Средство безопасной загрузки программных модулей

Владимир Аникеев, Михаил Марунин, Андрей Точилин (Нижегородская обл.), Олег Жердин (Челябинская обл.)

В статье представлены варианты структурного построения средств безопасной загрузки программных модулей в среде MS DOS/Windows.

Практическая эксплуатация программно-аппаратных средств значительно повышает роль обеспечения безопасности исполняемых файлов [1]. Основным требованием, предъявляемым к обеспечению безопасности исполняемых файлов, является ограничение их нелегального использования, а именно:

- невозможность нелегального запуска исполняемых файлов на выполнение;
- невозможность нелегального копирования исполняемых файлов;
- защита исполняемых файлов от заражения вирусом.

Кроме того, исполняемый файл должен быть защищён от действий дизассемблера, отладчика и трассировки по заданному прерыванию. Все эти требования должны учитываться в процессе создания и эксплуатации автоматизированных систем [2].

Обеспечить безопасность исполняемого файла от нелегального использования можно двумя основными способами:

- вставкой фрагмента проверочного кода в исполняемый файл;
- преобразованием исполняемого файла к неисполняемому виду и применением для загрузки не операционной среды, а некоторой программы, в теле которой и осуществляются необходимые проверки и преобразования.

Операционная система MS DOS поддерживает два типа исполняемых файлов: COM и EXE. Файлы типа COM – более простые в смысле обеспечения безопасности, они являются всего лишь двоичным образом задачи. При запуске такого файла MS DOS считывает его в память по смещению 100h и передаёт управление на его первый байт. В том же сегменте по смещению 0 записывается PSP (префикс программного сегмента) – структура данных, необходимая программе для получения доступа к элементам командной строки и др.

Файлы типа EXE более сложны по своей структуре. Они предназначены для создания программ, код или данные которых превышают по размеру 64 Кб (максимальный размер COM-файла). Поскольку максимальный размер сегмента в среде MS DOS также равен 64 Кб (что является ограничением на длину COM-файла), MS DOS записывает коды и данные из EXE-файла в несколько сегментов, используя информацию из структуры, находящейся в начале файла (из заголовка EXE-файла).

Файл EXE состоит из двух частей: управляющей информации для средства загрузки и собственно загрузочного модуля. Информация для средства загрузки расположена в начале файла и образует заголовок. За ним следует тело загрузочного модуля. Тело загрузочного модуля начинается на границе блока и представляет собой копию образа памяти задачи. Сложность обеспечения безопасности этого типа программ заключается в том, что, кроме модификации кода и добавления кода безопасности к файлу, приходится приводить данные в заголовке EXE-файла в соответствие с внесёнными в него изменениями для обеспечения правильной загрузки и выполнения основного кода после выполнения кода безопасности.

Необходимость дополнения готового исполняемого модуля некоторыми функциями, которые выполняются до начала работы основного модуля, встречается достаточно часто. Это может быть, например, просчёт контрольных сумм с целью обнаружения файлового вируса, установление парольного доступа к файлу или проверка наличия некоторой идентифицирующей информации (например, для защиты от копирования).

Модификация программы с целью добавления необходимого программного кода безопасности реализуется «вирусным» методом. Для программы в формате EXE-файла в конец файла добавляется необходимый код, корректируются точка входа и размер файла. Для программ в формате COM-файла (длина результирующего файла не превышает 64К) требуемый код добавляется в конец файла, а часть кода, находящегося в начале программы, корректируется для безусловного перехода на добавленный код.

Получив управление, добавленный код должен выполнить аутентификацию пользователя, персонального компьютера или дискеты. Эту функцию следует выполнять в закрытом от изучения коде. При положительном результате аутентификации управление передаётся основной программе. В противном случае выполнение программы прекращается, и может быть выведено сообщение о нелегальном действии.

Большинство программ загружаются в память, запускаются, а затем удаляются из памяти операционной системой. Если же для обеспечения безопасности программы от нелегального запуска преобразовать загрузочный модуль, то MS DOS не сможет выполнить загрузку. В этом случае загружаемая программа будет защищена от заражения вирусом, дизассемблирования и отладчика; копирование преобразованного файла также не приведёт к желаемому результату [3, 4].

Для загрузки программ MS DOS использует функцию 4Bh прерывания 21h, предполагая, что загрузочный модуль имеет определённую структуру. Естественно, в преобразованном модуле эта структура не соблюдается, и попытка загрузить такую программу окончится неудачей. Для запуска преобразованных программ необходимо средство загрузки (loader, загрузчик), которое будет считывать преобразованную программу в память, осуществлять её восстановление, выполнять

работу MS DOS по настройке адресов и заполнению служебных полей и после этого передавать ей управление. Загрузчик также может проверять легальность запуска преобразованных программ. После запуска программы загрузчик может «исчезать» из памяти, чтобы не отнимать ресурсы у программы.

Локализация проверочных кодов в средстве загрузки позволяет избежать увеличения размеров преобразованных программ. Если же средство загрузки не копировать на жёсткий диск и всегда запускать его с персонального носителя данных, который хранится у владельца преобразованных программ, то получить непреобразованную копию программ не сможет даже самый опытный хакер. В этом случае носитель с программой загрузки будет играть роль физического ключа для восстановления программ.

Трассировка программы по заданному прерыванию заключается в следующем: «программа-шпион» адресует вектор заданного прерывания на собственную функцию, которая до или после выполнения прерывания останавливает процесс выполнения программы и позволяет «программе-шпиону» ознакомиться с интересующей информацией.

Для защиты от трассировки можно предпринять следующие меры:

- не использовать прерывания в средстве загрузки;
- блокировать трассировку прерываний int 13h, int 40h, int 21h;
- после отработки значимого фрагмента кода приводить его в нерабочее состояние.

Для обеспечения безопасности средства загрузки должно быть исключено попадание его носителя посторонним лицам, а запуск с носителя необходимо осуществлять, указав в командной строке путь к преобразованному исполняемому файлу, например, *a:\loader c:\work\filename*.

Рассмотренное средство загрузки позволяет запускать на выполнение любые исполняемые файлы из командной строки, а его вызов может также осуществляться из другого файла.

Разработка программного обеспечения запуска исполняемых файлов проводилась по двум направлениям:

- разработка средства загрузки исполняемых файлов в среде MS DOS;
- разработка средства загрузки исполняемых файлов в среде MS Windows.

Блок-схема программы *LOADER* в среде MS DOS приведена на рисунке 1. Эту программу можно использовать и в среде MS Windows для запуска файлов, созданных для работы в среде MS DOS. В этом случае MS Windows открывает DOS-процесс для выполнения этих файлов.

Преобразование исполняемого файла может осуществляться по одному из известных алгоритмов преобразования, например ГОСТ 28147-89, при этом конфиденциальные параметры преобразования следует расположить на носителе вместе с программой загрузки (в преобразованном виде) или получать их только в процессе работы, например, с внешнего устройства, через последовательный порт.

Для носителя можно использовать нестандартное форматирование или иные способы достижения безопасности. Потенциально опасным моментом является легальное считывание конфиденциальных параметров с носителя. Для этого используется прерывание int 13h. Для обеспечения защиты от возможного его перехвата следует заменить данное прерывание прямым обращением к BIOS.

Загрузчик *LOADER* защищён от трассировки отладчиком с помощью измерения времени выполнения фрагмента программы. Кроме того, если работа программы контролируется отладчиком, то клавиатура персонального компьютера будет блокирована. Что касается защиты от дизассемблера, то обеспечить безопасность средства загрузки достаточно сложно, и этот вопрос в данной статье не затрагивается.

В начале выполнения программы *LOADER* производится замер времени выполнения эталонного участка программы. Для получения отсчётов времени используется системный счётчик 0000h:046Ch. Если программа работает под управлением отладчика, то происходит выход (в этом случае можно сгенерировать коды с помощью счётчика случайных чисел). Иначе происходит переход на процедуру *func*, где проводится проверка правильности ввода из командной строки. Если при вводе обнаружена ошибка, то происходит выход из программы.

Далее, проводится подготовка среды (иногда применяется термин «окружение», Environment) для запускаемой программы. Окружение MS DOS – это область памяти длиной до 32 Кб, в которой MS DOS сохраняет переменные окружения *COMSPEC, PATH, PROMPT* и т.п. Каждая переменная окружения представляет собой текстовую строку. Каждая строка окружения оканчивается нулём; конец окружения обозначается двумя нулями подряд. Начиная с версии MS DOS 3.0, после окружения располагается дополнительная строка, содержащая полное имя запущенной программы (с указанием маршрута) и, возможно, параметры обращения к программе. Таким образом, чтобы найти имя запускаемого файла, необходимо отыскать в окружении MS DOS два нуля подряд – это признак расширенной части окружения. Слово, следующее за этим признаком, содержит количество переменных в расширенной части, за ним помещаются сами переменные. Например, в терминах ассемблера структура окружения может быть такой:

```
db 'COMPREC=C:\COMMAND.COM',0
  ; Переменная COMSPEC
db 'PATH=C:\;C:\DOS',0
  ; Переменная PATH
db 'PROMPT='$p$g',0
  ; Переменная PROMPT
dw 2
  ; Количество переменных в расши-
ренной части
db 'A:\LOADER',0
  ; Имя файла
```

Для каждой программы MS DOS создаёт отдельную копию окружения. Сегментный адрес среды помещается в префикс программного сегмента (PSP). После запуска программы нет необходимости выделять новую память под окружение запускаемой программы, а можно воспользоваться окружением, которое создала MS DOS для загрузчика при его запуске, поскольку само средство загрузки уже не нужно. Для этого достаточно после окружения средства загрузки вписать имя запускаемой программы.

Если подобным же образом повторно использовать и PSP-средства загрузки (для этого надо загружать запускаемую программу в то же место памяти, куда был загружен *LOADER*), то подготовку среды можно считать законченной, поскольку в PSP уже стоит правильный адрес окружения.

Имя может оказаться длиннее, чем полное имя средства загрузки, и не поместиться в область памяти окружения. В этом случае можно попробовать расширить блок памяти окружения

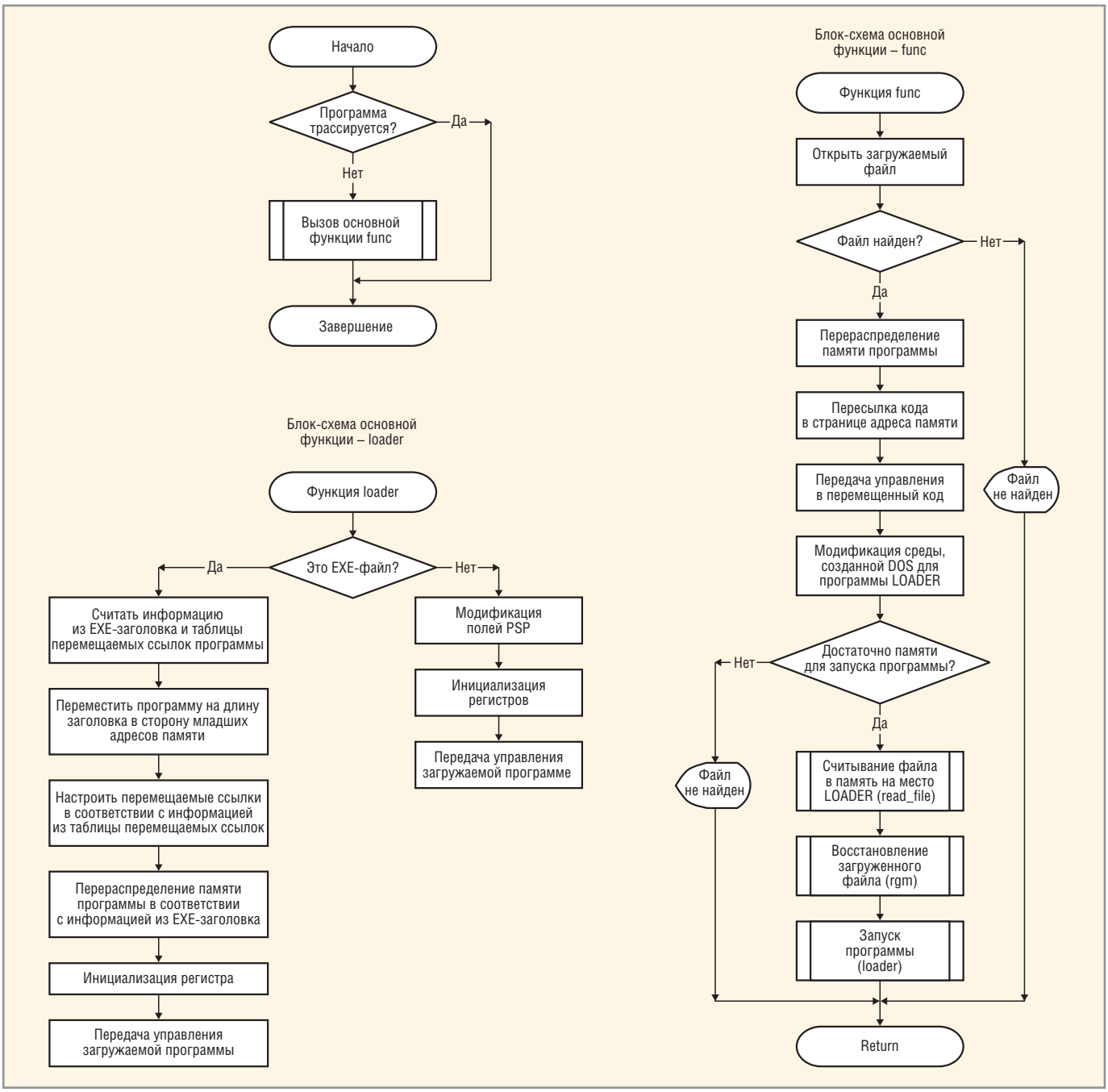

Рис. 1. Блок-схема программы *LOADER* в среде MS DOS

или запросить пользователя скопировать *LOADER* в один каталог с запускаемой программой и ещё раз запустить его. При этом длины полных имён средства загрузки и программы будут почти одинаковы.

На втором этапе работы необходимо загрузить программу в память, причём на то же место, куда был загружен *LOADER*. Сначала надо освободить место под загрузку программы. Для этого можно переместить код средства загрузки в старшие адреса памяти и передать туда управление. Затем следует определить длину загружаемого файла и записать её в поле длины области памяти блока MCB (блок управления памятью) того сегмента памяти, где изначально располагался загрузчик. Теперь можно считать программу в память, начиная со следующего после PSP средства загрузки параграфа. На третьем этапе производится восстановление программы в памяти.

Таким образом, алгоритм работы средства загрузки следующий. Сначала определяется формат загрузочного модуля: EXE или COM. Если программа имеет COM-формат, то достаточно заполнить некоторые поля PSP, соответствующим образом инициализировать регистры и передать управление загруженной программе с адреса CS:100h. С этого момента программа начнёт выполняться так, как если бы она была загружена MS DOS.

Если программа имеет формат EXE, то необходимо сделать несколько дополнительных шагов:

- считать информацию из EXE-заголовка и таблицы перемещаемых ссылок программы;
- переместить программу на длину заголовка в сторону младших адресов памяти;
- настроить перемещаемые ссылки в соответствии с информацией из таблицы перемещаемых ссылок;
- перераспределить память программы в соответствии с информацией из EXE-заголовка.

Только после этих шагов можно инициализировать регистры и передавать управление загруженной программе.

Поскольку при пересылке средства загрузки в старшие адреса адресного пространства никаких выделений памяти не производится, для MS DOS это перемещение остаётся незамеченным, следовательно, система будет считать всю память, расположенную после запускаемой программы, свободной. Таким образом, средство загрузки как бы «исчезает» из памяти после выполнения своей функции. По окончании работы программы, MS DOS либо очистит занимаемую программой память, либо оставит программу резидентной, в зависимости от того, как она завершает свою работу.

Рассмотрим методику работы средства загрузки исполняемых файлов для среды MS Windows. Данная методика позволяет легально запускать исполняемые файлы для 32-разрядных операционных систем MS Windows 95 и Windows NT. Блок-схема программы *LOADER* в среде Windows приведена на рисунке 2. В её основе лежит преобразование исполняемого файла в соответствии с выбранным алгоритмом преобразования.

Перед запуском любой исполняемой программы, на которой установлена защита от нелегального запуска, происходит запуск программы средства загрузки, проверяющей полномочия пользователя, который производит запуск программы. В состав проверки полномочий входит:

- проверка пароля, введённого пользователем;
- проверка текущей конфигурации машины и операционной системы (защита от нелегального копирования программ).

После того как пользователь произвёл запуск исполняемого файла, на котором установлена защита, загрузчик запрашивает у пользователя пароль, длина которого ограничена 32 символами. После ввода пароля производится восстановление конфиденциальных параметров. Из восстановленных конфиденциальных параметров производится проверка правильности введённого пароля и полномочий пользователя. Если результат проверки отрицательный, программа автоматически завершит работу с данным пользователем.

При положительном результате проверки пароля происходит проверка конфигурации компьютера. Если данная проверка выявила несовпадения в конфигурации машины, программа автоматически завершит работу с данным пользователем. После всех проверок загрузчик производит восстановление запускаемого файла в оперативную память и передаёт управление программе, а сам выгружается из памяти.

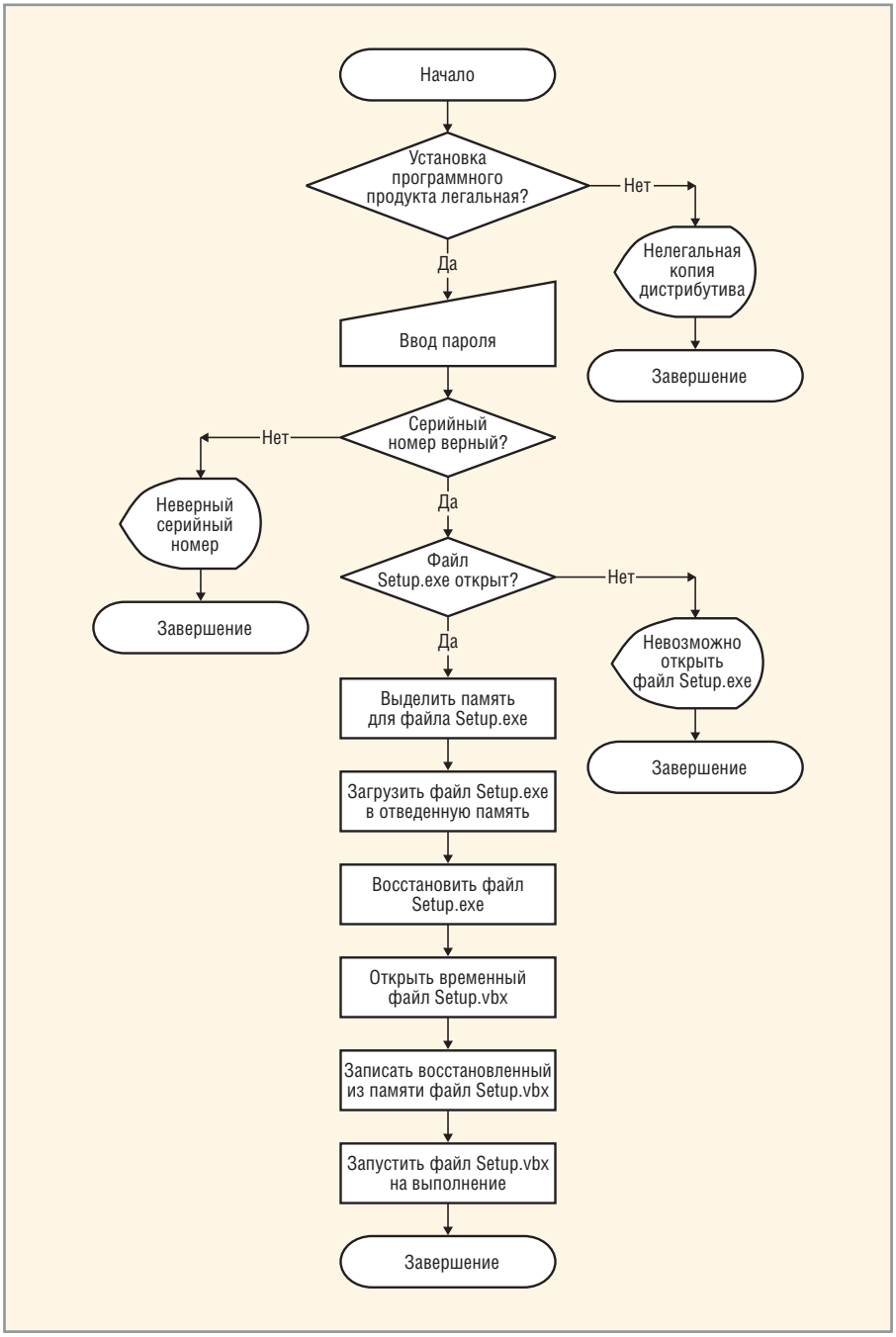

Рис. 2. Блок-схема программы *LOADER* в среде Windows

Таким образом, любая программа, на которой установлена защита от нелегального запуска, хранится на диске в преобразованном виде, и доступ к ней может получить практически любой пользователь компьютера. Но получить доступ к открытому исполняемому файлу практически невозможно, поскольку восстановление происходит непосредственно в память только на момент выполнения программы. По окончании сеанса работы программы вся занимаемая программой часть оперативной памяти очищается.

Литература

1. *Мартынов А.П., Фомченко В.Н.* Криптография и электроника. Под ред. *А.И. Астайкина*. ФГУП «РФЯЦ-ВНИИЭФ», 2006.
2. Защита от несанкционированного доступа к информации: Термины и определения, Руководящий документ Гостехкомиссии России. Военное издательство, 1992.
3. *Джордейн Р.* Справочник программиста персональных компьютеров типа IBM PC, XT и AT. Финансы и статистика, 1992.
4. *Щербаков А.* Защита от копирования. Эдэль, 1992.