УВАЖАЕМЫЕ СТУДЕНТЫ! Изучите приведенную лекцию, законспектируйте основные понятия, дайте ответы на контрольные вопросы.

Ответы на вопросы, фотоотчет, предоставить преподавателю на e-mail: r.bigangel@gmail.com до 23.03.2023.

При возникновении вопросов по приведенному материалу обращаться по следующему номеру телефона: (072)111-37-59, (Viber, WhatsApp), vk.com: https://vk.com/daykini

ВНИМАНИЕ!!! При отправке работы, не забывайте указывать ФИО студента, наименование дисциплины, дата проведения занятия (по расписанию).

Лекция 44 (продолжение)
«Программы защиты от несанкционированного копирования»

ПЛАН

Введение

Дальнейшее развитие информационных технологий невозможно без создания новых программных средств различного назначения, баз данных, компьютерных средств обучения и других продуктов, предназначенных для корпоративного или персонального использования. При этом возникает проблема защиты авторских прав создателей и владельцев продуктов информационных технологий. Отсутствие такой защиты может привести к оттоку из сферы производства программного обеспечения части способных к творческой деятельности специалистов, снижению качества создаваемых информационных ресурсов и другим негативным социальным последствиям.

К сожалению, в настоящее время попытки нарушения авторских прав на объекты интеллектуальной собственности становятся достаточно регулярным и повсеместным явлением. Недостаточная эффективность правовых методов защиты интересов создателей и владельцев информационных ресурсов приводит к необходимости создания программных средств их защиты.

Цель данной лекции познакомить студентов с методами защиты от несанкционированного копирования, а также с принципами построения системы защиты от несанкционированного копирования.

1. СИСТЕМА ЗАЩИТЫ ОТ НЕСАНКЦИОНИРОВАННОГО КОПИРОВАНИЯ

Под системой защиты от несанкционированного использования и копирования (защиты авторских прав, или просто защиты, от копирования) понимается комплекс программных или программно-аппаратных средств, предназначенных для усложнения или запрещения нелегального распространения, использования и (или) изменения программных продуктов и иных информационных ресурсов. Термин «нелегальное» здесь понимается как производимое без согласия правообладателя. Нелегальное изменение информационного ресурса может потребоваться нарушителю для того, чтобы измененный им продукт не подпадал под действие законодательства о защите авторских прав.

Под надежностью системы защиты от несанкционированного копирования понимается ее способность противостоять попыткам изучения алгоритма ее работы и обхода реализованных в нем методов защиты. Очевидно, что любая программная или программно-аппаратная система защиты от копирования может быть преодолена за конечное время, так как процессорные команды системы защиты в момент своего исполнения присутствуют в оперативной памяти компьютера в открытом виде. Также очевидно, что надежность системы защиты равна надежности наименее защищенного из ее модулей.

Выделим принципы создания и использования систем защиты от копирования.

1. Учет условий распространения программных продуктов:

- распространение дистрибутивных файлов на магнитных носителях через сеть торговых агентов или через сеть Интернет с последующей установкой самим пользователем, который при этом может пытаться копировать дистрибутивные магнитные диски, исследовать алгоритм работы системы защиты при помощи специальных программных средств (отладчиков и дисассемблеров), пытаться нарушить условия лицензионного соглашения и установить продукт на большем числе компьютеров, пытаться смоделировать алгоритм работы системы защиты для изготовления аналогичного варианта дистрибутивных файлов и распространения их от своего имени;
- установка программного продукта официальным представителем правообладателя, при котором пользователь может пытаться нарушить условия лицензионного соглашения или исследовать алгоритм работы системы защиты;
- приобретение и использование программного продукта лицами или организациями, не заинтересованными в его нелегальном распространении среди

их коммерческих конкурентов в этом случае возможны только попытки несанкционированного использования продукта другими лицами;

- приобретение программного продукта только для снятия с него системы защиты.

2. Учет возможностей пользователей программного продукта по снятию с него системы защиты (наличие достаточных материальных ресурсов, возможность привлечения необходимых специалистов и т.п.).
3. Учет свойств распространяемого программного продукта (предполагаемого тиража, оптовой и розничной цены, частоты обновления, специализированное™ и сложности продукта, уровня послепродажного сервиса для легальных пользователей, возможности применения правовых санкций к нарушителю и др.).
4. Оценка возможных потерь при снятии защиты и нелегальном использовании.
5. Учет особенностей уровня знаний и квалификации лиц, снимающих систему защиты.
6. Постоянное обновление использованных в системе защиты средств.

При добавлении к программному продукту системы его защиты от копирования возможен выбор уже имеющейся системы, что минимизирует издержки на установку системы защиты. Однако имеющаяся система защиты от копирования будет более легко сниматься с программного продукта (в силу ее пристыкованности к нему, см. подразд. 1.5), а также может оказаться несовместимой с защищаемой программой и имеющимся у пользователя программно-аппаратным обеспечением. Поэтому более целесообразной является разработка специализированной системы защиты от копирования конкретного программного продукта, что, однако, более заметно увеличит затраты на его производство.

Основные требования, предъявляемые к системе защиты от копирования:

- обеспечение некопируемости дистрибутивных дисков стандартными средствами (для такого копирования нарушителю по требуется тщательное изучение структуры диска с помощью специализированных программных или программно-аппаратных средств);
- обеспечение невозможности применения стандартных отладчиков без дополнительных действий над машинным кодом программы или без применения специализированных программно-аппаратных средств (нарушитель должен быть специалистом высокой квалификации);
- обеспечение некорректного дисассемблирования машинного кода программы стандартными средствами (нарушителю потребуется использование или разработка специализированных дисассемблеров);
- обеспечение сложности изучения алгоритма распознавания индивидуальных параметров компьютера, на котором установлен программный продукт, и его пользователя или анализа применяемых аппаратных средств защиты (нарушителю будет сложно эмулировать легальную среду запуска защищаемой программы).

Выделим основные компоненты системы защиты программных продуктов от несанкционированного копирования:

- модуль проверки ключевой информации (некопируемой метки на дистрибутивном диске, уникального набора характеристик компьютера, идентифицирующей информации для легального пользователя) - может быть добавлен к исполнимому коду защищаемой программы по технологии

компьютерного вируса, в виде отдельного программного модуля или в виде отдельной функции проверки внутри защищаемой программы;

- модуль защиты от изучения алгоритма работы системы защиты;
- модуль согласования с работой функций защищаемой программы в случае ее санкционированного использования;
- модуль ответной реакции в случае попытки несанкционированного использования (как правило, включение такого модуля в состав системы защиты нецелесообразно по морально-этическим соображениям).

2. Методы, затрудняющие считывание скопированной информации

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:

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

Методы из первой группы 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 ВЗУ (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 быть таким, чтобы исключался подбор режима методом перебора. Процесс смены режима должен исключать возможность автоматизированного подбора кода. Установку кода на ВЗУ всего объекта должен производить администратор системы безопасности.

3. 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 характеристики среды, в которой размещается программа, а также средства получения и сравнения характеристик.

В качестве характеристик используются характеристики ЭВМ или носителя информации, или совместно, характеристики ЭВМ и носителя. С помощью

характеристик программа связывается с конкретной ЭВМ и (или) носителем информации. Программа может выполняться только на тех ЭВМ или запускаться только с тех носителей информации, характеристики которых совпадут с характеристиками, записанными в блоке контроля среды выполнения.

В качестве характеристик ЭВМ используются особенности архитектуры: тип и частота центрального процессора, номер процессора (если он есть), состав и характеристики внешних устройств, особенности их подключения, режимы работы блоков и устройств и т. п.

Сложнее осуществляется привязка программ к 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. Например, на гибких магнитных дисках могут прожигаться лазером отверстия, используется нестандартное форматирование, пометка некоторых секторов как дефектных. Приведенные средства защиты от несанкционированного использования дискет эффективны против стандартных способов создания копий (COPY, XCOPY, Diskcopy, Pctools, Norton Utilities в MS-DOS и др.).

Однако существуют программные средства (COPYWRITE, DISK EXPLORER), позволяющие создавать полностью идентичные копии дискет с воспроизведением всех уникальных характеристик. Все же приведенный метод защиты нельзя считать абсолютно неэффективным, так как трудоемкость преодоления защиты велика и требования, предъявляемые к квалификации взломщика, высоки.

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

Шаг 1. Запоминание множества индивидуальных контрольных 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.

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.

4. Основные функции средств защиты от копирования

При защите программ от несанкционированного копирования применяются методы, которые позволяют привносить в защищаемую программу функции привязки процесса выполнения кода программы только на тех ЭВМ, на которые они были инсталлированы. Инсталлированная программа для защиты от копирования при каждом запуске должна выполнять следующие действия:

- анализ аппаратно-программной среды компьютера, на котором она запущена, формирование на основе этого анализа текущих характеристик своей среды выполнения;
- проверка подлинности среды выполнения путем сравнения ее текущих характеристик с эталонными, хранящимися на винчестере;
- блокирование дальнейшей работы программы при несовпадении текущих характеристик с эталонными.

Этап проверки подлинности среды является одним из самых уязвимых с точки зрения защиты. Можно детально не разбираться с логикой защиты, а немного "подправить" результат сравнения, и защита будет снята.

При выполнении процесса проверки подлинности среды возможны три варианта: с использованием множества операторов сравнения того, что есть, с тем, что должно быть, с использованием механизма генерации исполняемых команд в зависимости от результатов работы защитного механизма и с использованием арифметических операций. При использовании механизма генерации исполняемых команд в первом байте хранится исходная ключевая контрольная сумма BIOS, во второй байт записывается подсчитанная контрольная сумма в процессе выполнения задачи. Затем осуществляется вычитание из значения первого байта значение второго байта, а полученный результат добавляется к каждой ячейки оперативной памяти в области операционной системы. Понятно, что если суммы не совпадут, то операционная система функционировать не будет. При использовании арифметических операций осуществляется преобразование над данными арифметического характера в зависимости от результатов работы защитного механизма.

Для снятия защиты от копирования применяют два основных метода: статический и динамический.

Статические методы предусматривают анализ текстов защищенных программ в естественном или преобразованном виде. Динамические методы предусматривают слежение за выполнением программы с помощью специальных средств снятия защиты от копирования.

5. Основные методы защиты от копирования

5.1. Криптографические методы

Для защиты инсталлируемой программы от копирования при помощи криптографических методов инсталлятор программы должен выполнить следующие функции:

- анализ аппаратно-программной среды компьютера, на котором должна будет выполняться инсталлируемая программа, и формирование на основе этого анализа эталонных характеристик среды выполнения программы;
- запись криптографически преобразованных эталонных характеристик аппаратно-программной среды компьютер на винчестер.

Преобразованные эталонные характеристики аппаратно-программной среды могут быть занесены в следующие области жесткого диска:

- в любые места области данных (в созданный для этого отдельный файл, в отдельные кластеры, которые должны помечаться затем в FAT 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.

Можно выделить два основных метода защиты от копирования с использованием криптографических приемов:

- с использованием односторонней функции;
- с использованием шифрования.

Односторонние функции это функции, для которых при любом x из области определения легко вычислить f(x), однако почти для всех y из ее области значений, найти y=f(x) вычислительно трудно.

Если эталонные характеристики программно-аппаратной среды представить в виде аргумента односторонней функции x, то y - есть "образ" этих характеристик, который хранится на винчестере и по значению которого вычислительно невозможно получить сами характеристики. Примером такой односторонней функции может служить функция дискретного возведения в степень, описанная в разделах 2.1 и 3.3 с размерностью операндов не менее 512 битов.

При шифровании эталонные характеристики шифруются по ключу, совпадающему с этими текущими характеристиками, а текущие характеристики среды выполнения программы для сравнения с эталонными также зашифровываются, но по ключу, совпадающему с этими текущими характеристиками. Таким образом, при сравнении эталонные и текущие характеристики находятся в зашифрованном виде и будут совпадать только в том случае, если исходные эталонные характеристики совпадают с исходными текущими.

5.2. Метод привязки к идентификатору

В случае если характеристики аппаратно-программной среды отсутствуют в явном виде или их определение значительно замедляет запуск программ или снижает удобство их использования, то для защиты программ от несанкционированного копирования можно использовать методов привязки к идентификатору, формируемому инсталлятором. Суть данного метода заключается в том, что на винчестере при инсталляции защищаемой от копирования программы формируется уникальный идентификатор, наличие которого затем проверяется инсталлированной программой при каждом ее запуске. При отсутствии или несовпадении этого идентификатора программа блокирует свое дальнейшее выполнение.

Основным требованием к записанному на винчестер уникальному идентификатору является требование, согласно которому данный идентификатор не должен копироваться стандартным способом. Для этого идентификатор целесообразно записывать в следующие области жесткого диска: в отдельные кластеры области данных, которые должны помечаться затем в FAT как зарезервированные под операционную систему или как дефектные; в зарезервированные сектора системной области винчестера.

Некопируемый стандартным образом идентификатор может помещаться на дискету, к которой должна будет обращаться при каждом своем запуске программа. Такую дискету называют ключевой. Кроме того, защищаемая от копирования программа может быть привязана и к уникальным характеристикам ключевой u1076 дискеты. Следует учитывать, что при использовании ключевой дискеты значительно увеличивается неудобство пользователя, так как он всегда должен вставлять в дисковод эту дискету перед запуском защищаемой от копирования программы.

5.3. Методы, основанные на работа с переходами и стеком

Данные методы основаны на включение в тело программы переходов по динамически изменяемым адресам и прерываниям, а также самогенерирующихся команд (например, команд, полученных с помощью сложения и вычитания). Кроме того, вместо команды безусловного перехода (JMP) может использоваться возврат из подпрограммы (RET). Предварительно в стек записывается адрес перехода, который в процессе работы программы модифицируется непосредственно в стеке.

При работе со стеком, стек определяется непосредственно в области исполняемых команд, что приводит к затиранию при работе со стеком. Этот способ применяется, когда не требуется повторное исполнение кода программы. Таким же способом можно генерировать исполняемые команды до начала вычислительного процесса.

5.4. Манипуляции с кодом программы

При манипуляциях с кодом программы можно привести два следующих способа:

- включение в тело программы "пустых" модулей;
- изменение защищаемой программы.

Первый способ заключается во включении в тело программы модулей, на которые имитируется передача управления, но реально никогда не осуществляется. Эти модули содержат большое количество команд, не имеющих никакого отношения к логике работы программы. Но "ненужность" этих программ не должна быть очевидна потенциальному злоумышленнику.

Второй способ заключается в изменении начала защищаемой программы таким образом, чтобы стандартный дизассемблер не смог ее правильно дизассемблировать. Например, такие программы, как Nota и Copylock, внедряя защитный механизм в защищаемый файл, полностью модифицируют исходный заголовок EXE-файла.

Все перечисленные методы были, в основном направлены на противодействия статическим способам снятия защиты от копирования. В

следующем подразделе рассмотрим методы противодействия динамическим способам снятия защиты.

6. Методы противодействия динамическим способам снятия защиты программ от копирования

Набор методов противодействия динамическим способам снятия защиты программ от копирования включает следующие методы.

Периодический подсчет контрольной суммы, занимаемой образом задачи области оперативной памяти, в процессе выполнения. Это позволяет:

- заметить изменения, внесенные в загрузочный модуль;
- в случае, если программу пытаются "раздеть", выявить контрольные точки, установленные отладчиком.

Проверка количества свободной памяти и сравнение и с тем объемом, к которому задача "привыкла" или "приучена". Это действия позволит застраховаться от слишком грубой слежки за программой с помощью резидентных модулей.

Проверка содержимого незадействованных для решения защищаемой программы областей памяти, которые не попадают под общее распределение оперативной памяти, доступной для программиста, что позволяет добиться "монопольного" режима работы программы.

Проверка содержимого векторов прерываний (особенно 13h и 21h) на наличие тех значений, к которым задача "приучена". Иногда бывает полезным сравнение первых команд операционной системы, отрабатывающих этим прерывания, с теми командами, которые там должны быть. Вместе с предварительной очисткой оперативной памяти проверка векторов прерываний и их принудительное восстановление позволяет избавиться от большинства присутствующих в памяти резидентных программ.

Переустановка векторов прерываний. Содержимое некоторых векторов прерываний (например, 13h и 21h) копируется в область свободных векторов. Соответственно изменяются и обращения к прерываниям. При этом слежение за известными векторами не даст желаемого результата. Например, самыми первыми исполняемыми командами программы копируется содержимое вектора 21h (4 байта) в вектор 60h, а вместо команд int 21h в программе везде записывается команда int 60h. В результате в явном виде в тексте программы нет ни одной команды работы с прерыванием 21h.

Постоянное чередование команд разрешения и запрещения прерывания, что затрудняет установку отладчиком контрольных точек.

Контроль времени выполнения отдельных частей программы, что позволяет выявить "остановы" в теле исполняемого модуля.

Многие перечисленные защитные средства могут быть реализованы исключительно на языке Ассемблер. Одна из основных отличительных особенностей этого языка заключается в том, что для него не существует ограничений в области работы со стеком, регистрами, памятью, портами ввода/вывода и т.п.

Автокорреляция представляет значительный интерес, поскольку дает некоторую числовую характеристику программы. По всей вероятности

автокорреляционные функции различного типа можно использовать и тестировании программ на технологическую безопасность, когда разработанную программу еще не с чем сравнивать на подобие с целью обнаружения программных дефектов. Таким образом, программы имеют целую иерархию структур, которые могут быть выявлены, измерены и использованы в качестве характеристик последовательности данных. При этом в ходе тестирования, измерения не должны зависеть от типа данных, хотя данные, имеющие структуру программы, должны обладать специфическими параметрами, позволяющими указать меру распознавания программы. Поэтому указанные методы позволяют в определенной мере выявить те изменения в программе, которые вносятся нарушителем либо в результате преднамеренной маскировки, либо преобразованием некоторых функций программы, либо включением модуля, характеристики которого отличаются от характеристик программы, а также позволяют оценить степень обеспечения безопасности программ при внесении программных закладок.

Контрольные вопросы

1. Перечислите основные требования, предъявляемые к системе защиты от копирования.
2. Назовите методы, затрудняющие считывание скопированной информации.
3. Отобразите схематично общий алгоритм механизма защиты от несанкционированного использования программ в «чужой» среде размещения.
4. Приведите примеры статических и динамических методов для снятия защиты от копирования.
5. Сделайте сравнительный анализ основных методов защиты от копирования.
6. Почему многие перечисленные в этой главе защитные средства могут быть реализованы исключительно на языке Ассемблер?