва
Издательский центр «Академия»
2012

УДК 004(075.8)
ББК 32.973-018.2я73
П824

Рецензенты:
зав. кафедрой «Информационная безопасность» Московского института электроники и математики (Технического университета), канд. техн. наук, доцент *А. Б. Лось*;
зав. кафедрой «Информационная безопасность компьютерных систем» Санкт-Петербургского государственного политехнического университета, д-р техн. наук, проф. *П. Д. Зегжда*

Проскурин В. Г.

П824 Защита программ и данных : учеб. пособие для студ. учреждений высш. проф. образования / В. Г. Проскурин. — 2-е изд., стер. — М. : Издательский центр «Академия», 2012. — 208 с. — (Сер. Бакалавриат)

ISBN 978-5-7695-9288-1

Учебное пособие создано в соответствии с Федеральным государственным образовательным стандартом по направлению подготовки «Информационная безопасность» (квалификация «бакалавр»).

Подробно рассмотрены средства и методы анализа программных реализаций, а также защиты программ от анализа. Рассмотрены модели взаимодействия программных закладок с атакуемыми компьютерными системами, предпосылки к внедрению и методы внедрения программных закладок, средства и методы защиты от программных закладок. Отдельно рассмотрен наиболее многочисленный на сегодняшний день класс программных закладок — компьютерные вирусы. Подробно описаны средства и методы реализации комплексного подхода к решению задачи организации антивирусной защиты. Изложение теоретического материала иллюстрируется многочисленными практическими примерами. В конце каждого раздела приведен перечень вопросов для самопроверки, в конце пособия — методические рекомендации по его изучению.

Для студентов учреждений высшего профессионального образования.

УДК 004(075.8)
ББК 32.973-018.2я73

ISBN 978-5-7695-9288-1

ПРЕДИСЛОВИЕ

В настоящее время дисциплина «Защита программ и данных», предусмотренная федеральным стандартом высшего профессионального образования по специальности «Компьютерная безопасность» (квалификация специалист), практически не обеспечена учебно-методической литературой. Единственное учебное пособие по данной дисциплине, имеющее гриф Министерства образования Российской Федерации, было издано в 1999 г. и к настоящему времени сильно устарело.

Предлагаемое учебное пособие призвано заполнить данный пробел в методическом обеспечении специальности «Компьютерная безопасность», а также смежных с ней специальностей. Пособие построено на основе 13-летнего опыта преподавания дисциплины «Защита программ и данных» в Институте криптографии связи и информатики при Академии ФСБ России (ИКСИ). Согласно учебному плану факультета информационной безопасности ИКСИ на изучение слушателями дисциплины «Защита программ и данных» отводится 7-й семестр, в конце семестра студенты сдают зачет, в течение семестра — выполняют домашнее задание.

В рамках основного лекционного курса изучаются две основные темы: «Анализ программных реализаций» и «Программные закладки». В ходе практических занятий могут быть рассмотрены (на усмотрение преподавателя, ведущего практические занятия по данному курсу) дополнительные темы, например «Защита от копирования» или «Анализ остаточной информации». Вносить в основной лекционный курс эти и другие узкоспециализированные аспекты защиты программ и данных представляется нецелесообразным.

Детальное изучение средств и методов анализа программных реализаций целесообразно по следующим причинам.

Навыки анализа программных реализаций требуются выпускнику при решении целого ряда практических задач, например:

- экспертиза качества реализации программных и программно-аппаратных средств обеспечения информационной безопасности;
- исследование программного обеспечения на предмет наличия недокументированных возможностей;
- выявление уязвимостей программного обеспечения;

- выявление вредоносного программного обеспечения, оценка опасности обнаруженных вредоносных программ, планирование работ по локализации последствий и пресечению обнаруженной атаки.

Кроме того, владение навыками анализа программных реализаций упрощает отладку собственноручно написанного программного кода и тем самым повышает производительность труда программиста.

Программные закладки (в особенности компьютерные вирусы) на сегодняшний день являются одной из наиболее масштабных и опасных угроз информационной безопасности. Трудно представить себе объект информатизации, для которого данная угроза была бы неактуальна. К сожалению, работы по выявлению и уничтожению программных закладок, как правило, сводятся к бездумному применению программных и программно-аппаратных антивирусных комплексов, при этом планирование мероприятий по обеспечению защиты от данной угрозы либо осуществляется чисто формально, либо не осуществляется вообще. В значительной степени это обусловлено недостатком специалистов, обладающих квалификацией, позволяющей эффективно планировать и поддерживать адекватную политику антивирусной защиты компьютеров и компьютерных сетей. Настоящее учебное пособие призвано устранить данный недостаток хотя бы частично.

Стоит отметить, что большая часть дидактических единиц, определявших данную дисциплину в федеральных государственных образовательных стандартах высшего профессионального образования (ФГОС ВПО) первого и второго поколений, за прошедшие годы либо утратили актуальность, либо переместились в другие общепрофессиональные и специальные дисциплины. В связи с этим исторически сложившееся название дисциплины «Защита программ и данных» выглядит несколько устаревшим, сейчас более логично выглядело бы название «Защита программ». Однако название «Защита программ и данных» является общепринятым, присутствует в стандартах ФГОС ВПО третьего поколения и вряд ли изменится в обозримом будущем. В конце концов, современное толкование термина «Информатика» тоже далеко ушло от того, что понималось под этим термином в начале 1990-х гг. (автору в свое время довелось изучать в курсе «Основы информатики» исчисление предикатов, машины Тьюринга и метод резолюций).

Пособие предназначено для преподавания следующих дисциплин:

- «Защита программ и данных» специальности «Компьютерная безопасность»;
- «Программно-аппаратные средства защиты обеспечения информационной безопасности» для специальностей «Информационная безопасность телекоммуникационных систем» и 090303 «Информационная безопасность автоматизированных систем»;

- «Программно-аппаратные средства защиты информации» для бакалавров направления подготовки «Информационная безопасность».

Пособие разработано при поддержке Министерства образования и науки Российской Федерации.

Глава 1

АНАЛИЗ ПРОГРАММНЫХ РЕАЛИЗАЦИЙ, ЗАЩИТА ПРОГРАММ ОТ АНАЛИЗА

1.1. Общие сведения

Задачу изучения программы в общем случае можно сформулировать следующим образом. Мы имеем бинарный код программы (например, EXE-файл) и минимальную информацию о том, что эта программа делает. Нам нужно получить более детальную информацию о функционировании этой программы. Другими словами, мы знаем, *что* делает программа, и хотим узнать, *как* она это делает.

Конечно, во многих случаях аналитику доступна более детальная информация об анализируемой программе (техническая документация, исходный текст и т. д.), что существенно упрощает процесс анализа. Но мы будем рассматривать наиболее общий (и наиболее сложный) случай, когда ничего, кроме кода программы, аналитику неизвестно.

Понятие «программа» здесь и далее мы будем понимать в широком смысле. Под программой мы будем подразумевать не только бинарный исполняемый файл, но и библиотеку функций, драйвер устройства и т. д. Обычно мы будем употреблять слово «программа» в единственном числе, но это не означает, что изложенную далее методику нельзя применять к программным комплексам, включающим в себя более одной программы.

Хотя в дальнейшем мы будем рассматривать применение данной методики только к системам защиты информации, эта методика может быть применена к любым программам.

Актуальность задачи анализа программных реализаций алгоритмов защиты обусловливается следующими факторами.

1. *Компьютеризация всех областей национальной экономики России.* В настоящее время компьютерные технологии активно применяются практически во всех новых областях нашей жизни. Автоматизированные средства обработки информации используются как в высших эшелонах государственной власти и управления, так и в многочисленных предприятиях и организациях, банках и т. п. Практически каждая организациям имеет у себя конфиденциальную информацию, которую надо защищать.

2. *Многообразие программных средств обработки информации и используемых в них средств защиты.* Средства защиты встраиваются во многие программные продукты самого различного назначения (операционные системы, системы управления базы данных,

системы электронного документооборота, разнообразные утилиты и т. п.). При этом подробные технические описания, а также данные независмых экспертиз качества используемой системы защиты далеко не всегда доступны пользователю. В этих условия пользователю информационной системы жизненно важно знать, насколько надежна применяемая им защита.

3. *Большой разброс в уровне подготовленности разработчиков*. Разработкой систем защиты информации занимаются различные люди и организации, порой разительно отличающиеся по своему научно-техническому потенциалу, уровню подготовки, техническим возможностям. К их числу зачастую относятся любители, т. е. лица, не являющиеся специалистами в области защиты информации. Нельзя исключить с их стороны ошибки при выборе алгоритмов защиты и просто программистские ошибки в процессе разработки программного обеспечения. Хотя сегодня откровенно слабые решения в области защиты информации встречаются гораздо реже, чем 10 — 15 лет назад, они все же встречаются. Одним из последних примеров такого рода, получивших широкую огласку, является система фильтрации информации порнографического характера на «Школьном портале», которая считала поисковый запрос «мокрый снег» непристойным, а целый ряд явно порнографических запросов — вполне допустимыми для школьника.

4. *Широкое распространение программных средств зарубежного производства*. Среди всех программных средств, применяемых в России в настоящее время, львиную долю составляют программные средства зарубежного производства. В ряде стран, например в США и Франции, имеются ограничения на экспорт стойких систем защиты информации. Бесспорно высокий уровень квалификации подобных разработчиков, удобство эксплуатации информационной системы, сконструированной ведущими зарубежными фирмами, еще не дает гарантии, что информационная система обладает надежной защитой. Кроме того, для программных средств, применяемых в критически важных сегментах информационной инфраструктуры Российской Федерации, весьма важной является задача заблаговременного выявления в используемых программных средствах недокументированных возможностей, которые в случае начала информационной войны могут послужить информационным оружием.

5. *Регулярное появление новых версий программных продуктов, которые могут отличаться и средствами защиты*. Современные средства обработки информации — чрезвычайно быстро развивающаяся сфера деятельности человека, поэтому проверка надежности средств защиты информационной системы не может рассматриваться как кратковременный эпизод в жизни организации. К этому вопросу приходится возвращаться неоднократно, в связи с чем стоит задача разработки эффективных средств анализа, максимальной автоматизации наиболее трудоемких его этапов.

Следует отметить, что описываемую в данной главе методику анализа программ можно применять не только для защиты информации, но и для «нападения» — получения незаконных копий программ, встраивания в программы закладок и вирусов и т. д. Поскольку автор не рассматривает данную книгу как пособие для начинающего хакера, вопросы, связанные с применением методов изучения программ для выполнения подобных незаконных действий, рассматриваться не будут.

Работа по анализу программы включает в себя три основных этапа.

1. *Подготовительный этап*. На данном этапе аналитик проводит первичное знакомство с анализируемой программой, изучает доступную документацию, планирует дальнейшие исследования, подбирает коллектив и организует его работу. Важность этого этапа нельзя недооценивать. Эффективность дальнейшей работы очень сильно зависит от того, насколько полная информация об анализируемой программе была получена на первом этапе. Если аналитик сумел получить исходный текст анализируемой программы, задача анализа программы в большинстве случаев решается тривиально. Наличие у аналитика отладочной информации об анализируемой программе также существенно упрощает дальнейшую работу.

2. *Восстановление алгоритмов функционирования программы*. На данном этапе, собственно, и производится изучение программы. Методам, применяемым на этом этапе, посвящена бóльшая часть данной главы.

3. *Проверка полученных результатов*. На данном этапе аналитик проверяет результаты проведенных исследований. Обычно эта проверка заключается в написании тестовой программы, которая реализует восстановленные алгоритмы анализируемой программы. Если поведение тестовой программы не отличается от поведения анализируемой в отношении анализируемых алгоритмов, задачу можно считать решенной. Если же поведение тестовой программы отличается от поведения анализируемой, это означает, что в анализе программы допущены ошибки, которые необходимо устранить. Как правило, правильно восстановить анализируемые алгоритмы с первого раза не удается.

В настоящее время сформировались следующие подходы к восстановлению алгоритмов, реализуемых программой:

- метод экспериментов;
- статический метод;
- динамический метод.

В методе экспериментов программа рассматривается как «черный ящик», осуществляющий определенные преобразования в зависимости от поступающего на него входа. Аналитик проводит многократные эксперименты, манипулируя входными данными, анализируя и сравнивая получаемые результаты. На основе этих экспериментов он восстанавливает, а точнее, угадывает алгоритмы преобразований.

В статическом методе по файлам программного обеспечения восстанавливаются основные элементы исходной программы, которая затем анализируется с целью получения описания собственно алгоритмов защиты. Основным инструментом статического метода служат программы дизассемблирования, восстанавливающие по исполняемым файлам листинги программы на языке ассемблер. Одним из наиболее удачных дизассемблеров в настоящее время считается пакет IDA.

В динамическом методе анализируемая программа запускается под контролем других специализированных программных средств. Наличие этих средств позволяет прогонять программу в пошаговом режиме, останавливать ее работу при осуществлении тех или иных событий, что значительно облегчает поиск и анализ фрагментов программы, реализующие преобразования, связанные с защитой информации. Основным инструментом динамического метода являются так называемые программы-отладчики. Вместе с тем могут использоваться и другие программные средства.

Перечисленные методы имеют свои достоинства и недостатки и во многом дополняют друг друга. Порой между ними трудно провести четкую грань. Так, при реализации динамического метода нередко проводится «прозванивание» отдельных функций анализируемой программы, как и в методе экспериментов с «черным ящиком». Некоторые методы автоматизации статического метода предполагают эмуляцию выполнения отдельных процедур процессором в режиме интерпретатора в целях идентификации алгоритмов по результатам их реализации. При восстановлении алгоритмов защиты аналитик, как правило, использует все три подхода в комбинации. Какой из подходов играет ведущую роль, определяется спецификой конкретной задачи и предпочтениями аналитика. Многие аналитики отдают предпочтение динамическому методу, считая его наиболее эффективным.

1.2. Метод экспериментов с «черным ящиком»

Термин «черный ящик» взят из математической теории автоматов, где ставится задача, наблюдая вход и выход автомата, построить автомат, эквивалентный данному, т. е. такой, что, подавая одинаковый вход на исходный автомат и построенный нами, мы получим одинаковый выход.

Различают два варианта задачи:

1) вход на автомат является случайным, и его можно только наблюдать;

2) исследователь может по своему усмотрению задавать вход автомата и наблюдать выход (метод прозванивания).