МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ (ГУ)
Факультет радиотехники и кибернетики

Эссе по курсу «Защита информации»

Базовые идеи защищенных информационных систем

Подготовил студент 312 группы
Токарев Р.С.

Долгопрудный, 2007

1. ВВЕДЕНИЕ

О чём это эссе

В данном эссе будет изложено понятие защищенной информационной системы и причины, которые могут нарушить ее правильную работу. А также будут изложены базовые идеи построения таких систем.

Обеспечение защищенных вычислений

Существует несколько причин, которые могут нарушить правильную работу системы и от которых желательно защитить её работу.

Во-первых, сбои и отказы оборудования. Защищенная система должна надежно их обнаруживать и ликвидировать их последствия на работу системы. Современное оборудование обладает достаточно высокой надежностью, поэтому в дальнейшем мы будем предполагать, что система полностью защищена от сбоев и отказов оборудования.

Современные информационные системы состоят из множества машин, рассредоточенных на отдаленном расстоянии друг от друга и объединённых в сеть посредством различных линий связи. Ненадежная работа линий связи может стать другой причиной неправильной работы системы. При этом следует бороться не только со сбоями на них, но и со злонамеренными атаками с использованием линий связи. В дальнейшем мы также будем предполагать надежную работу линий связи и сосредоточимся именно на организации защищенных вычислений.

Таким образом, мы будем предполагать надежную работу оборудования и правильную работу линий связи, защищенную от всех атак.

Для достижения полной защищенности наших вычислений нам еще необходимо сделать крайне важный, и быть может, наиболее трудный шаг – защитить их от всевозможного влияния злонамеренных атак и ошибок других вычислений на данной или другой машине, работающей в сети.

Таким образом, обеспечив надежную работу оборудования и линий связи, а также защитив вычисления от своих программных ошибок и/или злых намерений со стороны других программистов, мы можем считать, что система обеспечивает защищенность вычислений.

Что надо защищать в информационной системе

Основа любых вычислений и информационных систем – обрабатывающие устройства и памяти всех уровней.

Обрабатывающие устройства (ОУ) – устройства, выполняющие все операции и хранящие некоторый объем данных, которые непосредственно используются и/или предположительно будут использоваться в качестве аргументов операций.
Информационное пространство (ИП) – хранилище данных и исполняемого кода.

Обрабатывающие устройства, работающие по программе данных вычислений, сами по себе не требуют защиты от внешних воздействий. Проблему здесь представляют ошибки в самой программе вычислений, от которых и следует защищаться, и о которых пойдет речь в дальнейшем.

В отличие от ОУ, память является уязвимой по отношению к внешним воздействиям компонентой системы.

Тем самым, мы должны защитить память данных вычислений от внешних воздействий и побороться с ошибками в собственной программе. Обе эти проблемы в технологии, которая будет изложена ниже, решаются на достаточно фундаментальном уровне.

2. БАЗОВЫЕ ИДЕИ ЗАЩИЩЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

Общая структура

Как уже отмечалось выше, модель информационной системы можно себе представить состоящей из ОУ и ИП.

В описываемой «идеальной» в смысле защищенности системе все вычисления разбиваются на фрагменты или вычислительные модули (ВМ). Это наименьшая единица защищенности, в том смысле, что различные части одного и того же модуля уже не защищены друг от друга.

Соотношение между ИП, ОУ и ВМ можно представить следующим образом:
ВМ, так же как и обрабатываемые данные, размещается в ИП. Информационная система устроена таким образом, что ОУ работает под управлением ВМ, и в процессе работы он может обращаться к ИП, считывать из него информацию во внутренние ячейки ОУ, обрабатывать их, вычисляя новые значения и различным образом менять ИП.

Далее рассмотрим более подробно базовые операции, которые может выполнять ОУ в ИП.

Создание нового узла

В наборе операций, которые может выполнять ОУ, предусмотрена операция создания узла в ИП произвольного объема для хранения данных. В результате выполнения данной операции в ИП создается узел заданного объема, а в одной из ячеек ОУ появляется ссылка на него.

После создания новый узел не доступен никому кроме создавшего его ОУ. Доступ к новому узлу возможен только через новую ссылку. Новый узел «пуст», т.е. не содержит информации.

«Пустая» информация – это такая информация, которая вызывает диагностическое прерывание, при попытке использовать ее в вычислениях.

Во всех случаях, пустой узел не должен содержать старых ссылок, присутствие которых могло бы открыть доступ к информации, не принадлежащей текущим вычислениям, что по существу является нарушением защищенности.

Ссылка

Ссылка – объект, описывающий единичный элемент данных или их агрегат и содержащий разрешенные права доступа к нему. Она не только описывает данные, но и содержит всю необходимую информацию для фактического доступа к ним.

Обсудим разрешенные операции над ссылкой и её свойства.

Ссылка на вновь созданный узел позволяет читать и изменять информацию, находящуюся в этом узле, а так же использовать эту информацию в качестве исполняемого кода. Использования вычислительным модулем ссылки для доступа к данным, фактически означает то, что данный ВМ имеет права доступа к данным, доступным по этой ссылке. Это обстоятельство является важнейшим свойством ссылки. Оно является следствием того факта, что ссылки в ВМ могут появиться только вследствие создания данным ВМ новых узлов либо при передачи ссылки данному ВМ снаружи.

Однако в ОУ предусмотрены операции создания из ссылки новой ссылки с ограниченными правами на доступ. Тем самым, существует возможность ограничивать доступ к данным доступным по ссылке для других ВМ, в которые она была передана.

Приведем теперь важнейшее правило работы со ссылкой, выполнение которого в значительной степени ответственно за обеспечение защищенности информационной системы в целом.

Система должна обеспечивать контроль над тем, чтобы в операциях, предусматривающих использование в качестве аргумента ссылки, не были использованы данные других типов (целые, вещественные и т.д.), а в операциях с аргументами других типов ссылка не могла быть модифицирована.

Т.е. при работе со ссылками должен быть обеспечен строгий контроль типов.

Уничтожение узла

Уничтожение узла может происходить либо самой системой, если на какой-либо узел не осталось ссылок, либо внутри самой программы. Основным требованием к уничтожению узла в обоих случаях является следующий момент: обращение по любой зависшей ссылки на уничтоженный узел должно вызывать диагностическое прерывание.

Контекст

Контекст – множество данных, доступных для вычислений по программе текущего ВМ.

Контекстная ссылка (КС) – единая ссылка, через которую можно получить доступ ко всей совокупности данных, принадлежащих ОУ во время выполнения текущего ВМ.

КС – это единственная величина, которая по логическим соображениям должна храниться в ОУ.

Смена контекста

На одном и том же ОУ в различное время могут исполняться различные ВМ. Один ВМ, может запускать другой для выполнения операций, необходимых согласно алгоритму. Это обычный запуск процедур. Также работа текущего ВМ может быть прервана системой и система переключиться на обработку прерывания. Затем произойдет возврат в ВМ, работа которого была прервана.

Каждый такой процесс влечет за собой смену КС в ОУ. Рассмотрим этот процесс более подробно.

При запуске нового ВМ с самого начала, можно говорить о создании нового контекста и соответствующей КС. Этот новый контекст состоит из глобальных данных, передаваемых параметров, локальных данных и средств общения с ОС.

Идентификация контекста

Если контекст А собирается переключиться на контекст В, то в первом должна, по крайней мере присутствовать идентификация контекста В. В общем случае надо рассматривать не пересекающиеся контексты. Исходя из этого факта, в первом контексте не допускается присутствие ссылки на данные второго контекста, так как в этом случае контекст В был бы частью контекста А.

Здесь возможны различные решения. Может быть введен новый тип данных, содержащий ссылку на новый контекст, но не позволяющий работать с ним как со ссылкой. Его можно использовать только для переключения контекстов. Возможны и другие варианты.

Переключение контекста

Переключение контекста должно происходить таким образом, чтобы сразу после переключения была известна следующая инструкция исполняемого кода в новом коде, так как после переключения контекста мы уже не имеем доступа к исполняемому коду предыдущего ВМ, так как противное нарушило бы все принципы защищенности.

Переключение номера команды

Как и в случае идентификации контекста, в общем случае контекст А может и не содержать ссылку на код их контекста В.

В этом случае так же необходим новый тип данных, который был бы подобием ссылки на код в контексте В. Но эту «ссылку» можно использовать только для переключения.

Реально, можно иметь один новый тип, содержащий как идентификацию контекста, так и номер команды для переключения. Будем называть этот тип МЕТКОЙ ВЫЧИСЛИТЕЛЬНОГО МОДУЛЯ (МВМ). Именно она может быть полезной в качестве параметра переключения контекста и кода программы.

Формирование переменных нового типа

Очевидно, что сформировать описанные выше переменные или МВМ можно только в контексте, в который происходит переключение (контекст В). По существу это преобразование переменной типа ссылка в новый тип.

В результате последовательность действий выглядит следующим образом:

1. В контексте B (или в контексте, включающим в себя глобальные данные контекста B и код) создаются переменные нового типа необходимые для переключения в контекст B.
2. Эти переменные пересылаются или делаются доступными каким-либо другим способом в других контекстах (например, в A).
3. В нужный момент по программе контекста A происходит переключение в контекст B.

Литература

проф. Борис Бабаян «Защищенные информационные системы» (2003г.)