ВОЕННО-КОСМИЧЕСКАЯ АКАДЕМИЯ ИМЕНИ А.Ф. МОЖАЙСКОГО

Кафедра № 27 Математического и программного обеспечения

УТВЕРЖДАЮ
Начальник 27 кафедры
полковник_____С.Войцеховский

«___» _____ 2018 г.

Автор: старший преподаватель 27 кафедры
кандидат технических наук
подполковник С.Краснов

Лекция № 10

Тема: «ЗАЩИТА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ОТ НЕСАНКЦИОНИРОВАННОГО КОПИРОВАНИЯ, ИССЛЕДОВАНИЯ И МОДИФИКАЦИИ»

по дисциплине: «Защита информации»

Обсуждено и одобрено на заседании 27 кафедры
протокол № __ «____» _____ 2018 г.

Санкт-Петербург
2018

Содержание занятия и время

Введение……..13-15 мин.

Учебные вопросы (основная часть):

1. Защита программ от копирования. – 40 мин.
2. Защита программ от исследования. – 20 мин.
3. Защита ПО от модификаций. – 10 мин.

Заключение……5-7 мин.

Литература:

Основная:

1. план-конспект.
2. Воробьев Е.Г. Защита информации в автоматизированных системах военного назначения. Учебное пособие – СПб, ВКА им.А.Ф. Можайского, 2007. – 252 с.
3. Соколов А.В., Шаньгин В.Ф. Защита информации в распределённых корпоративных сетях и системах. – М.: ДМК Пресс, 2002. – 656 с.
4. Головко Н.И., Новожилов С.В. Методы и средства защиты компьютерной информации. Учебное пособие. – Череповец: изд. ЧВИИРЭ, 2004г. – 184 с.
5. Скляров Д.В. Искусство защиты и взлома информации. – СПб. БХВ-Петербург 2004. – 271 с.

Дополнительная:

1. Пирогов В.Ю. Ассемблер и дизассемблирование. – С.-Пб, «БХВ-Петербург», 2006.
2. Питер Абель Ассемблер. Язык и программирование для IBM PC. – М.: Корона – ВЭК, 2007.
3. Юров В.И. Assembler. Учебник для вузов. 2-е изд. – СПб.: Питер, 2007. – 637 с.

Материально-техническое обеспечение:

1. Наглядные средства обучения - доска, мел.

Организационно-методические указания:

Во введении сформулировать тему лекции, цель и название изучаемых вопросов. При этом в течение 10 мин. произвести опрос курсантов по пройденному материалу в виде летучки № 1.

Метод проведения занятия – рассказ. В основной части сконцентрировать внимание курсантов на основных методах защиты программ от копирования, исследования и модификации.

За 3 – 5 мин. до конца занятия делаю обобщающие выводы, задаю контрольные вопросы для проверки, как военнослужащие усвоили тему занятия:

1. Какие особенности архитектуры ЭВМ могут использоваться в качестве эталонных характеристик?
2. В чём заключается суть метода защиты программы StarForce 2.0 для защиты дистрибутивов от копирования.
3. Перечислите простые методы защиты компакт дисков.

Отвечаю на вопросы по теме занятия, даю задание на самоподготовку.

Лекция № 10
«Защита ПО от несанкционированного копирования, исследования и модификации»

В. 1. Защита программ от копирования

Создание копий программных средств для изучения или несанкционированного использования осуществляется с помощью устройств вывода или каналов связи.

Одним из самых распространенных каналов несанкционированного копирования является использование накопителей на съемных магнитных носителях. Угроза несанкционированного копирования информации блокируется методами, которые могут быть распределены по двум группам [4]:

- методы, затрудняющие считывание скопированной информации;
- методы, препятствующие использованию информации.

1. Методы, затрудняющие считывание скопированной информации

Эти методы основываются на придании особенностей процессу записи информации, которые не позволяют считывать полученную копию на других накопителях, не входящих в защищаемую КС. Таким образом, эти методы направлены на создание совместимости накопителей только внутри объекта. В КС должна быть ЭВМ, имеющая в своем составе стандартные и нестандартные накопители. На этой ЭВМ осуществляется ввод (вывод) информации для обмена с другими КС, а также переписывается информация со стандартных носителей на нестандартные, и наоборот. Эти операции осуществляются под контролем администратора системы безопасности. Такая организация ввода-вывода информации существенно затрудняет действия злоумышленника не только при несанкционированном копировании, но и при попытках несанкционированного ввода информации.

Особенности работы накопителей на съемных магнитных носителях должны задаваться за счет изменения программных средств, поддерживающих их работу, а также за счет простых аппаратных регулировок и настроек. Такой подход позволит использовать серийные образцы накопителей.

Самым простым решением является нестандартная разметка (форматирование) носителя информации.

Изменение длины секторов, межсекторных расстояний, порядка нумерации секторов и некоторые другие способы нестандартного форматирования дискет затрудняют их использование стандартными средствами операционных систем. Нестандартное форматирование защищает только от стандартных средств работы с накопителями. Использование специальных программных средств (например, DISK EXPLORER для IBM-совместимых ПЭВМ) позволяет получить характеристики нестандартного форматирования.

Перепрограммирование контроллеров ВЗУ, аппаратные регулировки и настройки вызывают сбой оборудования при использовании носителей на стандартных ВЗУ, если форматирование и запись информации производились на нестандартном ВЗУ. В качестве примеров можно привести изменения стандартного алгоритма подсчета контрольной суммы и работы системы позиционирования накопителей на гибких магнитных дисках.

В контроллерах накопителей подсчитывается и записывается контрольная сумма данных сектора. Если изменить алгоритм подсчета контрольной суммы, то прочитать информацию на стандартном накопителе будет невозможно из-за сбоев.

Позиционирование в накопителях на магнитных дисках осуществляется следующим образом. Определяется номер дорожки, на которой установлены магнитные головки. Вычисляется количество Дорожек, на которое необходимо переместить головки и направление движения. Если нумерацию дорожек магнитного диска начинать не с дорожек с максимальным радиусом, как это делается в стандартных накопителях, а нумеровать их в обратном направлении, то система позиционирования стандартного накопителя не сможет выполнять свои функции при установке на него такого диска. Направление движения будет задаваться в направлении, обратном фактически записанным на дискете номерам дорожек, и успешное завершение позиционирования невозможно.

Выбор конкретного метода изменения алгоритма работы ВЗЧ (или их композиции) осуществляется с учетом удобства практической реализации и сложности повторения алгоритма злоумышленником. При разработке ВЗУ необходимо учитывать потребность использования устройств в двух режимах: в стандартном режиме и в режиме совместимости на уровне КС. Выбор одного из режимов, а также выбор конкретного алгоритма нестандартного использования должен осуществляться, например, записью в ПЗУ двоичного кода. Число нестандартных режимов должно быт таким, чтобы исключался подбор режима методом перебора. Процесс смены режима должен исключать возможность автоматизированного подбора кода. Установку кода на ВЗУ всего объект должен производить администратор системы безопасности.

2. Методы, препятствующие использованию скопированной информации

Эта группа методов имеет целью затруднить использование полученных копированием данных. Скопированная информация может быть программой или данными. Данные и программы могут быть защищены, если они хранятся на ВЗУ в преобразованном *криптографическими методами* виде. Программы, кроме того, могут защищаться от несанкционированного исполнения и тиражирования, а также от исследования.

Наиболее действенным (после криптографического преобразования) методом противодействия несанкционированному выполнению скопированных программ является использование *блока контроля среды размещения программы*. Блок контроля среды размещения является дополнительной частью программ. Он создается при инсталляции (установке) программ. В него включаются эталонные характеристики среды, в которой размещается программа, а также средства получения и сравнения характеристик.

В качестве эталонных характеристик используются характеристики ЭВМ или носителя информации, или совместно, характеристики ЭВМ и носителя. С помощью эталонных характеристик программа связывается конкретной ЭВМ и (или) носителем информации. Программа может выполняться только на тех ЭВМ или запускаться только с тех носителей информации, характеристики которых совпадут с эталонными характеристиками, записанными в блоке контроля среды выполнения. В случае несоответствия текущих и эталонных показателей программа может блокировать свое дальнейшее выполнение.

В качестве эталонных характеристик ЭВМ могут использоваться особенности архитектуры:

- тип и частота центрального процессора,
- номер процессора (если он есть),
- состав и характеристики внешних устройств, особенности их подключения (например, уникальный MAC-адрес сетевой карты),
- показатели быстродействия жесткого диска и процессора,
- режимы работы блоков и устройств,
- и т. п.

Основным требованием к записанному на винчестер уникальному идентификатору является требование, согласно которому данный идентификатор не должен копироваться стандартным способом.

Для того чтобы злоумышленник не мог свободно изменить эти эталонные характеристики, они пред помещением их на любой носитель могут предварительно шифроваться.

Например, **зашифрованные эталонные характеристики** аппаратно-программной среды компьютера могут быть занесены в следующие области жесткого диска:

- в любые свободные места области данных:
- в созданный для этой цели отдельный файл;
- в отдельные кластеры, которые должны помечаться затем как зарезервированные под операционную систему или дефектные;
- в зарезервированные сектора системной области винчестера;
- непосредственно в файлы размещения защищаемой программной системы, например, в файл настройки ее параметров функционирования

Не копируемый стандартным образом идентификатор может помещаться и на диск, к которому должна будет обращаться при каждом своем запуске защищаемая от копирования программа. Такой диск называют ***ключевым***. Кроме того, защищаемая от копирования программа

может быть привязана и к уникальным характеристикам диска. Следует учитывать, что при использовании ключевого диска значительно увеличивается неудобство работы пользователя, так как он всегда должен вставлять этот диск в дисковод перед запуском защищаемой от копирования программы.

Общий *алгоритм механизма защиты от несанкционированного использования программ* в «чужой» среде размещения сводится к выполнению следующих шагов:

Шаг 1. Запоминание множества индивидуальных эталонных характеристик ЭВМ и (или) съемного носителя информации на этапе инсталляции защищаемой программы.

Шаг 2. При запуске защищенной программы управление передается на блок контроля среды размещения. Блок осуществляет сбор и сравнение характеристик среды размещения с контрольными характеристиками.

Шаг 3. Если сравнение прошло успешно, то программа выполняется, иначе - отказ в выполнении. Отказ в выполнении может быть дополнен выполнением деструктивных действий в отношении этой программы, приводящих к невозможности выполнения этой Программы, если такую самоликвидацию позволяет выполнить ОС.

Наиболее высокий уровень защиты программ от копирования достигается при *комбинировании различных способов привязки* к уникальным характеристикам аппаратно-программной среды компьютера.

Различают следующие **способы привязки к аппаратной конфигурации персонального IBM-совместимого компьютера:**

1) привязка к особенностям постоянного запоминающего устройства компьютера (ROM BIOS);

2) привязка к списку компьютерного оборудования.

В первом случае в качестве эталонных характеристик выступают контрольная сумма или дата изготовления BIOS. Дата изготовления BIOS хранится в восьми байтах внутренней памяти компьютера по адресу FOOO:FFF516.

При привязке к списку компьютерного оборудования в качестве эталонных характеристик выступает сам список оборудования, который можно получить путем использования соответствующей функции операционной системы.

Привязка программ к среде размещения требует повторной их инсталляции после проведения модернизации, изменения структуры или ремонта КС с заменой устройств. Для защиты от несанкционированного использования программ, могут применяться и электронные ключи. Электронный ключ HASP имеет размеры со спичечный коробок и подключается к параллельному порту принтера. Принтер подключается к компьютеру через электронный ключ. На работу принтера ключ не оказывает никакого влияния. Ключ распространяется с защищаемой программой. Программа в начале и в ходе выполнения считывает контрольную информацию из ключа. При отсутствии ключа выполнение программы блокируется.

Наиболее надежным способом привязки к аппаратной конфигурации компьютера является *привязка к уникальному номеру процессора*. Однако для большинства персональных IBM-совместимых компьютеров этот способ нереализуем по причине невозможности программного доступа к этому уникальному номеру.

Сложнее осуществляется ***привязка программ к дистрибутивным носителям информации***, так как они стандартны и не имеют индивидуальных признаков. Поэтому такие индивидуальные признаки создаются искусственно путем нанесения физических повреждений или изменением системной информации и структуры физических записей на носителе.

Основным требованием к свойствам дистрибутивных носителей, на основе которых формируются эталонные характеристики, является требование по уникальности. Согласно этому требованию контролируемыми свойствами не должны обладать другие носители информации и эти свойства не должны копироваться при копировании содержимого дистрибутивных носителей.

Несмотря на внешнюю несхожесть дискет, и компакт-дисков, очень многие методы создания некопируемых магнитных носителей были успешно перенесены на оптические диски.

Разумеется, большинство используемых компакт-дисков не допускает перезаписи, а те, что допускают, не позволяют изменять информацию в произвольном месте — можно только дописать новые данные или стереть все, что было записано ранее. Поэтому на компакт-дисках не делают

защиты со счетчиком установок.

Однако существует несколько способов создать диски, которые не копируются стандартными средствами или для которых существует надежный способ отличить копию от оригинала.

Прежде всего, стоит отметить, что получать доступ к содержимому компакт диска можно на нескольких уровнях [15].

1. Самый высокий уровень — это **уровень файловой системы**. Данные записываются на диск в определенном формате (например ISO-9660), и драйвер файловой системы компакт-диска (CD-ROM File System, CDFS) отвечает за то, чтобы представить содержимое диска в виде дерева каталогов и файлов. На этом уровне доступны такие операции, как получение списка файлов, открытие файла с определенным именем и чтение из него информации.
2. Второй уровень — **уровень секторов**. Грубо говоря, на этом уровне диск представляется как последовательность секторов, содержащих полезные данные, и таблица, описывающая содержимое диска (Table Of Contents, TOC). Доступны операции чтения TOC и секторов с заданными номерами.
3. Самый низкий уровень — **уровень команд контроллера**. Разные приводы CD-ROM могут иметь различия в доступном наборе команд, но только на этом уровне можно получить самую полную информацию, которую способен выдать привод относительно установленного диска. Использование этого уровня требует разработки драйвера.

Простые методы защиты компакт дисков

Если пользователь пытается сделать копию диска на уровне файловой системы, он сначала копирует все дерево каталогов и файлов на винчестер, а потом с помощью любой программы для создания компакт-дисков записывает диск, содержащий скопированные файлы. Программа, привязанная к компакт-диску, может *проверять метку тома диска*, которая не сохраняется при копировании файлов на винчестер.

Также *на* какой-нибудь *файл* можно *сделать несколько ссылок из разных директорий*. При этом легко добиться того, что суммарный размер файлов, скопированных на жесткий диск, превысит размер компакт-диска.

У какого-нибудь *файла* в директории компакт-диска можно *установить очень большой размер*, что не позволит прочитать этот файл, т. к. его данные просто не будут существовать.

Но все эти методы оказываются бессильны, если выполняется копирование диска на уровне секторов, а не на уровне файловой системы. Сейчас посекторное копирование поддерживает почти любая хорошая программа для создания компакт-дисков, например Nero Burning ROM, разработанная компанией Ahead Software AG.

Для борьбы с посекторным копированием применяются другие методы.

Отклонение от стандарта записи на диск

Иногда создатели защищенных дисков сознательно идут на нарушение стандарта, описывающего, как и что должно записываться на диск. Драйвер файловой системы использует далеко не всю информацию, которую можно получить о диске, а только то, что необходимо для определения размера диска и доступа к отдельным файлам. А программы посекторного копирования стремятся использовать максимум информации и часто отказываются работать с диском, если встречают противоречивые данные.

Правда, сейчас многие программы позволяют игнорировать некоторые нарушения стандарта и успешно копируют большинство дисков, защищенных таким способом.

Нарушения стандарта имеют еще один негативный аспект: они могут привести к тому, что диск не будет читаться на некоторых компьютерах, а то и вообще вызовет поломку привода CD-ROM.

Физические ошибки на диске

Если диск содержит сознательно внесенные нарушения в области данных, которые приводят

к ошибкам чтения, это не обязательно является нарушением стандарта — ошибки могли возникнуть и по естественным причинам, таким как загрязнение или механическое повреждение носителя. Следовательно, все приводы должны правильно отрабатывать ситуации, когда определенный сектор не может быть прочитан. А программа может принимать решение о подлинности диска на основании того, что некоторые строго определенные сектора не читаются.

Программы посекторного копирования часто отказываются продолжать работу с диском, если не могут прочитать очередной сектор, а если и создают новый диск, то на нем непрочитанные с оригинала секторы заполняются нулями или произвольными данными и больше не содержат ошибок.

Но существуют и другие программы, работающие с контроллером напрямую и выполняющие копирование не на уровне логических секторов, а, фактически, на уровне "сырых" данных, которые привод получает с диска. Иногда это называют побитовым копированием.

Наверное, самым популярным инструментом для побитового копирования компакт-дисков была программа CloneCD, разработанная компанией; Elaborate Bytes AG. Однако кроме CloneCD существует много других программ, успешно справляющихся с побитовым копированием практически любых компакт-дисков и создающих копии, принимаемые защитой за оригинал.

Также существуют программы, эмулирующие компакт-диски. Они позволяют использовать заранее сохраненный образ компакт-диска для того, чтобы изображать привод CD-ROM с установленным в нем диском. Многие эмуляторы, например Daemon Tools, Alcohol 120%, умеют передавать не только содержимое диска, но и все ошибки, которые используются для предотвращения копирования и проверки аутентичности компакт-диска.

Однако существуют и системы защиты компакт-дисков, которые долгое время весьма успешно противостояли программам побитового копирования и эмуляторам. Примером такой защиты может служить программа StarForce.

Про StarForce (SF), систему защиты программного обеспечения, распространяемого на дисках CD-ROM, от несанкционированного тиражирования, пока написано не очень много. На официальном интернет-сайте приводится следующее описание характеристик системы защиты StarForce Professional:

- компакт-диски, защищенные SF Professional версии 3.0 и выше, не копируются такими программами, как CloneCD, CDRWin, BlindWrite и им подобными. Защищенные приложения не запускаются на эмуляторах компактдисков, к которым относятся Daemon Tools, Alcohol 120%,Virtual CD-ROM и т. п.;
- используя комплект разработчика на этапе создания программного кода, можно значительно усилить защиту приложения против самых эффективных методов взлома;
- для встраивания защиты SF Professional не требуется специального технологического оборудования, нужен только компьютер и доступ на один из серверов StarForce;
- компакт-диски, защищенные SF Professional, максимально совместимы с разнообразными моделями существующих устройств CD/DVD-ROM. Это обусловлено тем, что в SF Professional используется уникальный метод определения подлинности диска без вмешательства в его физическую структуру;
- система защиты использует алфавитно-цифровой 24-значный ключ, который вводится пользователем защищенного программного приложения в процессе эксплуатации только один раз — в момент первого запуска. Ключ будет работать исключительно с дисками данной партии программного обеспечения.

Как же StarForce опознает оригинальный диск? Правильный ответ на этот вопрос знают только разработчики, однако в форуме поддержки Daemon Tools можно найти высказывание, что StarForce использует информацию об углах между секторами и метод получения этой информации совместим с 99.9 % приводов CD-ROM.

В работе [15] автор описывает проверку подлинности диска программой StarForce 2.0, которая состоит из нескольких главных шагов:

1. Чтение содержания диска (Table Of Content, TOC).
2. Чтение одиночных секторов с номерами 16, 17, 17 (дважды читается 17-ый сектор).

3. Чтение одиночных секторов с номерами 173117, 173099, 173081, 173063, 173045, 173027, 173009, 172991, 172973.

4. Чтение случайных 17 блоков по 8 секторов с номерами первого читаемого сектора в диапазоне примерно от 168100 до 173200.

5. SCSI-команда с кодом OxBB, описание которой не удалось найти в документации, но которая, скорее всего, отвечает за управление скоростью вращения привода.

6. Чтение одиночного сектора с номером 173117.

Причем если с первой попытки диск не опознан как оригинальный, то шаги 3 и 4 повторяются в цикле. Значит, после выполнения шага 4 вся информация, необходимая для аутентификации диска, уже получена.

Получив значения восьми интервалов (между девятью операциями чтения) и зная длительность и периодов обращения диска (полученную повторным чтением сектора), можно с большой точностью определить скорость вращения диска.

А дальше выполняется 17 чтений блоков со случайными номерами с целью измерения 16 интервалов времени. Если все интервалы хорошо (с малыми отклонениями) укладываются в определённую формулу, то диск признается подлинным. Если же отклонения от ожидаемых величин превышают некоторое пороговое значение, то проводится повторное вычисление скорости вращения и повторное измерение задержек между чтением блоков по 8 секторов.

Таким образом, метод защиты программы **StarForce 2.0** состоит в том, чтобы *сравнить задержки между чтениями соответствующих секторов CD-ROM* (DVD-ROM) *и эталонными значениями* (которые могут храниться в любом месте программы или компакт-диска). А для этого необходимо знать точные характеристики диска: радиус, на котором начинается спираль, и размер сектора.

Какой метод (или методы) защиты используется в программе StarForce 3.0, содержится в строжайшем секрете. Так что по состоянию на сегодняшний день, компакт-диски, защищенные при помощи StarForce версии 3.0 и выше, продолжают оставаться устойчивыми к взлому. В настоящее время последней версией программы доступной для пользователей является StarForce FrontLine 4.7.

B. 2. Защита программ от исследования

Изучение логики работы программы может выполняться в одном из двух режимов: статическом и динамическом.

Сущность статического режима заключается в изучении исходного текста программы. Для получения листингов исходного текста выполняемый программный модуль дизассемблируют, то есть получают из программы на машинном языке программу на языке Ассемблер [4].

Динамический режим изучения алгоритма программы предполагает выполнение трассировки программы. Под трассировкой программы понимается выполнение программы на ЭВМ с использованием специальных средств, позволяющих выполнять программу в пошаговом режиме, получать доступ к регистрам, областям памяти, производить остановку программы по определенным адресам и т. д. В динамическом режиме изучение алгоритма работы программы осуществляется либо в процессе трассировки, либо по данным трассировки, которые записаны в запоминающем устройстве.

Средства противодействия дизассемблированию не могут защитить программу от трассировки и наоборот: программы, защищенные только от трассировки, могут быть дизассемблированы. Поэтому для защиты программ от изучения необходимо иметь средства противодействия, как дизассемблированию, так и трассировке. Существует несколько методов противодействия дизассемблированию:

- шифрование;
- архивация;
- использование самогенерирующих кодов;
- "обман" дизассемблера.

Архивацию можно рассматривать как простейшее шифрование. Причем архивация может быть объединена с шифрованием. Комбинация таких методов позволяет получать надежно

закрытые компактные программы. Зашифрованную программу невозможно дизассемблировать без расшифрования. Зашифрование (расшифрование) программ может осуществляться аппаратными средствами или отдельными программами. Такое шифрование используется перед передачей программы по каналам связи или при хранении ее на ВЗУ. Дизассемблирование программ в этом случае возможно только при получении доступа к расшифрованной программе, находящейся в ОП перед ее выполнением (если считается, что преодолеть криптографическую защиту невозможно).

Другой подход к защите от дизассемблирования связан с **совмещением процесса расшифрования с процессом выполнения программ**. Если расшифрование всей программы осуществляется блоком, получающим управление первым, то такую программу расшифровать довольно просто. Гораздо сложнее расшифровать и дизассемблировать программу, которая поэтапно расшифровывает информацию, и этапы разнесены по ходу выполнения программы. Задача становится еще более сложной, если процесс расшифрования разнесен по тексту программы.

Сущность метода, основанного на **использовании самогенерируемых кодов**, заключается в том, что исполняемые коды программы получаются самой программой в процессе ее выполнения. Самогенерируемые коды получаются в результате определенных действий над специально выбранным массивом данных. В качестве исходных данных могут использоваться исполняемые коды самой программы или специально подготовленный массив данных. Данный метод показал свою высокую эффективность, но он сложен в реализации.

Под **"обманом" дизассемблера** понимают такой стиль программирования, который вызывает нарушение правильной работы стандартного дизассемблера за счет нестандартных приемов использования отдельных команд, нарушения общепринятых соглашений. "Обман" дизассемблера осуществляется несколькими способами:

- нестандартная структура программы;
- скрытые переходы, вызовы процедур, возвраты из них и из прерываний;
- переходы и вызовы подпрограмм по динамически изменяемым адресам;
- модификация исполняемых кодов.

Для дезориентации дизассемблера часто используются скрытые переходы, вызовы и возвраты за счет применения нестандартных возможностей команд.

Маскировка скрытых действий часто осуществляется с применением стеков.

Трассировка программ обычно осуществляется с помощью программных продуктов, называемых отладчиками. Основное назначение их - выявление ошибок в программах. При анализе алгоритмов программ используются такие возможности отладчиков как пошаговое (покомандное) выполнение программ, возможность останова в контрольной точке.

Покомандное выполнение осуществляется процессором при установке пошагового режима работы. Контрольной точкой называют любое место в программе, на котором обычное выполнение программы приостанавливается, и осуществляется переход в особый режим, например, в режим покомандного выполнения. Для реализации механизма контрольной точки обычно используется прерывание по соответствующей команде ЭВМ (для IBM-совместных ПЭВМ такой командой является команда INT). В современных процессорах можно использовать специальные регистры для установки нескольких контрольных точек при выполнении определенных операций: обращение к участку памяти, изменение участка памяти, выборка по определенному адресу, обращение к определенному порту ввода-вывода и т. д.

В. 3. Защита ПО от модификаций

Для повышения эффективности защиты программы от исследования можно внести в секретную часть программы *дополнительные функции безопасности, ориентированные на защиту от модификации*. К ним относятся:

- периодический подсчет контрольной суммы области оперативной памяти, занимаемой защищаемым исходным кодом, сравнение текущей контрольной суммы с предварительно сформированной эталонной и принятие необходимых мер в случае несовпадения;

- проверка количества занимаемой защищаемой программой оперативной памяти;
- сравнение с объемом, к которому программа адаптирована, и принятие необходимых мер в случае несоответствия;
- контроль времени выполнения отдельных частей программы;
- блокировка клавиатуры на время отработки особо секретных алгоритмов.

Для противодействия трассировке программы в ее состав вводятся следующие механизмы:

- изменение среды функционирования;
- модификация кодов программы;
- "случайные" переходы.

Под изменением среды функционирования понимается запрет или переопределение прерываний (если это возможно), изменение режимов работы, состояния управляющих регистров, триггеров и т. д. Такие изменения вынуждают аналитика отслеживать изменения и вручную восстанавливать среду функционирования.

Изменяющиеся коды программ, например, в процедурах приводят к тому, что каждое выполнение процедуры выполняется по различным ветвям алгоритма.

"Случайные" переходы выполняются за счет вычисления адресов переходов. Исходными данными для этого служат характеристики среды функционирования, контрольные суммы процедур (модифицируемых) и т. п. Включение таких механизмов в текст программ существенно усложняет изучение алгоритмов программ путем их трассировки.

При наличии современных средств отладки программ полностью исключить возможность изучения алгоритма программы невозможно, но существенно затруднить трассировку возможно. Основной задачей противодействия трассировке является увеличение числа и сложности ручных операций, которые необходимо выполнить программисту-аналитику.

Старший преподаватель 27 кафедры
подполковник С.Краснов