19. Технологическая и эксплуатационная безопасность ПО. Классификация систем защиты ПО. Системы защиты от несанкционированного копирования и изменения

Разработка терминологии в области обеспечения безопасности ПО является базисом для формирования нормативно-правового обеспечения и концептуальных основ по рассматриваемой проблеме.

Единая терминологическая база является ключом к единству взглядов в области, информационной безопасности, стимулирует скорейшее развитие методов и средств защиты ПО. Термины освещают основные понятия, используемые в рассматриваемой области на данный период времени. Определения освещают толкование конкретных форм, методов и средств обеспечения информационной безопасности.

Термины и определения

Непреднамеренный дефект - объективно и (или) субъективно образованный дефект, приводящий к получению неверных решений (результатов) или нарушению функционирования КС.

Преднамеренный дефект - криминальный дефект, внесенный субъектом для целенаправленного нарушения и (или) разрушения информационного ресурса.

Разрушающее программное средство (РПС) - совокупность программных и/или технических средств, предназначенных для нарушения (изменения) заданной технологии обработки информации и/или целенаправленного разрушения извне внутреннего состояния информационно-вычислительного процесса в КС.

Средства активного противодействия - средства защиты информационного ресурса КС, позволяющие блокировать канал утечки информации, разрушающие действия противника, минимизировать нанесенный ущерб и предотвращать дальнейшие деструктивные действия противника посредством ответного воздействия на его информационный ресурс.

Несанкционированный доступ - действия, приводящие к нарушению целостности, конфиденциальности и доступности информационных ресурсов.

Нарушитель (злоумышленник, противник) - субъект (субъекты), совершающие несанкционированный доступ к информационному ресурсу.

Модель угроз - вербальная, математическая, имитационная или натурная модель, формализующая параметры внутренних и внешних угроз безопасности ПО.

Оценка безопасности ПО - процесс получения количественных и/или качественных показателей информационной безопасности при учете преднамеренных и непреднамеренных дефектов в системе.

Система обеспечения информационной безопасности - объединенная совокупность мероприятий, методов и средств, создаваемых и поддерживаемых для обеспечения требуемого уровня безопасности информационного ресурса.

Информационная технология - упорядоченная совокупность организационных, технических и технологических процессов создания ПО и обработки, хранения и передачи информации.

Технологическая безопасность - свойство программного обеспечения и информации не быть преднамеренно искаженными и (или) начиненными избыточными модулями (структурами) диверсионного назначения на этапе создания КС.

Эксплуатационная безопасность - свойство программного обеспечения и информации не быть не санкционированно искаженными (измененными) на этапе их эксплуатации.

Жизненный цикл программного обеспечения компьютерных систем. Технологическая и эксплуатационная безопасность программ

Необходимость определения этапов жизненного цикла (ЖЦ) ПО обусловлена стремлением разработчиков к повышению качества ПО за счет оптимального управления

разработкой и использования разнообразных механизмов контроля качества на каждом этапе, начиная от постановки задачи и заканчивая авторским сопровождением ПО.

Наиболее общим представлением жизненного цикла ПО является модель в виде базовых этапов – процессов [Лип1], к которым относятся:

- системный анализ и обоснование требований к ПО;
- предварительное (эскизное) и детальное (техническое) проектирование ПО;
- разработка программных компонент, их комплексирование и отладка ПО в целом;
- испытания, опытная эксплуатация и тиражирование ПО;
- регулярная эксплуатация ПО, поддержка эксплуатации и анализ результатов;
- сопровождение ПО, его модификация и совершенствование, создание новых версий.

Данная модель является общепринятой и соответствует как отечественным нормативным документам в области разработки программного обеспечения, так и зарубежным. С точки зрения обеспечения технологической безопасности целесообразно рассмотреть более подробно особенности представления этапов ЖЦ.

Графическое представление моделей ЖЦ позволяет наглядно выделить их особенности и некоторые свойства процессов. Первоначально была создана каскадная модель ЖЦ [Лип1], в которой крупные этапы начинались друг за другом с использованием результатов предыдущих работ.

Наиболее специфической является спиралевидная модель ЖЦ [Лип1]. В этой модели внимание концентрируется на итерационном процессе начальных этапов проектирования. На этих этапах последовательно создаются концепции, спецификации требований, предварительный и детальный проект. На каждом витке уточняется содержание работ и концентрируется облик создаваемого ПО.

Для проектирования ПО сложной системы, особенно системы реального времени, целесообразно использовать общесистемную модель ЖЦ, основанную на объединении всех известных работ в рамках рассмотренных базовых процессов. Эта модель предназначена для использования при планировании, составлении рабочих графиков, управлении различными программными проектами.

Совокупность этапов данной модели ЖЦ целесообразно делить на две части, существенно различающихся особенностями процессов, технико-экономическими характеристиками и влияющими на них факторами.

В первой части ЖЦ производится системный анализ, проектирование, разработка, тестирование и испытания ПО. Номенклатура работ, их трудоемкость, длительность и другие характеристики на этих этапах существенно зависят от объекта и среды разработки. Изучение подобных зависимостей для различных классов ПО позволяет прогнозировать состав и основные характеристики графиков работ для новых версий ПО.

Этой совокупности этапов ЖЦ ПО соответствует процесс внесения в разрабатываемые программы определенных защитных функций. Этот процесс называется обеспечением технологической безопасности и характеризуется необходимостью предотвращения модификации ПО за счет внедрения РПС априорного типа (алгоритмических и программных закладок).

Вторая часть ЖЦ, отражающая поддержку эксплуатации и сопровождения ПО, относительно слабо связана с характеристиками объекта и среды разработки. Номенклатура работ на этих этапах более стабильна, а их трудоемкость и длительность могут существенно изменяться, и зависят от массовости применения ПО.

Для любой модели ЖЦ обеспечение высокого качества программных комплексов возможно лишь при использовании регламентированного технологического процесса на каждом из этих этапов. Такой процесс поддерживается CASE-средствами (средствами автоматизации разработки ПО), которые целесообразно выбирать из имеющихся или создавать с учетом объекта разработки и адекватного ему перечня работ.

Этапам эксплуатации и сопровождения ПО соответствует процесс обеспечения эксплуатационной безопасности ПО. Этот процесс характеризуется необходимостью защиты

программ от компьютерных вирусов и программных закладок апостериорного типа. Последние могут внедряться за счет злонамеренного использования методов исследования программ и их спецификаций. Кроме того, на этапе обеспечения эксплуатационной безопасности ПО применяются методы защиты программ от несанкционированного копирования, распространения и использования.

Использование при создании программного обеспечения КС сложных операционных систем, инструментальных средств разработки ПО увеличивают потенциальную возможность внедрения в программы преднамеренных дефектов диверсионного типа. Помимо этого, при создании прикладного программного обеспечения всегда необходимо исходить из возможности наличия в коллективе разработчиков программистов - злоумышленников, которые в силу тех или иных причин могут внести в разрабатываемые программы РПС.

Характерным свойством РПС в данном случае является возможность внезапного и незаметного нарушения или полного вывода из строя КС. Функционирование РПС реализуется в рамках модели угроз безопасности ПО, основные элементы которой рассматриваются в следующем разделе.

Принципы классификации систем защиты программного обеспечения (по)

Средства защиты(СЗ) ПО можно классифицировать по ряду признаков:

- По используемому механизму защиты
- По методу установки
- По принципу функционирования

Классификация систем защиты по по методу установки

Для производителей ПО удобней всего использовать защиту, устанавливаемую на скомпилированные модули. Но такая защита наименее стойка к атакам. Системы с внедрением СЗ в исходный код неудобны для производителей, так как им приходится обучать персонал работе с СЗ и ряд других неудобств. Лучшим решением является использование СЗ комбинированного типа.

Системы защиты ПО по методу установки можно подразделить:

- на системы, устанавливаемые на скомпилированные модули ПО;
- системы, встраиваемые в исходный код ПО до компиляции;
- комбинированные.

Системы первого типа наиболее удобны для производителя ПО, так как легко можно защитить уже полностью готовое и оттестированное ПО (обычно процесс установки защиты максимально автоматизирован и сводится к указанию имени защищаемого файла и нажатию "Enter"), а потому и наиболее популярны. В то же время стойкость этих систем достаточно низка, так как для обхода защиты достаточно определить точку завершения работы "конверта" защиты и передачи управления защищенной программе, а затем принудительно ее сохранить в незащищенном виде.

Системы второго типа неудобны для производителя ПО, так как возникает необходимость обучать персонал работе с программным интерфейсом (API) системы защиты с вытекающими отсюда денежными и временными затратами. Но такие системы являются более стойкими к атакам, потому что здесь исчезает четкая граница между системой защиты и как таковым ПО.

Наиболее живучими являются комбинированные системы защиты. Сохраняя достоинства и недостатки систем второго типа, они максимально затрудняют анализ и дезактивацию своих алгоритмов.

Классификация систем защиты ПО по используемым механизмам защиты

Возможно два подхода решения проблемы защиты ПО. Один – это решение проблемы с помощью технических средств защиты. Включать СЗ в состав ПО. Другой – использование юридической защиты. ПО не содержит технические СЗ, а сопровождается информацией об управлении правами.

По используемым механизмам защиты СЗ можно классифицировать на:

- системы, использующие сложные логические механизмы;
- системы, использующие шифрование защищаемого ПО - комбинированные системы.

Системы первого типа используют различные методы и приёмы, ориентированные на затруднение дизассемблирования, отладки и анализа алгоритма СЗ и защищаемого ПО. Этот тип СЗ наименее стоек к атакам, так как для преодоления защиты достаточно проанализировать логику процедур проверки и должным образом их модифицировать.

Более стойкими являются системы второго типа. Для дезактивации таких защит необходимо определение ключа дешифрации ПО. Самыми стойкими к атакам являются комбинированные системы.

Классификация систем защиты по по принципу функционирования

По принципу функционирования СЗ можно подразделить на:

- упаковщики/шифраторы;
- СЗ от несанкционированного копирования
- СЗ от несанкционированного доступа (НСД).

Назначение упаковщиков-шифраторов

Их цель защита ПО от анализа его алгоритмов и несанкционированной модификации. Для достижения этого используются алгоритмы компрессии данных; шифрование данных, алгоритмы мутации, запутывание логики программы, приведение ОС в нестабильное состояние на время работы ПО и др.

Системы защиты от несанкционированного копирования

СЗ от несанкционированного копирования осуществляют "привязку" ПО к дистрибутивному носителю (гибкий диск, CD ...). Данный тип защит основывается на изучении работы контроллеров накопителей, их физических показателей, нестандартных режимах разбивки, чтения/записи и т.п.

Системы защиты от копирования можно разделить на следующие группы:

- привязка к дискете;
- привязка к компьютеру;
- привязка к ключу;
- опрос справочников
- ограничение использования ПО.

Системы защиты от несанкционированного доступа

В защите информации ПК от НСД можно выделить три основных направления:

- специальные технических средства опознавания пользователя;
- специальное программное обеспечение по защите информации;
- специальные средства защиты информации ПК от несанкционированного доступа.

Системы защиты информации от НСД обеспечивают выполнение следующих функций:

1. идентификация, т.е. присвоение уникальных признаков - идентификаторов, по которым в дальнейшем система производит аутентификацию
2. аутентификация, т.е. установление подлинности на основе сравнения с эталонными идентификаторами;
3. разграничение доступа пользователей к ПЭВМ;
4. разграничение доступа пользователей по операциям над ресурсами (программы, данные и т.д.);
5. администрирование: a. определение прав доступа к защищаемым ресурсам, b. обработка регистрационных журналов, c. установка системы защиты на ПЭВМ, d. снятие системы защиты с ПЭВМ;
6. регистрация событий: a. входа пользователя в систему, b. выхода пользователя из системы, c. нарушения прав доступа;
7. реакция на попытки НСД;

8. контроль целостности и работоспособности систем защиты;
9. обеспечение информационной безопасности при проведении ремонтно-профилактических работ;
10. обеспечение информационной безопасности в аварийных ситуациях. Права пользователей по доступу к программам и данным описывают таблицы, на основе которых и производится контроль и разграничение доступа к ресурсам. Доступ должен контролироваться программными средствами защиты. Если запрашиваемый доступ не соответствует имеющемуся в таблице прав доступа, то системы защиты регистрирует факт НСД и инициализирует соответствующую реакцию.

Системы привязки по к компьютеру

Системы "привязки" ПО при установке на ПК пользователя осуществляют поиск уникальных признаков компьютерной системы, либо они устанавливаются самой системой защиты. После этого модуль защиты в самом ПО настраивается на поиск и идентификацию данных признаков, по которым в дальнейшем определяется авторизованное или неавторизованное использование ПО.

При этом возможно применение методик оценки скоростных и иных показателей процессора, материнской платы, дополнительных устройств, ОС, чтение/запись в микросхемы энергонезависимой памяти, запись скрытых файлов, настройка на наиболее часто встречаемую карту использования ОЗУ и т.п.

Выделение объективных характеристик программы

Идентификация программы или отдельного модуля представляет интерес в том случае, когда другие методы защиты не приносят успеха. Широко обсуждаются проблемы авторского права для отдельной процедуры программы и взаимосвязь между идеей и способом ее реализации.

Выделение объективных характеристик программы - довольно сложная процедура, тем не менее, признаки подобия двух программ или модулей, содержащихся в больших программах, указать можно. Проблема заключается в том, чтобы уметь идентифицировать программы, которые изменены хакером, погружены в другую программу или откомпилированы в машинный код.

"Родимые пятна"

Понятие "родимые пятна" используется для описания характеристик, появляющихся в результате естественного процесса разработки программы и относящихся к особенностям стиля программирования, ошибкам и избыточностям, которые не должны иметь места в независимо написанной программе. Каждое из них может служить убедительной уликой нарушения авторского права.

Водяные знаки как метод пассивной защиты.

Использование водяных знаков как метода выявления подделки занимает особое место, поскольку препятствует созданию точной копии, которую пользователь не мог бы отличить от оригинала.

Психологические методы защиты.

Психологические методы основаны на том, чтобы создать у нарушителя чувство неуверенности и психологического напряжения, заставляя его все время помнить, что в похищенном программном продукте могут сохраняться средства защиты. Поэтому полезно было бы дать объявление, что в программное обеспечение встроены механизмы защиты (независимо от того, так ли это на самом деле). Существует огромное число хитроумных способов расстановки отличительных меток в программе и никакой хакер не может быть уверен, что ему удалось уничтожить все ключи и механизмы защиты.

Существующие системы защиты ПС (СЗ ПС) можно классифицировать по ряду признаков: методы установки, используемые механизмы защиты и принципы функционирования.

Методы установки

Системы, устанавливаемые на скомпилированные модули ПС, удобны для производителя ПС, так как позволяют легко защитить полностью готовое и оттестированное ПС. Стойкость этих систем достаточно низка, так как для обхода защиты достаточно определить точку завершения работы «конверта» защиты и передачи управления защищенной программе, а затем принудительно ее сохранить в незащищенном виде.

Системы, встраиваемые в исходный код ПС до компиляции, неудобны для производителя ПС, так как возникает необходимость обучать персонал работе с программным интерфейсом (API) системы защиты с вытекающими отсюда денежными и временными затратами.

Кроме того, усложняется процесс тестирования ПС и снижается его надежность, поскольку, кроме самого ПС, ошибки могут быть как в API системе защиты, так и в процедурах, его использующих. Но такие системы являются стойкими к атакам, потому что в них исчезает четкая граница между системой защиты и ПС как таковым.

Комбинированные системы защиты являются наиболее живучими. Сохраняя достоинства и недостатки систем второго типа, они максимально затрудняют анализ и дезактивацию своих алгоритмов.

Методы защиты

Рассмотрим основные методы:

- алгоритмы запутывания: используются хаотичные переходы в разные части кода, внедрение ложных процедур-«пустышек», холостые циклы, искажение количества реальных параметров процедур ПС, разброс участков кода по разным областям ОЗУ;
- алгоритмы мутации: создаются таблицы соответствия операндов-синонимов и замена их друг на друга при каждом запуске программы по определенной схеме или случайным образом, случайные изменения структуры программы;
- алгоритмы компрессии данных: программа упаковывается, а затем распаковывается по мере выполнения;
- алгоритмы шифрования данных: программа шифруется, а затем расшифровывается по мере выполнения;
- вычисление сложных математических выражений в процессе отработки механизма защиты: элементы логики защиты зависят от результата вычисления значения какой-либо формулы или группы формул;
- методы затруднения дизассемблирования: используются различные приемы, направленные на предотвращение дизассемблирования в пакетном режиме;
- методы затруднения отладки: используются различные приемы, направленные на усложнение отладки программы;
- эмуляция процессоров и операционных систем: создаются виртуальный процессор и/или операционная система (не обязательно реально существующие) и программа-переводчик из системы команд IBM в систему команд созданного процессора или ОС; после такого перевода ПС может выполняться только при помощи эмулятора, что резко затрудняет исследование алгоритма ПС;
- нестандартные методы работы с аппаратным обеспечением: модули системы защиты обращаются к аппаратуре ЭВМ, минуя процедуры операционной системы, и используют мало известные или недокументированные ее возможности;
- шифрование защищаемого ПС: для дезактивации защиты необходимо определение ключа дешифрации ПС;
- комбинированные методы.

Принципы функционирования

Упаковщики/шифраторы защищают ПС от анализа его алгоритмов и несанкционированной модификации. Для достижения этого используются: алгоритмы компрессии данных; приемы, связанные с использованием недокументированных особенностей операционных систем (ОС) и процессоров; шифрование данных, алгоритмы

мутации, запутывание логики программы и др. Этот вид защиты замедляет выполнение ПС, затрудняет обновление и исправление ошибок в ПС.

Защита от несанкционированного копирования обеспечивает «привязку» ПС к дистрибутивному носителю (гибкий диск, CD...). Данный тип защиты основан на глубоком изучении работы контроллеров-накопителей, их физических показателей, нестандартных режимах разбивки, чтения/записи и т.п.

При этом на физическом уровне создается дистрибутивный носитель, обладающий (предположительно) неповторимыми свойствами (обычно это достигается при помощи нестандартной разметки носителя информации или/и записи на него дополнительной информации – пароля или метки), а на программном – создается модуль, настроенный на идентификацию и аутентификацию носителя по его уникальным свойствам. При этом возможно применение приемов, используемых упаковщиками/шифраторами.