А.П. Зайцев, И.В. Голубятников
Р.В. Мещеряков, А.А. Шелупанов

ПРОГРАММНО-АППАРАТНЫЕ СРЕДСТВА ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Учебное пособие

А.П. Зайцев, И.В. Голубятников,
Р.В. Мещеряков, А.А. Шелупанов

ПРОГРАММНО-АППАРАТНЫЕ СРЕДСТВА ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Учебное пособие

2006

Корректор: Лопатин В.Д.

Зайцев А.П., Голубятников И.В.,
Мещеряков Р.В., Шелупанов А.А.
Программно-аппаратные средства обеспечения информационной безопасности: Учебное пособие. Издание 2-е испр. и доп.– М.:Машиностроение-1, 2006. – 260 с.

ISBN 5-94275-345-7

В пособии изложены основные принципы организации защиты в компьютерных системах обработки конфиденциальной информации, приведено описание конкретных аппаратно-программных систем защиты информации.

ISBN 5-94275-345-7

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

1 Необходимость и цели защиты информации

Бурное развитие средств вычислительной техники, автоматизированных информационных систем, появление новых информационных технологий в нашей стране сопровождается, к сожалению, и появлением таких малоприятных явлений, как промышленный шпионаж, компьютерная преступность и, прежде всего, несанкционированный доступ (НСД) к конфиденциальной информации. Этим обуславливается актуальность и значимость проблемы защиты информации.

Острая необходимость в защите информации нашла выражение в создании Государственной системы защиты информации (ГСЗИ). Развивается правовая база информационной безопасности. Приняты и введены в действие законы «О государственной тайне», «Об информации, информатизации и защите информации», «О правовой охране программ для электронных вычислительных машин и баз данных» и др. Целями защиты информации являются: предотвращение ущерба, возникновение которого возможно в результате утери (хищения, утраты, искажения, подделки) информации в любом ее проявлении; реализация адекватных угрозам безопасности информации мер защиты в соответствии с действующими Законами и нормативными документами по безопасности информации (приложение 1), потребностями владельцев (пользователей) информации. «Защите подлежит любая документированная информация, неправомерное обращение с которой может нанести ущерб ее собственнику, владельцу, пользователю и иному лицу» [2].

Любое современное предприятие (учреждение, фирма и т.д.), независимо от вида деятельности и форм собственности, не может сегодня успешно развиваться и вести хозяйственную и иную деятельность без создания надежной системы защиты своей информации, включающей не только организационно-нормативные меры, но и технические средства контроля безопасности информации при ее обработке, хранении и передаче в автоматизированных системах (АС), прежде всего, программно-аппаратные.

2 Основные принципы организации защиты информации от НСД и обеспечения ее конфиденциальности

Мероприятия по защите информации от НСД являются составной частью управленческой, научной, производственной (коммерческой) деятельности предприятия (учреждения, фирмы и т.д.), независимо от их ведомственной принадлежности и формы собственности, и осуществляются в комплексе с другими мерами по обеспечению установленного режима конфиденциальности. Практика организации защиты информации от НСД при ее обработке и хранении в АС должна учитывать следующие принципы и правила обеспечения безопасности информации [3]:

1. Соответствие уровня безопасности информации законодательным положениям и нормативным требованиям по охране сведений, подлежащих защите по действующему законодательству, в т.ч. выбор класса защищенности АС в соответствии с особенностями обработки информации (технология обработки, конкретные условия эксплуатации АС) и уровнем ее конфиденциальности.

2. Выявление конфиденциальной информации и документальное оформление в виде перечня сведений, подлежащих защите, его своевременная корректировка.

3. Наиболее важные решения по защите информации должны приниматься руководством предприятия (организации, фирмы), владельцем АС.

4. Определение порядка установки уровней полномочий субъектов доступа, а также круга лиц, которым это право предоставлено.

5. Установление и оформление правил разграничения доступа (ПРД), т.е. совокупности правил, регламентирующих права доступа субъектов доступа к объектам доступа.

6. Установление личной ответственности пользователей за поддержание уровня защищенности АС при обработке сведений, подлежащих защите по действующему законодательству, путем:

- ознакомления с перечнем защищаемых сведений, организационно-распорядительной и рабочей документацией, опреде-

ляющей требования и порядок обработки конфиденциальной информации;

- определение уровня полномочий пользователя в соответствии с его должностными обязанностями;

- получения от субъекта доступа расписки о неразглашении доверенной ему конфиденциальной информации.

7. Обеспечение физической охраны объекта, на котором расположена защищаемая АС (территория, здания, помещения, хранилища информационных носителей), путем установления соответствующих постов, технических средств охраны или любыми другими способами, предотвращающими или существенно затрудняющими хищение средств вычислительной техники (СВТ), информационных носителей, а также НСД к СВТ и линиям связи.

8. Организация службы безопасности информации (ответственные лица, администратор АС), осуществляющей учет, хранение и выдачу информационных носителей, паролей, ключей, ведение служебной информации СЗИ НСД (генерацию паролей, ключей, сопровождение правил разграничения доступа), приемку включаемых в АС новых программных средств, а также контроль за ходом технологического процесса обработки конфиденциальной информации и т.д.

9. Планомерный и оперативный контроль уровня безопасности защищаемой информации согласно НД по безопасности информации, в т.ч. проверка защитных функций средств защиты информации.

Средства защиты информации должны иметь сертификат, удостоверяющий их соответствие требованиям по безопасности информации.

Современные компьютеры за последние годы приобрели гигантскую вычислительную мощь, но одновременно с этим стали гораздо проще в эксплуатации. Пользоваться ими стало легче, поэтому все большее количество новых, как правило, малоквалифицированных людей получают доступ к компьютерам, что существенно облегчает задачу нарушителям, т.к. в результате «персонализации» средств ВТ большинство пользователей осуществляет администрирование компьютеров самостоятельно.

Повсеместное распространение сетевых технологий объединило отдельные машины в локальные сети, совместно использующие общие ресурсы, а применение технологий клиент-сервер и кластеризации преобразовало такие сети в распределенные вычислительные среды. Безопасность сети определяется защищенностью всех входящих в нее компьютеров и сетевого оборудования, и злоумышленнику достаточно нарушить работу только одного компонента, чтобы скомпрометировать всю сеть.

Современные телекоммуникационные технологии объединили локальные компьютерные сети в глобальную информационную среду – Internet. Именно развитие Internet вызвало всплеск интереса к проблеме безопасности и поставило вопрос об обязательном наличии средств защиты у сетей и систем, подключенных к Internet, независимо от характера обрабатываемой в них информации. Дело в том, что Internet обеспечивает возможности злоумышленникам для осуществления нарушений безопасности в глобальном масштабе. Если компьютер, который является объектом атаки, подключен к Internet, то для атакующего не имеет большого значения, где он находится – в соседней комнате или на другом континенте.

По заключению экспертов самым привлекательным сектором российской экономики для преступников является кредитно-финансовая система. Анализ преступных действий, совершенных в этой сфере, и опросы представителей банковских учреждений позволяют выделить наиболее типичные способы совершения преступлений:

- Наиболее распространенные компьютерные преступления, совершенные путем несанкционированного доступа к банковским базам данных посредством телекоммуникационных сетей.
- За последнее время не отмечено ни одно компьютерное преступление, которое было бы совершено одним человеком. Известны случаи, когда преступными группировками нанимались бригады из десятков хакеров.
- Большинство компьютерных преступлений в банковской сфере совершается при непосредственном участии самих служащих коммерческих банков.

➢ Много компьютерных преступлений совершается в России с использованием возможностей, которые предоставляет своим пользователям сеть Internet.

Уникальность сети Internet заключается в том, что она не находится во владении какого-то физического лица, частной компании, государственного ведомства или отдельной страны. Поэтому практически во всех ее сегментах отсутствует централизованное регулирование, цензура и другие методы контроля информации. Благодаря этому открываются практически неограниченные возможности доступа к любой информации. Сеть Internet можно рассматриать не только как инструмент совершения компьютерных преступлений, но и как среду для ведения преступной деятельности.

При использовании сети Internet в качестве среды для преступной деятельности привлекательной для правонарушителей является возможность обмена информацией криминального характера.

Другая особенность сети Internet – возможность осуществлять в глобальных масштабах информационно-психологическое воздействие на людей. Преступное сообщество весьма заинтересовано в распространении своих доктрин и учений для укрепления позиций представителей преступного мира.

Однако наибольший интерес сеть Internet представляет именно как орудие для совершения преступлений обычно в сфере экономики и финансов. В самом простом варианте эти преступления связаны с нарушением авторских прав (незаконное копирование и продажа программ, находящихся на серверах компаний).

Во вторую группу преступлений можно включить нелегальное получение товаров и услуг, в частности, бесплатное пользование услугами, представляемыми за плату различными телефонными компаниями.

Другие способы незаконного пользования услугами основываются на модификации сведений о предоставлении этих услуг в базах данных компаний, которые их оказывают. Информация о предоставлении какой-то услуги в кредит может либо просто уничтожаться, либо изменяться для того, чтобы потребителем уже оплаченной кем-то услуги стал член преступного сообщества.

Через сеть Internet преступники стремятся также получить возможность нужным для себя образом модифицировать конфиденциальную служебную информацию, которая используется руководством банка для принятия каких-либо важных решений.

Дополнительная сфера компьютерных преступлений, совершаемых через сеть Internet, появилась с возникновением электронных банковских расчетов, т.е. с введением в обращение так называемой электронной наличности.

Способы хищения основываются на модификации информации, отображающей электронную наличность. Информация переписывается на счета преступников.

Защищенная система обработоки информации для определенных условий эксплуатации обеспечивает безопасность (конфиденциальность и целостность) обрабатываемой информации и поддерживает свою работоспособность в условиях воздействия на нее множества угроз.

Защищенная система обработки информации должна обладать следующими свойствами:

- Осуществление автоматизации некоторого процесса обработки конфиденциальной информации, включая все аспекты этого процесса, связанные с обеспечением безопасности информации.
- Успешное противостояние угрозам безопасности, действующими в определенной среде.
- Соответствие требованиям и критериям стандартов информационной безопасности.

Защита информации в КС – комплекс организационных, организационно-технических и технических мер, предотвращающих или снижающих возможность образования каналов утечки информации и/или каналов воздействия на КС.

Для защиты информации в компьютерной системе принимаются следующие меры.

Организационные меры защиты – меры общего характера, затрудняющие доступ к ценной информации злоумышленникам вне зависимости от особенностей способа обработки информации и каналов утечки и воздействия.

Организационно-технические меры защиты – меры, связанные со спецификой каналов утечки (воздействия) и метода

обработки информации, но не требующие для своей реализации нестандартных приемов, оборудования или программных средств.

Технические (программно-технические) меры защиты – меры, жестко связанные с особенностями каналов утечки и воздействия и требующие для своей реализации специальных приемов, оборудования или программных средств.

Глава 1. МЕТОДЫ ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ КОМПЬЮТЕРНЫХ СИСТЕМ

1.1 Методы создания безопасных систем обработки информации

В данном разделе рассмотрим самые общие методы обеспечения информационной безопасности автоматизированных систем с позиций автоматизации процессов обработки конфиденциальной информации и методы противодействия угрозам безопасности [1]. Подробно эти вопросы рассматриваются в стандартах информационной безопасности.

1.2 Автоматизация процесса обработки конфиденциальной информации

До применения компьютерных технологий в любой организации, для которой безопасность информации имела определенное значение, был установлен определенный порядок работы с информацией (например, система работы с секретными документами), регламентирующий информационные потоки внутри организации и обмен информацией с внешним миром. Этот порядок включал схему информационных потоков внутри организации и правила управления этими потоками. Таким образом, если основная цель внедрения информационных технологий – автоматизировать процесс обработки информации, то частная задача автоматизации – обеспечить адекватную реализацию в компьютерной системе схемы информационных потоков и правил управления ими, существовавших до применения компьютерных средств.

1.3 Противодействие угрозам безопасности путем устранения предпосылок их осуществления

Это основная задача защиты. Любая успешная атака непременно использует определенные особенности системы обработки информации или недостатки средств защиты. В большинстве случаев наличие «изъянов защиты» определяется особенностями архитектуры и реализации средств защиты.

Создание средств защиты от каждого вида угроз не зависит напрямую от назначения системы и не требует модификации по мере ее развития. Недостаток: анализ всех типов угроз, количество которых растет.

Устранение причин, обуславливающих успешную реализацию угроз, не зависит от развития угроз, так как ликвидирует причину, а не следствие.

Недостаток: необходимость модернизации защищенных систем.

1.4 Стандарты информационной безопасности и их роль

Главная задача стандартов информационной безопасности – создать основу для взаимодействия между производителями, потребителями и экспертами по квалификации продуктов информационных технологий. Каждая из этих групп имеет свои интересы и свои взгляды на проблему информационной безопасности.

Потребители, во-первых, заинтересованы в методике, позволяющей обоснованно выбрать продукт, отвечающий их нуждам, для чего необходима шкала оценки безопасности, и, во-вторых, нуждаются в инструменте, с помощью которого они могли бы сформулировать свои требования производителям. При этом потребителя интересуют исключительно характеристики и свойства конечного продукта, а не методы и средства их достижения.

Производители также нуждаются в стандартах как средстве сравнения возможностей своих продуктов, а также в стандартизации определенного набора требований безопасности. Эти требования не должны противоречить парадигмам обработки информации, архитектуре вычислительных систем и технологиям создания информационных продуктов.

Эксперты по квалификации и специалисты по сертификации рассматривают стандарты как инструмент, позволяющий им оценить уровень безопасности.

Производители в результате квалификации уровня безопасности получают объективную оценку возможностей своего продукта.

Таким образом, перед стандартами информационной безопасности стоит задача – примирить эти три точки зрения и создать эффективный механизм взаимодействия всех сторон.

Прежде чем приступить к созданию защищенной системы обработки информации, надо сначала получить четкий и недвусмысленный ответ на вопрос: что представляет собой защищенная система. Для ответа на этот вопрос и согласования всех точек зрения разрабатываются стандарты информационной безопасности. Это документы, регламентирующие основные понятия и концепции информационной безопасности на государственном или межгосударственном уровне, определяющие понятие «защищенная система» посредством стандартизации требований и критериев безопасности, образующих шкалу оценки степени защищенности вычислительной системы (ВС). В соответствии с этими документами защищенная система – это система, сответствующая тому или иному стандарту информационной безопасности (приложение 2).

1.5 Основные понятия и определения

Эти понятия составляют базовые концепции безопасности компьютерных систем.

Политика безопасности (Security Policy). Совокупность норм и правил, обеспечивающих эффективную защиту системы обработки информации от заданного множества угроз безопасности.

Модель безопасности (*Security *Model). Формальное представление политики безопасности.

Дискреционное, или произвольное управление доступом (Discretionery Access Control). Управление доступом, осуществляемое на основании заданного администратором множества разрешенных отношений доступа (например в виде «троек» – <объект, субъект, тип доступа>).

Мандатное, или нормативное, управление доступом (Mandatory Access Control). Управление доступом, основанное на совокупности правил предоставления доступа, определенных на множестве атрибутов безопасности субъектов и объектов, например, в зависимости от грифа секретности информации и уровня доступа пользователя.

Ядро безопасности (Trusted Computing Base (TCB)). Совокупность аппаратных, программных и специальных компонент ВС, реализующих функции защиты и обеспечения безопасности.

Идентификация (Identification). Процесс распознания сущностей путем присвоения им уникальных меток (идентификаторов).

Аутентификация (Autentification). Проверка подлинности идентификаторов сущностей с помощью различных (преимущественно криптографических) методов.

Адекватность (Assurance). Показатель реально обеспечиваемого уровня безопасности, отражающий степень эффективности и надежности реализованных средств защиты и их соответствия поставленным задачам (в основном политике безопасности).

Квалификационный анализ, квалификация уровня безопасности (Evaluation). Анализ ВС с целью определения уровня ее защищенности и соответствия требованиям безопасности на основе критериев стандарта безопасности. *Квалификация уровня безопасности* является конечным этапом технологического цикла создания защищенных систем. Непосредственно предшествует процедуре сертификации и завершается присвоением ВС того или иного класса или уровня безопасности.

Таксономия (Taxonomy). Наука о систематизации и классификации сложноорганизованных объектов и явлений, имеющих иерархическое строение. В отличие от классификации, устанавливающей связи и отношения между объектами снизу-вверх, таксономия основана на декомпозиции явлений и поэтапном уточнении свойств объектов (иерархия сверху-вниз).

Прямое взаимодействие (Trusted Path). Принцип организации информационного взаимодействия (как правило, между пользователем и системой), гарантирующий, что передаваемая информация не подвергается перехвату или искажению.

1.6 Угрозы безопасности компьютерных систем

Под угрозой безопасности вычислительной системе понимаются воздействия на систему, которые прямо или косвенно могут нанести ущерб ее безопасности.

Разработчики требований безопасности и средств защиты выделяют три вида угроз:

- угрозы нарушения конфиденциальности обрабатываемой информации;
- угрозы нарушения целостности обрабатываемой информации;
- угрозы нарушения работоспособности ВС.

Угрозы конфиденциальности направлены на разглашение секретной информации (несанкционированный доступ).

Угрозы целостности представляют собой любое искажение или изменение неуполномоченным на это действие лицом хранящейся в вычислительной системе или передаваемой информации. Наиболее актуальна эта угроза для систем передачи информации – компьютерных сетей и систем телекоммуникаций.

Угрозы нарушения работоспособности (отказ в обслуживании) направлены на создание ситуаций, когда в результате преднамеренных действий ресурсы вычислительной системы становятся недоступными или снижается ее работоспособность.

Цель защиты систем обработки информации – противодействие угрозам безопасности.

1.7 Методы взлома компьютерных систем

В общем случае программное обеспечение универсальной компьютерной системы (КС) состоит из трех основных компонентов: операционной системы (ОС), сетевого программного обеспечения (СПО) и системы управления базами данных (СУБД). Поэтому попытки взлома защиты компьютерных систем можно разделить на три группы [4]:

- атаки на уровне операционной системы;
- атаки на уровне сетевого программного обеспечения;
- атаки на уровне систем управления базами данных.

Атаки на уровне СУБД. Защита СУБД является одной из самых простых задач. Это связано с тем, что СУБД имеет строго определенную внутреннюю структуру, а операции над элементами СУБД заданы довольно четко. Есть четыре основных действия – поиск, вставка, удаление и замена элемента. Другие операции являются вспомогательными и используются довольно редко. Все

это упрощает решение защиты СУБД. В большинстве случаев злоумышленники предпочитают взламывать защиту компьютерной системы на уровне ОС и получать доступ к файлам СУБД с помощью средств ОС. Однако в случае, если используется СУБД, не имеющая достаточно надежных защитных механизмов, или плохо протестированная версия СУБД, содержащая ошибки, или если при определении политики безопасности администратором СУБД были допущены ошибки, то становится вполне вероятным преодоление взломщиком защиты, реализуемой на уровне СУБД.

Имеются два специфических сценария атаки на СУБД, для защиты от которых требуется применить специальные методы. В первом случае результаты арифметических операций над числовыми полями СУБД округляются в меньшую сторону, разница суммируется в некоторой другой записи СУБД (например, личный счет злоумышленника), а округляемые числовые поля относятся к счетам других клиентов банка. Во втором случае злоумышленник получает доступ к полям записей СУБД, для которых доступной является только статистическая информация. Сущность атаки злоумышленника – так хитро сформулировать запрос, чтобы множество записей, для которого собирается статистика, состояло только из одной записи.

Атаки на уровне операционной системы. Защитить операционную систему (ОС), в отличие от СУБД, намного сложнее. Это связано с тем, что внутренняя структура современных ОС чрезвычайно сложна, и поэтому соблюдение адекватной политики безопасности является очень трудной задачей. Среди неопытных пользователей бытует мнение, что самые эффективные атаки на ОС могут быть организованы только с помощью сложнейших средств. На самом деле это не совсем так. Высокая квалификация взломщика – качество не лишнее, но искусство взлома заключается в поиске слабых мест в конкретной системе защиты. При этом простейшие методы взлома оказываются ничуть не хуже самых изощренных, поскольку чем проще алгоритм атаки, тем больше вероятность его работы без ошибок.

Успех реализации того или иного алгоритма атаки взломщика в значительной степени зависит от архитектуры и конфигурации ОС. Однако есть атаки, которым может быть подвергнута практически любая ОС:

➢ кража пароля:

- подглядывание за пользователем, когда тот вводит пароль (злоумышленник может легко узнать пароль, просто следя за перемещением пальцев при его наборе);
- получение пароля из файла, в котором этот пароль был сохранен в незашифрованном виде пользователем, не желающим затруднять себя вводом пароля при подключении к сети;
- поиск пароля, который пользователи записывают где-угодно;
- кража внешнего носителя парольной информации (дискеты или электронного ключа, где хранится пароль для входа в ОС);
- полный перебор всех возможных вариантов пароля;
- подбор пароля по частоте встречаемости символов с помощью словарей наиболее часто употребляемых паролей с применением знаний о конкретном пользователе (фамилия, номер телефона, год рождения и т.п.);

➢ сканирование жестких дисков компьютера (злоумышленник последовательно пытается обратиться к каждому файлу, хранимому на жестких дисках компьютерной системы; если объем дискового пространства достаточно большой, то вполне вероятно, что при описании доступа к файлам и каталогам администратор допустил хотя бы одну ошибку, в результате которой все каталоги и файлы будут прочитаны злоумышленником; для сокрытия следов взломщик может организовать эту атаку под чужим именем, например, под именем пользователя, пароль которого ему известен);

➢ сборка «мусора» (если средства ОС позволяют восстанавливать ранее удаленные объекты, например, из корзин);

➢ превышение полномочий (используя ошибки в программном обеспечении или в администрировании ОС, злоумышленник получает полномочия, превышающие полномочия, предоставленные ему согласно действующей политике безопасности):

- запуск программы от имени пользователя, имеющего необходимые полномочия;

- подмена динамически загружаемой библиотеки, используемой системными программами, или изменение переменных среды, описывающих путь к таким библиотекам;
- модификация кода или данных подсистемы защиты самой операционной системы;

➢ отказ в обслуживании (целью этой атаки является частичный или полный вывод из строя операционной системы):

- захват ресурсов (программа злоумышленника производит захват всех имеющихся в ОС ресурсов, а затем входит в бесконечный цикл);
- бомбардировка запросами (программа злоумышленника постоянно направляет ОС запросы, реакция на которые требует привлечения значительных ресурсов компьютера);
- использование ошибок в программном обеспечении или администрировании.

Если в программном обеспечении компьютерной системы нет ошибок и ее администратор строго соблюдает политику безопасности, то атаки всех перечисленных типов окакжутся малоэффективными. Тем не менее полностью устранить угрозу взлома компьютерной системы на уровне ОС невозможно. Поэтому политика безопасности должна быть такой, чтобы даже при преодолении защиты ОС злоумышленник не смог нанести серьезного ущерба.

Атаки на уровне сетевого программного обеспечения. Сетевое программное обеспечение (СПО) является наиболее уязвимым, потому что канал связи, по которому передаются сообщения, чаще всего не защищен, и всякий, кто может иметь доступ к этому каналу, может перехватывать сообщения и отправлять свои собственные. Поэтому на уровне СПО возможны следующие атаки взломщиков:

➢ прослушивание сегмента локальной сети (в пределах одного и того же сегмента локальной сети любой подключенный к нему компьютер в состоянии принимать сообщения, адресованные другим компьютерам сегмента, а следовательно, компьютеру взломщика, подключенному к этому же сегменту сети, становится доступен весь информационный обмен между компьютерами этого сегмента);

➢ перехват сообщений на маршрутизаторе (если взломщик имеет привилегированный доступ к сетевому маршрутизатору, то он получает возможность перехватывать все сообщения, проходящие через этот маршрутизатор, и хотя тотальный перехват невозможен из-за слишком большого объема информации, значительный интерес для взломщика представляет выборочный перехват сообщений, содержащих пароли пользователей и их электронную почту);

➢ создание ложного маршрутизатора (путем отправки в сеть сообщений специального вида взломщик добивается, чтобы его компьютер стал маршрутизатором сети, после чего получает доступ ко всем проходящим через него сообщениям);

➢ навязывание сообщений (отправляя в сеть сообщения с ложным обратным сетевым адресом, взломщик переключает на свой компьютер уже установленные сетевые соединения и в результате получает права пользователей, чьи соединения обманным путем были переключены на компьютер взломщика);

➢ отказ в обслуживании (взломщик отправляет в сеть сообщения специального вида, после чего одна или несколько компьютерных систем, подключенных к сети, частично или полностью выходят из строя).

Поскольку атаки злоумышленников на уровне СПО спровоцированы открытостью сетевых соединений, то для отражения этих атак необходимо максимально защитить каналы связи и тем самым затруднить обмен информацией по сети для тех, кто не является легальным пользователем. Ниже перечислены некоторые способы такой защиты:

- максимальное ограничение размеров компьютерной сети (чем больше сеть, тем труднее ее защитить);
- изоляция сети от внешнего мира (ограничение физического доступа к сети извне);
- шифрование сетевых сообщений (снижается угроза перехвата, но также снижается и производительность сети);
- электронная цифровая подпись сетевых сообщений (если неподписанные сообщения игнорируются, то можно забыть про угрозу навязывания сообщений и про большинство угроз отказа в обслуживании);

- использование брандмауэров (межсетевых экранов), которые являются вспомогательным средством защиты, применяемым только в том случае, если компьютерную сеть нельзя изолировать от других сетей.

1.8 Защита компьютерной системы от взлома

Для обобщенной модели взлома компьютерных систем можно сформулировать универсальные рекомендации, чтобы свести риск к минимуму:

- постоянно повышайте квалификацию в области защиты компьютерных систем;
- руководствуйтесь принципом разумной достаточности: не стремитесь построить абсолютно надежную защиту (чем мощнее защита, тем больше ресурсов компьютерной системы она требует);
- храните в секрете информацию о принципах действия защитных механизмов КС;
- постарайтесь максимально уменьшить размеры КС и без крайней необходимости не подключайте ее к Internet;
- перед покупкой нового ПО поищите информацию о нем на хакерских сайтах Internet;
- размещайте серверы в охраняемых помещениях, не подключайте к ним клавиатуру и дисплеи, чтобы доступ к ним осуществлялся только через сеть;
- абсолютно все сообщения, передаваемые по незащищенным каналам связи, должны шифроваться и снабжаться цифровой подписью;
- при стыковке защищенной сети с незащищенной все сообщения должны проходить через межсетевые экраны, шифроваться и снабжаться цифровой подписью;
- не пренебрегайте возможностями аудита (интервал просмотра журнала аудита не должен превышать одних суток);
- если окажется, что число событий в журнале аудита велико, изучите новые записи, так как не исключено, что КС подверглась атаке взломщика;

- регулярно проводите проверку целостности программного обеспечения КС, проверяйте КС на наличие в ней программных закладок;
- регистрируйте все изменения в политике безопасности в обычном бумажном журнале (регулярная проверка поможет обнаружить присутствие программной закладки);
- пользуйтесь защищенными ОС;
- создайте несколько ловушек для взломщиков (например, заведите на диске файл с привлекательным именем, и если будет зафиксировано успешное обращение к этому файлу, значит в защищаемую КС была внедрена программная закладка);
- регулярно тестируйте КС с помощью специальных программ.

1.9 Защита КС от программных закладок

1.9.1 Программные закладки

Основным условием правильного функционирования компьютерной системы является ее защита от воздействия тех программ, присутствие которых в этой системе излишне. В первую очередь такими программами являются компьютерные вирусы, от которых необходимо освобождаться и защищаться известными методами. Имеются также вредоносные программы другого типа – так называемые программные закладки, которые могут выполнять одно из перечисленных действий [4]:

- искажать коды программ, загруженных в оперативную память компьютера (программная закладка первого типа);
- перемещать фрагменты информации из одних областей оперативной или внешней памяти компьютера в другие (программная закладка второго типа);
- искажать выводимую на внешние устройства или в канал связи информацию (программная закладка третьего типа).

По методу внедрения в КС программные закладки подразделяются на:

- программно-аппаратные закладки, связанные с аппаратными средствами компьютера (их средой обитания обычно является BIOS);
- загрузочные закладки, связанные с программами начальной загрузки, которые располагаются в загрузочных секторах, из которых компьютер при начальной загрузке считыват программу, берущую на себя последующую загрузку самой операционной системы;
- драйверные закладки;
- прикладные закладки, ассоциированные с прикладным программным обеспечением общего назначения (текстовые редакторы, утилиты, антивирусные мониторы и программные оболочки);
- исполняемые закладки, ассоциированные с исполняемыми программными модулями, содержащими код этой закладки (чаще всего эти модули представляют собой пакетные файлы, которые состоят из команд операционной системы, выполняемых одна за одной, как если бы их набирали на клавиатуре компьютера);
- закладки-имитаторы, интерфейс которых идентичен с интерфейсом некоторых служебных программ, требующих ввод конфиденциальной информации (паролей, криптографических ключей, номеров кредитных карточек);
- замаскированные закладки, которые маскируются под программные средства оптимизации работы компьютера (файловые архиваторы, дисковые дефрагментаторы) или под программы игрового и развлекательного назначения.

Чтобы программная закладка выполнила свое назначение, процессор должен приступить к выполнению команд, входящих в код закладки. Это осуществимо, если программная закладка находится в оперативной памяти компьютера (программная закладка первого типа должна быть загружена до начала работы основной программы) и имели место активизирующие условия.

Иногда сам пользователь может спровоцировать запуск исполняемого файла, содержащего код программной закладки. Известен случай, когда среди пользователей свободно распространялся набор из архивированных файлов. Для извлечения файлов

из него требовалось вызвать специальную утилиту, которая запускается после указания ее имени в командной строке. Однако мало кто замечал, что в полученном наборе файлов уже имелась программа с таким же именем и что запускалась именно она. Кроме разархивирования файлов, эта программная закладка выполняла ряд негативных действий.

Можно выделить *резидентные* закладки, которые находятся в оперативной памяти постоянно, начиная с некоторого момента и до окончания работы компьютера, и *нерезидентные* закладки, которые попадают в оперативную память компьютера аналогично резидентным, но выгружаются при выполнении особых условий.

Существует три вида негативных действий программных закладок:

- копирование информации пользователя компьютерной системы (паролей, криптографических ключей, кодов доступа и т.п.);
- изменение алгоритмов работы системных, прикладных и служебных программ (например, изменение программы разграничения доступа может привести к тому, что доступ будет разрешен любому пользователю независимо от правильности пароля);
- навязывание определенных режимов работы (например, блокирование записи на диск при удалении информации, при этом запись не уничтожается и может быть прочитана).

У всех программных закладок имеется одно общее свойство: они обязательно выполняют операцию записи в оперативную или внешнюю память системы. При отсутствии этой операции никакого негативного воздействия программная закладка сделать не может. Понятно, что для избирательного действия она должна выполнять и операцию чтения.

1.9.2 Воздействия программных закладок на компьютеры

Перехват. В модели *перехват* программная закладка внедряется в ПЗУ, системное или программное обеспечение и сохраняет вводимую информацию с внешних устройств или выводи-

мую на эти устройства в скрытой области памяти локальной или удаленной компьютерной системы.

Модель может действовать в два этапа. На первом этапе перехватываются, например, только имена или начала файлов. После анализа этой информации выбираются конкретные объекты для следующей атаки.

Эта модель может быть успешно использована для атаки на защищенную операционную систему Windows NT. Как известно, после старта Windows NT на экране монитора появляется приглашение активизировать клавиши <Ctrl>+<Alt>+<Del>, после нажатия которых загружается динамическая библиотека MSGINA.dll, осуществляющая процедуры идентификации и аутентификации пароля. Существует простой способ подмены исходной библиотеки MSGINA.dll на пользовательскую добавлением специальной строки в реестр операционной системы и указанием пути к пользовательской библиотекие. В итоге злоумышленник может изменить процедуру контроля доступа к компьютерной системе.

Искажение. В этой модели воздействия программная закладка модифицирует информацию в памяти компьютерной системы или создает ошибочные ситуации.

Различают *статическое* и *динамическое* искажение. Статическое искажение происходит только один раз, после чего модифицированные параметры программной среды позволяют злоумышленнику выполнять необходимые для него действия. Например, измененный файл AUTOEXEC.bat в операционной системе Windows 95/98 может запустить программную закладку раньше, чем загрузятся программы, указанные в этом файле.

Известны случаи статического искажения, связанного с проверкой корректности цифровой подписи. В исполняемом EXE-модуле проверки цифровой подписи символьная строка «Подпись некорректна» была заменена на строку «Подпись корректна», в результате чего документы с некорректной цифровой подписью перестали фиксироваться, а в электронные документы стало возможным вносить изменения.

Суть динамичесого искажения заключается в модификации каких-либо параметров программной среды системных или при-

кладных процессов при помощи заранее активизированной программной закладки. Различают искажение на входе, когда на обработку попадает уже искаженный документ, и искажение на выходе, когда искажается информация, необходимая для работы других программ или анализа.

Программная реализация цифровой подписи – наиболее привлекательный объект для воздействия программных закладок типа «динамическое искажение». Например, в одной из реализаций криптосистемы PGP электронный документ, подлежащий удостоверению цифровой подписью, считывался блоками по 512 байт. Процесс считывания считался законченным, если в последнем блоке данные занимали менее 512 байт. Действие одной программной закладки основывалось на навязывании длины файла. Закладка позволяла считывать только первые 512 байт исходного документа, в результате чего цифровая подпись ставилась после первого блока данных. Оставшаяся часть этого документа могла быть произвольно искажена, а цифровая подпись под ним оставалась «корректной».

На цифровую подпись программные закладки могут действовать следующим образом [4]:

- искажение входной информации, когда изменяется поступающий на подпись электронный документ;
- искажение результата проверки истинности цифровой подписи, когда независимо от работы программы подпись объявляется подлинной;
- навязывание длины электронного документа;
- искажение программы цифровой подписи, когда вносятся изменения в исполняемый код программы.

В рамках модели «искажение» также создаются программные закладки, которые инициируют или подавляют сигнал о возникновении в компьютерной системе ошибочных ситуаций, которые приводят к ненормальному завершению исполняемой программы.

Для инициирования статической ошибки на устройствах хранения информации злоумышленником создается область, при обращении к которой (чтение, запись, форматировние и т.п.) возникает ошибка, что может блокировать действия системных или

прикладных программ (например, не позволять корректно уничтожать конфиденциальную информацию на жестком диске).

Для инициирования динамической ошибки для некоторой операции генерируется ложная ошибка из числа тех, которые могут возникать при выполнении данной операции. Например, для блокирования приема или передачи информации может постоянно инициироваться ошибочная ситуация «Модем занят». Или при прочтении первого блока информации длиной 512 байт может устанавливаться флажок для того, чтобы не допустить прочтения второго и последующего блоков и подделать цифровую подпись под документом.

Чтобы маскировать ошибочные ситуации, злоумышленники используют подавление сигналов статических и динамических ошибок. Целью таких действий является желание блокировать нормальное функционирование компьютерной системы.

Разновидностью программных закладок являются программы типа *троянский конь*. Троянской программой называется:

- программа, которая является частью другой программы с известными пользователю свойствами, способная втайне от него выполнять некоторые дополнительные функции с целью причинения ущерба;
- программа с известными пользователю свойствами, в которую были внесены изменения, чтобы помимо известных функций, она могла втайне от него выполнять некоторые разрушительные действия.

Такие программные закладки по некоторому активизирующему событию могут вызывать сбойную ситуацию в компьютерной системе. При этом достигаются две цели: нарушается нормальное функционирование системы и становится возможным доступ злоумышленника к компьютерной системе под предлогом устранения неисправности. При этом злоумышленник может извлечь информацию, перехваченную другими закладками. В качестве активизирующего события используется наступление определенного момента времени, состояние некоторых счетчиков (например, счетчика запусков программы) или сигнал из канала модемной связи.

В общем случае действия троянской программы могут быть любыми – от определения регистрационных номеров программного обеспечения, установленного на компьютере, до составления списка каталогов на его жестком диске. А сама троянская программа может маскироваться под текстовый редактор, под сетевую утилиту или любую программу. Троянцы написаны для всех операционных систем и представляют значительную угрозу компьютерам, поскольку их действия могут носить не только деструктивный характер, но и сбор конфиденциальной информации о системе. Обнаружить такие троянские программы удается, как правило, чисто случайно.

Уборка мусора. При хранении компьютерной информации на внешних носителях прямого доступа имеется несколько уровней иерархии: сектора, кластеры и файлы. Сектора являются единицами хранения информации на аппаратном уровне. Кластеры состоят из одного или нескольких подряд идущих секторов. Файл – это множество кластеров, связанных по определенному закону.

Работа с конфиденциальными документами обычно связана со следующими действиями с файлами: создание, хранение, коррекция и уничтожение.

Для защиты конфиденциальной информации применяется шифрование. Надежность защиты информации зависит не только от алгоритмов шифрования и стойкости криптографических ключей, но и от особенностей работы обыкновенных текстовых редакторов и баз данных, применяемых для создания и коррекции конфиденциальных документов. Подобные программные средства в процессе работы создают в оперативной или внешней памяти временные копии документов, с которыми они работают. Эти временные файлы не шифруются и могут быть использованы злоумышленником.

Необходимо также иметь в виду, что при записи отредактированной информации меньшего объема в тот же файл, где хранилась исходная информация до начала ее редактирования, образуются так называемые «хвостовые» кластеры, в которых информация полностью сохраняется. И тогда информация из «хвостовых» кластеров не только не шифруется, но и остается неповрежденной даже при применении средств гарантированного стира-

ния. Со временем информация из «хвостовых» кластеров затирается данными из других файлов, однако по оценкам специалистов через сутки из «хвостовых» кластеров можно извлечь до 85% исходной информации.

Пользователям надо иметь в виду, что команда удаления DEL операционной системы DOS не изменяет содержание файла, и оно может быть восстановлено в любой момент, если поверх не была записана другая информация.

Распространенные средства гарантированного стирания файлов сначала записывают на его место случайные числа, а потом удаляют файл стандартными средствами DOS. Однако и такие средства оказываются малоэффективными против программных закладок, которые предназначены для увеличения в «мусоре» фрагментов конфиденциальной информации. Например, закладка может инициировать статическую ошибку, пометив один или несколько кластеров меткой «сбойный». В итоге при удалении файла та его часть, которая размещена в «сбойных» кластерах, останется нетронутой и может быть восстановлена с помощью стандартных утилит.

Наблюдение и компрометация. При использовании модели воздействия типа «наблюдение» программная закладка размещается в сетевом или телекоммуникационном программном обеспечении. Так как это программное обеспечение постоянно находится в активном состоянии, то программная закладка может следить за процессами обработки информации, а также осуществлять установку новых и удаление других закладок.

Модель типа «компрометация» позволяет получать доступ к информации, перехваченной другими закладками. Например, инициируется постоянное обращение к такой информации, приводящее к росту сигнал/шум, что значительно облегчает перехват побочных излучений компьютерной системы и выделение сигналов, генерируемых закладкой типа «компрометация».

1.10 Защита от программных закладок

Защита от программных закладок может рассматриваться в трех аспектах [4]:

- предотвращение внедрения программной закладки;
- обнаружение внедренной программной закладки;
- удаление программной закладки.

Проблемы защиты от программных закладок сходны с проблемами защиты от вирусов. Обнаружение программных закладок можно эффективно осуществить, контролируя целостность информации запускаемых системных и прикладных программ, а также критические для работы системы события. При этом средства контроля сами не должны быть подвержены влиянию программных закладок, которые могут:

- искажать результаты контрольных действий;
- влиять на процесс считывания и запуск контролируемых программ;
- изменять алгоритмы функционирования средств контроля.

Выявление внедренной программной закладки. Обнаружение внедренной программной закладки заключается в выявлении признаков ее присутствия в компьютерной системе. Эти признаки могут быть *качественными* и *визуальными*, а также *обнаруживаемыми средствами тестирования и диагностики.*

Качественные и визуальные признаки обнаруживаются пользователем компьютерной системы по определенным отклонениям в ее работе, например, по изменению состава и длины файлов, замедлению или ускорению работы программ и т.п.

Признаки, выявляемые с помощью средств тестирования и диагностики, однотипны как для программных закладок, так и для компьютерных вирусов. Например, загрузочные закладки обнаруживаются антивирусными программами, сигнализирующими о наличии подозрительного кода в загрузочном секторе диска. Инициирование статической ошибки обнаруживает программа Disk Doctor. Средства проверки целостности данных на диске типа Adinf выявляют изменения в файлах, вызванные программными закладками. Эффективен также поиск фрагментов кода программных закладок по характерным для них последовательностям нулей и единиц (сигнатурам), а также разрешение выполнения только программ с известными сигнатурами.

Программные средства для защиты от троянских программ используют так называемое *согласование объекта*. При этом в качестве объектов фигурируют файлы и каталоги, а согласование заключается в установлении факта неизменности файлов и каталогов с момента последней проверки. В процессе согласования характеристики объектов сравниваются с характеристиками, которые они имели раньше. Например, архивная копия системного файла и ее атрибуты сравнивается с атрибутами этого же файла, находящегося на диске в данный момент времени.

Одним из атрибутов любого файла являются сведения о времени его последней модификации, которое фиксируется автоматически при внесении поправки в файл. Но фиксация времени модификации файла не может служить надежным признаком наличия в системе троянской программы, так как можно подкрутить системные часы назад, модифицировать файл, а затем снова вернуть часы в исходное состояние, и отметка времени об изменении файла останется неизменной. Если доступ к системным часам для пользователя запрещен, то такой метод выявления программной закладки может оказаться действенным.

Для обнаружения программной закладки можно использовать контроль размера файла. Если файл текстовый и занимает на диске, например 10 Кбайт, то после незначительного редактирования он будет иметь такой же размер, иначе говоря, килобайт не является достаточно точной единицей определения размера файла. Иначе обстоит дело с двоичными файлами. Внедрить постороннюю программу в исходную с точностью до одного бита при сохранении работоспособности исходной программы после компиляции – задача далеко не простая.

Злоумышленник обычно пытается сделать троянскую программу частью системного файла. Такие файлы входят в дистрибутив операционной системы, и их присутствие на компьютере вполне обоснованно. Но любой системный файл имеет известную длину, определяющую эталон контрольной суммы битов. Если этот атрибут будет изменен, то пользователь обратит на это внимание. Для маскировки злоумышленник тщательно проанализирует исходный текст соотоветствующей программы на наличие избыточных элементов, удалит их и вставит в сократившуюся программу свою закладку. Если размер полученного двоичного

кода будет больше или меньше размера исходного, то процедура повторяется до тех пор, пока указанные размеры файлов не станут одинаковыми. Описанная процедура может быть выполнена только очень квалифицированным злоумышленником.

Для проверки целостности файловой системы применяется алгоритм вычисления контрольной суммы – *одностороннее хэширование*. Напомним, что функция называется односторонней, если нахождение двух аргументов, для которых ее значения совпадают, является труднорешаемой задачей. Все попытки злоумышленника изменить какой-либо файл таким образом, чтобы значение контрольной суммы, полученное путем одностороннего хэширования, осталось неизменным, обречены на провал.

Защита от внедрения программных закладок. Надежным средством защиты от внедрения программной закладки является использование изолированного компьютера. Компьютер является изолированным, если:

- в нем BIOS и операционная система не содержат программных закладок;
- гарантированно установлена неизменность BIOS и операционной системы в данном сеансе работы компьютера;
- на компьютере не запускалось и не запускается никаких других программ, не проверенных на закладки;
- исключен запуск проверенных программ вне изолированного компьютера.

Для определения изолированности компьютера может быть осуществлен ступенчатый контроль. Сначала проверяется наличие закладок BIOS. При положительном результате считываются для проверки загрузочный сектор диска и драйверы операционной системы. Затем при помощи операционной системы активизируется драйвер контроля вызовов программ, который отслеживает, чтобы в компьютере запускались только проверенные программы.

Действенный метод борьбы с внедрением программных закладок применяется в банковской системе, в которой обрабатываются исключительно файлы-документы. Чтобы препятствовать проникновению программной закладки через каналы связи, в

системе не допускается прием исполняемого кода. Применяется контроль на присутствие в файле запрещенных символов, которые не встречаются в файлах-документах.

Недавно на рынке появились программные пакеты для комплексной защиты от угроз при работе в Internet [4]. Одним из таких пакетов является eSafe Protect компании Aladdin Knowledge Systems. Функционально eSafe Protect делится на три компонента – антивирус, персональный брандмауэр и модуль защиты компьютерных ресурсов. Антивирус избавляет компьютер от вредоносных программ с помощью антивирусного модуля VisuSafe. Персональный брандмауэр контролирует весь входящий и выходящий трафик по протоколу TCP/IP, наделяя используемые IP-адреса определенными правами (например, ограничивая доступ в Internet в определенные часы или запрещая посещение некоторых Web-узлов). С помощью специального окна осуществляется доступ к конфигурационным настройкам входящих в eSafe Protect компонентов, производится запуск антивирусной программы и дискретно устанавливается степень защищенности системы.

Для защиты ресурсов компьютера создается специальная изолированная область – *песочница*. Все автоматически загружаемые из Internet Java-аплеты и компоненты ActiveX сначала помещаются в «песочницу», где они находятся под наблюдением eSafe Protect. Если попавшая в «песочницу» программа попытается выполнить какое-то непредусмотренное действие, то оно будет немедленно блокировано. В течение заданного интервала времени (от 1 до 30 дней) каждое приложение, загруженное из Internet, проходит «карантинную» проверку в «песочнице». Полученная в ходе такой проверки информация заносится в особый журнал. По истечении «карантина» приложение будет выполняться вне «песочницы», но ему будут дозволены только те действия, перечень которых определяется на основе журнальных записей.

Таким образом, по сравнению с другими подобными программными пакетами eSafe Protect обеспечивает наиболее развитые и эффективные средства комплексной защиты от троянских программ. Входящий в состав eSafe Protect антивирус помогает быстро выявлять троянцев. Персональный брандмауэр блокирует любые попытки связаться извне с проникнувшими в компьютер-

ную систему троянскими программами. С помощью «песочницы» своевременно предотвращается внедрение троянцев в компьютеры под видом Java-аплетов и компонентов ActiveX.

Удаление внедренной программной закладки. Предпочтительный способ удаления программной закладки определяется методом ее внедрения. Если имеют дело с программно-аппаратной закладкой, то следует перепрограммировать ПЗУ компьютера. Если это загрузочная, драйверная, прикладная, замаскированная закладка или закладка-имитатор, то можно заменить их на соответствующую загрузочную запись, драйвер, утилиту, прикладную или служебную программу, полученную от источника, заслуживающего доверия. И наконец, если это исполняемый программный модуль, то можно попытаться получить его исходный текст, убрать из него закладки или подозрительные фрагменты, а затем заново откомпилировать.

1.11 Политика безопасности

Рассматривая вопросы безопасности информации в компьютерных системах, можно говорить о наличии некоторых «желательных» состояний данных систем [5]. Эти желательные состояния описывают «защищенность» системы. Понятие «защищенности» принципиально не отличается от любых других свойств технической системы, например, «надежной» работы, и является для системы априорно заданным. Особенностью понятия «защищенность» является его тесная связь с понятиями «злоумышленник» (как обозначение внешней причины для вывода системы из состояния «защищенности») или «угроза» (понятие, обезличивающее причину вывода системы из защищенного состояния из-за действий злоумышленника).

При рассмотрении понятия «злоумышленник» практически всегда выделяется объект его воздействия – часть системы, связанная с теми или иными действиями злоумышленника («объект атаки»). Таким образом, можно выделить три компоненты, связанные с нарушением безопасности системы: «злоумышленник» – внешний по отношению к системе источник нарушения свойства «безопасность», «объект атаки» – часть, принадлежащая системе,

на которую злоумышленник производит воздействие, «канал воздействия» – среда переноса злоумышленного воздействия.

Интегральной характеристикой, описывающей свойства защищаемой системы, является политика безопасности (ПБ) – качественное (или качественно-количественное) описание свойств защищенности, выраженное в терминах, описывающих систему. Описание политики безопасности может включать или учитывать свойства злоумышленника и объект атаки.

Описание политики безопасности включает:

1. Множество возможных операций над объектами.

2. Для каждой пары «субъект-объект» (S_iO_i) назначение множества разрешенных операций, являющегося подмножеством всего множества возможных операций. Операции связаны обычно с целевой функцией защищаемой системы (т.е. с категорией, описывающей назначение системы и решаемые задачи), например, операциями «создание объекта», «удаление объекта», «перенос информации от произвольного объекта к предопределенному – чтение» и т.д.

В теории компьютерной безопасности практически всегда рассматривается модель произвольной компьютерной системы (КС) в виде конечного множества элементов. Указанное множество можно разделить на два подмножества: множество объектов и множество субъектов. Данное разделение основано на свойстве элемента «быть активным» или «получать управление» (применяются также термины «использовать ресурсы» или «пользоваться вычислительной мощностью»). На основе модели вычислительной системы последовательность исполняемых инструкций (программа, соответствующая понятию «субъект») находится в единой среде с данными (соответствующими понятию «объект»).

Подчеркнем отличие понятия субъекта КС от человека-пользователя следующим образом. Пользователь – физическое лицо, аутентифицируемое некоторой информацией и управляющее субъектом КС через органы управления ЭВМ. Пользователь является, таким образом, внешним фактором, управляющим состоянием субъектов. Пользователь не может влиять на свойства субъекта.

Можно сформулировать аксиомы защищенных компьютерных систем:

Аксиома 1. В защищенной КС всегда присутствует активная компонента (субъект), выполняющая контроль операций субъектов над объектами. Данная компонента фактически отвечает за реализацию некоторой политики безопасности.

Аксиома 2. Для выполнения в защищенной КС операций над объектами необходима дополнительная информация (и наличие содержащего ее объекта) о разрешенных и запрещенных операциях субъектов с объектами.

В данном случае мы оперируем качественными понятиями «контроль», «разрешенная и запрещенная операция».

Аксиома 3.

Все вопросы безопасности информации описываются доступами субъектов к объектам.

Важно отметить, что политика безопасности описывает в общем случае нестационарное состояние защищенности. КС может изменяться, дополняться новыми компонентами. Очевидно, что политика безопасности должна быть поддержана во времени, следовательно, в процесс изучения свойств защищаемой системы должны быть добавлены процедуры *управления безопасностью.*

С другой стороны, нестационарность защищаемой КС, а также вопросы реализации политики безопасности в конкретных конструкциях защищаемой системы (например, программирование контролирующего субъекта в командах конкретного процессора компьютера) предопределяют необходимость рассмотрения задачи *гарантирования заданной политики безопасности.*

Таким образом, компьютерная безопасность решает четыре класса взаимосвязанных задач:

1. Формулирование и изучение политик безопасности.
2. Реализация политик безопасности.
3. Гарантирование заданной политики безопасности.
4. Управление безопасностью.

Типовой *жизненный цикл КС* состоит из следующих стадий:

1. Проектирование КС и проектирование политики безопасности.

2. Моделирование ПБ и анализ корректности ПБ, включающий установление адекватности ПБ и целевой функции КС.

3. Реализация ПБ и механизмов ее гарантирования, а также процедур и механизмов управления безопасностью.

4. Эксплуатация защищенной системы.

Безопасность КС достаточно часто описывается в категориях «достоверность», «конфиденциальность», «целостность» и «доступность».

Свойство *достоверности* понимается как сохранение информацией своих семантических свойств в любой момент времени от момента ввода в систему. Свойство *доступности* понимается как возможность пользования ресурсом КС и информацией в произвольный момент времени. Свойство *целостности* (связанное со свойством достоверности) подразумевает неизменность свойств информации и ресурсов в любой момент времени от момента их порождения или ввода в систему. Свойство *конфиденциальности* понимается как недоступность информации или сервисов для пользователей, которым априорно не задана возможность использования указанных сервисов или информации.

Рассмотрим также качественное описание и классификацию различных угроз в КС.

По цели реализации угрозы – нарушение конфиденциальности, целостности, доступности.

По принципу и типу воздействия – с использованием физического доступа (локально) или удаленно – пассивно (с использованием каналов утечки) и активно (с использованием каналов удаленного воздействия). Более общим понятием по сравнению с каналом утечки является канал воздействия на КС. Он может включать изменение компонент КС – активное воздействие – угроза свойству целостности. Несанкционированный доступ в КС может иметь как пассивный, так и активный характер, поэтому его корректнее отнести к воздействию на КС.

По используемым средствам атаки на КС – с использованием штатных средств КС и с использованием дополнительных средств.

1.12 Модель КС. Понятие монитора безопасности

Модели, связанные с реализацией ПБ, не учитывают возможности субъектов по изменению КС, которые могут привести

к изменению ее свойств и, как предельный случай, к полной неприменимости той или иной модели к описанию отношений «субъект-объект» в измененной КС.

Этот факт не является недостатком политики безопасности. Достоверность работы механизмов реализации политики безопасности считается априорно заданной, поскольку в противном случае невозможна формализация и анализ моделей. Однако вопрос гарантий политики безопасности является ключевым как в теории, так и в практике.

Рассматривая активную роль субъектов в КС, необходимо упомянуть о ряде важнейших их свойств, на которых базируется модель КС [5].

Во-первых, необходимо заметить, что человек-пользователь воспринимает объекты и получает информацию о состоянии КС через субъекты, которыми он управляет и которые производят отображение информации в воспринимаемом человеком виде.

Во-вторых, угрозы компонентам КС (КС рассматривается в модели потоков или состояний) исходят от субъектов как активной компоненты, порождающей потоки и изменяющей состояние объектов в КС.

В-третьих, субъекты могут влиять друг на друга через изменяемые ими объекты, связанные с другими субъектами, порождая в конечном итоге в системе субъекты (или состояния системы), которые представляют угрозу для безопасности информации или работоспособности самой системы.

Будем считать разделение КС на субъекты и объекты априорным. Будем считать также, что существует априорный безошибочный критерий различения субъектов и объектов КС (по свойству активности). Кроме того, считаем в условиях всех утверждений, что декомпозиция КС на субъекты и объекты фиксирована.

Будем также полагать, что в любой дискретный момент времени множество субъектов КС не пусто.

Аксиома 4. Субъекты в КС могут быть порождены только активной компонентой (субъектами) из объектов.

Специфицируем механизм порождения новых субъектов следующими определениями.

Определение 1. Объект $\mathbf{O_i}$ называется источником для субъекта $\mathbf{S_m}$, если существует субъект $\mathbf{S_j}$, в результате воздействия которого на объект $\mathbf{O_i}$ в КС возникает субъект $\mathbf{S_m}$.

Субъект $\mathbf{S_j}$, порождающий новый субъект из объекта $\mathbf{O_i}$, в свою очередь, называется активизирующим субъектом для порожденного субъекта S_m.

Введем обозначение: ***Create (*$\mathbf{S_j}$, $\boldsymbol{O_i}$*)->*** $\mathbf{S_k}$ из объекта $\mathbf{O_i}$ порожден субъект $\mathbf{S_k}$ при активизирующем воздействии субъекта $\mathbf{S_j}$. ***Create*** назовем операцией порождения субъектов (рис. 1).

Операция ***Create*** задает отображение декартова произведения множеств субъектов и объектов на объединение множества субъектов с пустым множеством. Заметим также, что в КС действует дискретное время и фактически новый субъект $\mathbf{S_k}$ порождается в момент времени t+1 относительно момента t, в который произошло воздействие порождающего субъекта на объект-источник.

Считаем, что если ***Create (*** S_j, $\boldsymbol{O_i}$***)>NULL*** **(**конструкция **NULL** далее обозначает пустое множество), то порождение нового субъекта из объекта $\mathbf{O_i}$ при активизирующем воздействии $\mathbf{S_j}$ невозможно. Так, практически во всех операционных средах существует понятие исполняемого файла – объекта, могущего быть источником для порождения субъекта. Например, для MS DOS файл edit.com является объектом-источником для порождения субъекта-программы текстового редактора, а порождающим субъектом является, как правило, командный интерпретатор shell (объект-источник – command.com).

Из архитектуры фон Неймана следует также, что с любым субъектом связан (или ассоциирован) некоторый объект (объекты), отображающий его состояние,– например, для активной программы (субъекта) ассоциированным объектом будет содержание участка оперативной памяти с исполняемым кодом данной программы.

Определение 2. Объект $\mathbf{O_i}$ в момент времени t ассоциирован с субъектом $\mathbf{S_m}$, если состояние объекта $\mathbf{O_i}$ повлияло на состояние субъекта в следующий момент времени (т.е. субъект использует информацию, содержащуюся в объекте).

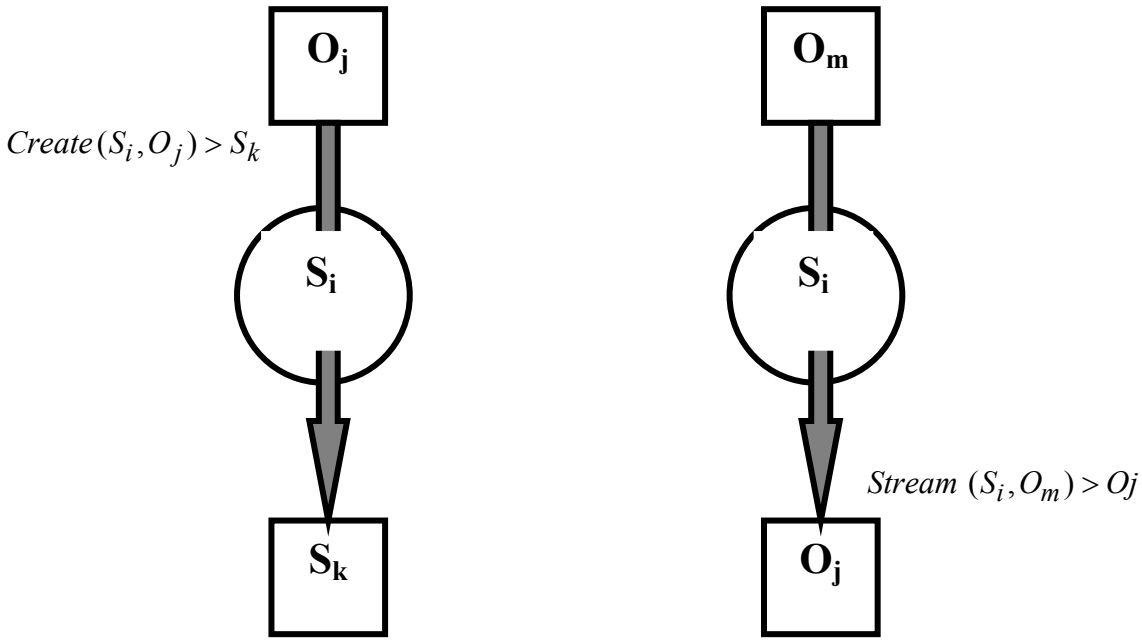


Рис. 1 – Порождение субъекта и понятие потока

Введем обозначение «множество объектов **{O_m}t** ассоциировано с субъектом **S_j** в момент времени t»: **S_j({O_m}t)**.

В данном случае определение не в полной мере является формально строгим, поскольку состояние субъекта описывается упорядоченной совокупностью ассоциированных с ним объектов.

В момент порождения субъекта **S_m** из объекта **O_i** он является ассоциированным объектом для субъекта **S_m**.

Определение 3. Потоком информации между объектом **O_m** и объектом **O_J** называется произвольная операция над объектом **O_J**, реализуемая в субъекте **S_j** и зависящая от **O_m**.

Заметим, что как **O_J**, так и **O_m** могут быть ассоциированными или неассоциированными объектами, а также «пустыми» объектами (NULL).

Обозначения: ***Stream(S_i,O_m)->O_j*** – поток информации от объекта **O_m** к объекту **O_J**. При этом будем выделять источник (**O_m**) и получатель (приемник) потока (**O_J**). В определении подчеркнуто, что поток информации рассматривается не между субъектом и объектом, а между объектами. Активная роль субъекта выражается в реализации данного потока (это означает, что операция порождения потока локализована в субъекте и отображается состоянием его функционально ассоциированных объек-

тов). Отметим, что операция ***Stream*** может создавать новый объект или уничтожать его.

Определение 4. Доступом субъекта $\mathbf{S_j}$ к объекту $\mathbf{O_J}$ будем называть порождение потока информации между некоторым объектом (например, ассоциированным с субъектом объектами $S_i(O_m)$ и объектом $\mathbf{O_J}$).

Выделим все множество потоков **P** для фиксированной декомпозиции КС на субъекты и объекты во все моменты времени (все множество потоков является объединением потоков по всем моментам дискретного времени) и произвольным образом разобьем его на два непересекающихся подмножества: **N** и **L**, $\mathbf{P=N\cup L}$.

Обозначим:

N – подмножество потоков, характеризующее несанкционированный доступ;

L – подмножество потоков, характеризующее легальный доступ.

Дадим некоторое пояснение к разделению множеств **N** и **L**. Понятие «безопасность» подразумевает наличие и некоторого состояния «опасности» – нежелательных состояний какой-либо системы (в данном случае КС). Будем считать парные категории типа «опасный – безопасный» априорно заданными для КС и описываемыми политикой безопасности, а результатом применения политики безопасности к КС – разделение на множество «опасных» потоков **N** и множество «безопасных» потоков **L**. Деление на потоки можно описывать как свойство целостности (потоки из **N** нарушают целостность КС) или свойство конфиденциальности (потоки из **N** нарушают конфиденциальность КС), так и любое другое свойство.

Определение 5. Правила разграничения доступа субъектов к объектам есть формально описанные потоки, принадлежащие подмножеству **L**.

В предлагаемой субъектно-ориентированной модели не производятся уточнения известных моделей политики безопасности (политика безопасности описывает только критерии разбиения на множества), но формулируются условия корректного существования элементов КС, обеспечивающих реализацию той или иной политики безопасности.

Определение 6. Монитор обращений (МО) – субъект, активизирующийся при возникновении потока от любого субъекта к любому объекту.

Можно выделить два вида МО:

Индикаторный МО – устанавливающий только факт обращения субъекта к объекту.

Содержательный МО – субъект, функционирующий таким образом, что при возникновении потока от ассоциированного объекта $\mathbf{O_m}$ любого субъекта $\mathbf{S_i(S_i((O_m))}$ к объекту $\mathbf{O_J}$ и обратно существует ассоциированный с МО объект $\mathbf{O_mo}$ (в данном случае речь идет об ассоциированных объектах-данных), тождественный объекту $\mathbf{O_m}$ или $\mathbf{S_j(O_m)}$.

Определение 7. Монитор безопасности объектов (МБО) – монитор обращений, который разрешает поток, принадлежащий только множеству легального доступа L. Разрешение потока в данном случае понимается как выполнение операции над объектом – получателем потока, а запрещение – как невыполнение.

1.13 Обеспечение гарантий выполнения политики безопасности

Представляется очевидным, что при изменении функционально ассоциированных с субъектом реализации политики безопасности (МБО) объектов могут измениться и свойства самого МБО, заключающиеся в фильтрации потоков, и, как следствие, могут возникнуть потоки, принадлежащие множеству N (рис. 2). Введем в связи с этим понятие корректности субъектов [5].

Определение 8. Пара субъектов S_i и S_j называются не влияющими друг на друга (или корректными относительно друг друга), если в любой момент времени отсутствует поток (изменяющий состояние объекта) между ассоциированным объектом субъекта S_i(Osi) и S_j(Osj), причем Osj не является ассоциированным объектом S_i, а Osi не является ассоциированным объектом.

Смысл понятия корректности можно пояснить на примере: существующие в едином пространстве ОП программы не должны иметь функциональных возможностей изменения «чужого» вектора кода и состояния переменных.

Определение 9. Пара субъектов называются абсолютно не влияющими друг на друга (или абсолютно корректными относительно друг друга), если множества ассоциированных объектов указанных субъектов не имеют пересечения.

Абсолютная корректность легко достижима в случае виртуального адресного пространства.

Определение абсолютной корректности позволяет сформулировать условия гарантированного осуществления только легального доступа.

Утверждение 1. *Достаточное условие гарантированного выполнения политики безопасности в КС.*

Монитор безопасности объектов разрешает порождение потоков только из множества **L**, если все существующие в системе субъекты абсолютно корректны относительно него и друг друга.

Доказательство.

Условие абсолютной корректности предполагает неизменность функционально ассоциированных объектов МБО (поскольку потоков, изменяющих ассоциированные объекты МБО, не су-

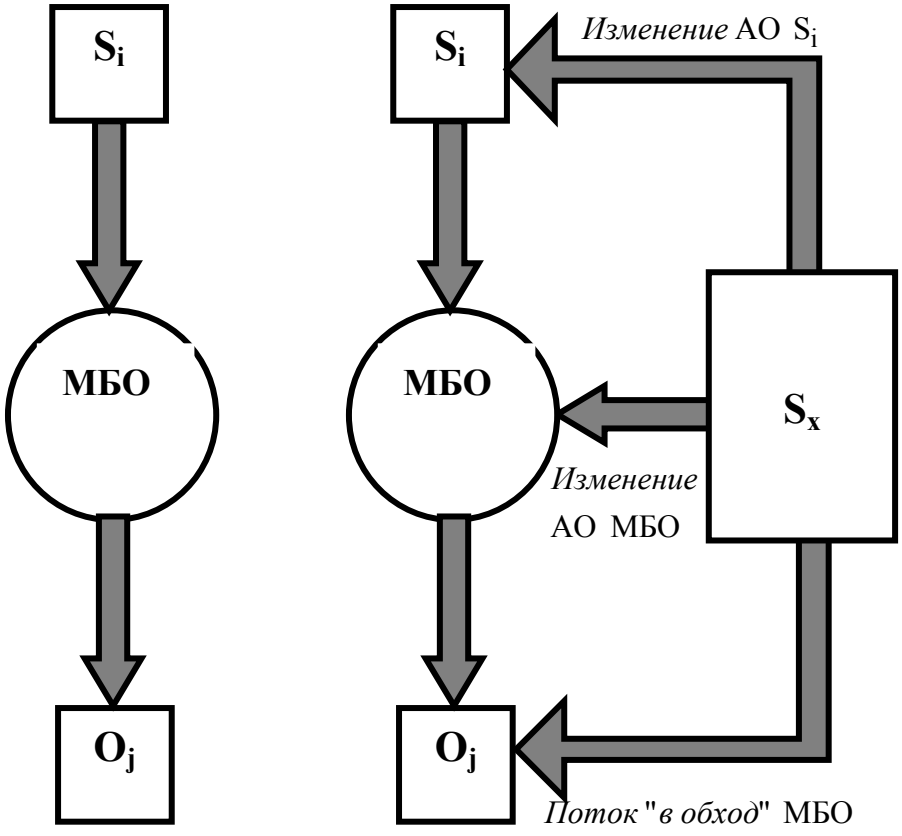


Рис. 2 – Возможные пути нарушения ПБ (АО – ассоциированные объекты)

ществует). С другой стороны, такие потоки могут появиться при изменении ассоциированных объектов, принадлежащих другим субъектам КС (изменятся свойства субъекта, в том числе и возможно по порождению потоков к МБО). Условие корректности субъектов относительно друг друга делает это невозможным по определению абсолютной корректности. Это, в свою очередь, означает, что МБО реализует только потоки из подмножества **L**. Утверждение доказано.

Определение 10. Монитор порождения субъектов (МПС) – субъект, активизирующийся при любом порождении субъектов.

По аналогии с переходом от МО к МБО введем понятие монитора безопасности субъектов.

Определение 11. Монитор безопасности субъектов (МБС) – субъект, который разрешает порождение субъектов только для фиксированного подмножества пар активизирующих субъектов и порождающих объектов.

Воздействие МБС выделяет во всем множестве субъектов **S** подмножество разрешенных **E**.

Глава 2. ЗАМКНУТАЯ ПРОГРАММНАЯ СРЕДА

2.1 Основные определения

Определение 12. КС называется замкнутой по порождению субъектов, если в ней действует МБС, разрешающий порождение только фиксированного конечного подмножества субъектов для любых объектов-источников, рассматриваемых для фиксированной декомпозиции компьютерной системы на субъекты и объекты [5].

При рассмотрении вопросов реализации защищенных сред будет рассматриваться термин «замкнутая программная среда», который по существу эквивалентен приведенному определению.

Однако замкнутости КС по порождению субъектов недостаточно для описания свойств системы в части защищенности, поскольку необходимо обеспечить корректность порождаемых МБС субъектов относительно его самого и МБО. Механизм замкнутой программной среды сокращает множество возможных субъектов до некоторого множества фиксированной мощности, но при этом допускает существование некорректных субъектов, включенных в замкнутую среду.

Сформулируем определение изолированности КС.

Определение 13. Множество субъектов КС называется изолированным (абсолютно изолированным), если в ней действует МБС и субъекты из порождаемого множества корректны относительно друг друга и МБС.

С учетом ограничения множества субъектов за счет применения механизма МБС можно сформулировать утверждение о достаточном условии гарантированного выполнения политики безопаности по иному: если в абсолютно изолированной КС существует МБО и порождаемые субъекты абсолютно корректны относительно МБО, а также МБС абсолютно корректен относительно МБО, то в такой системе реализуется только доступ, описанный в правилах разграничения доступа.

При рассмотрении технической реализации изолированности субъектов в КС будет употребляться термин «изолированная программная среда» (ИПС), который описывает механизм реализации изолированности.

Определение 13. Операция порождения субъекта ***Create (*$\mathbf{S_k}$, $\boldsymbol{O_m}$*)->*** $\mathbf{S_i}$ называется порождением с контролем неизменности объекта, если для любого момента времени t>t0, в который активизирована операция порождения ***Create***, порождение субъекта $\mathbf{S_i}$ возможно только при тождественности объектов **Om**[t0] и **Om**[t].

Следствие. В условиях определения 13 порожденные субъекты Si[t1] и Si[t2] тождественны, если t1>t0 t2>t0. При t1= t2 порождается один и тот же субъект.

При порождении субъектов с контролем неизменности объекта в КС допустимы потоки от субъектов к объектам-источникам, участвующим в порождении субъектов, с изменением их состояния.

Утверждение 2 (*базовая теорема ИПС).*

Если в момент времени t0 в изолированной КС действует только порождение субъектов с контролем неизменности объекта и существуют потоки от любого субъекта к любому объекту, не противоречащие условию корректности субъектов, то в любой момент времени t>t0 КС также остается изолированной (абсолютно изолированной).

Доказательство.

По условию утверждения в КС возможно существование потоков, изменяющих состояние объектов, не ассоциированных в этот момент времени с каким-либо субъектом. Если объект с измененным состоянием не является источником для порождения субъекта, то множество субъектов изолированной среды нерасширяемо, в противном случае (измененный объект является источником для порождения субъекта) по условиям утверждения (порождение субъекта с контролем) порождение субъекта невозможно. Следовательно, мощность множества субъектов не может превышать той, которая была зафиксирована до изменения состояния любого объекта. По следствию из определения 13 (о замкнутости множества субъектов в ИПС с невозрастанием мощности множества субъектов) получим, что множество субъектов КС изолировано. Утверждение доказано.

Можно сформулировать методологию проектирования гарантированно защищенных КС. Сущность данной методологии состоит в том, что при проектировании защитных механизмов КС необходимо опираться на совокупность приведенных выше в ут-

верждениях достаточных условий, которые должны быть реализованы для субъектов, что гарантирует защитные свойства, определенные при реализации МБО в КС (т.е. гарантированное выполнение заданной МБО политики безопасности).

Рассмотренная концепция изолированной программной среды является расширением зарубежных подходов к реализации ядра безопасности.

Обычно модель функционирования ядра безопасности изображается в виде следующей схемы, показанной на рис. 3.

На рис. 3 «база данных защиты» означает объект, содержащий в себе информацию о потоках множества L (защита по «белому списку» – разрешение на потоки) или N (защита по «черному списку» – запрещение на потоки).

Для учета влияния субъектов в КС необходимо рассматривать расширенную схему взаимодействия элементов системы реализации и гарантирования ПБ.

Рис. 4 наглядно поясняет взаимодействие элементов ядра безопасности с учетом контроля порождения субъектов. На рис. 4 подчеркнута роль монитора безопасности при порождении субъектов из объектов. Взаимодействие субъектов и объектов при порождении потоков уточнено введением ассоциированных с субъектом объектов.

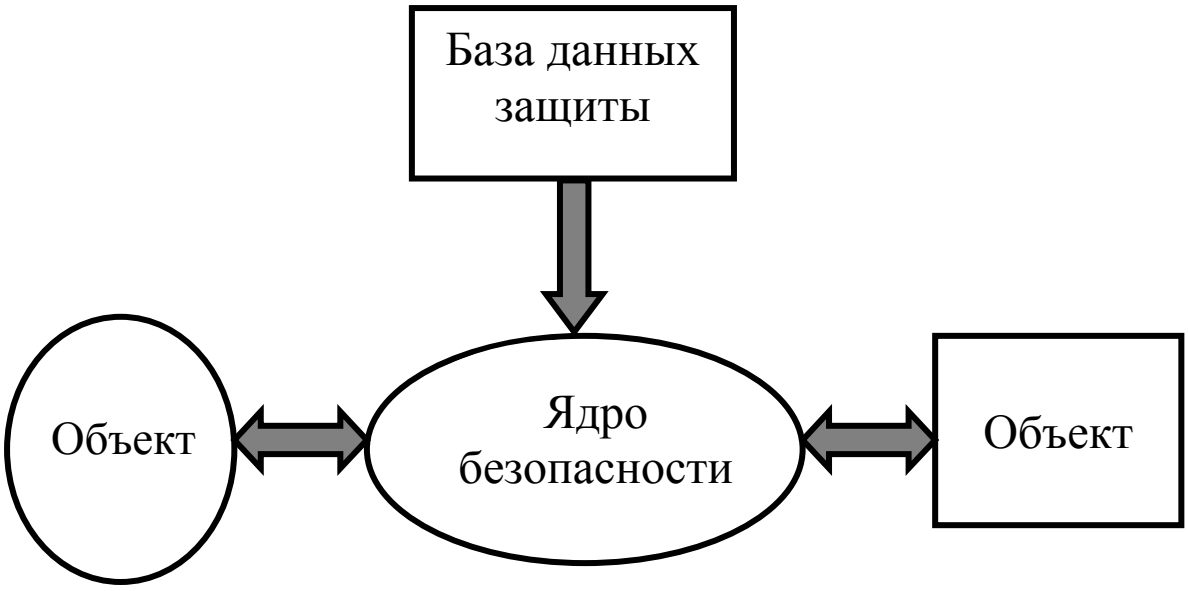


Рис. 3 – Классическая модель ядра безопасности

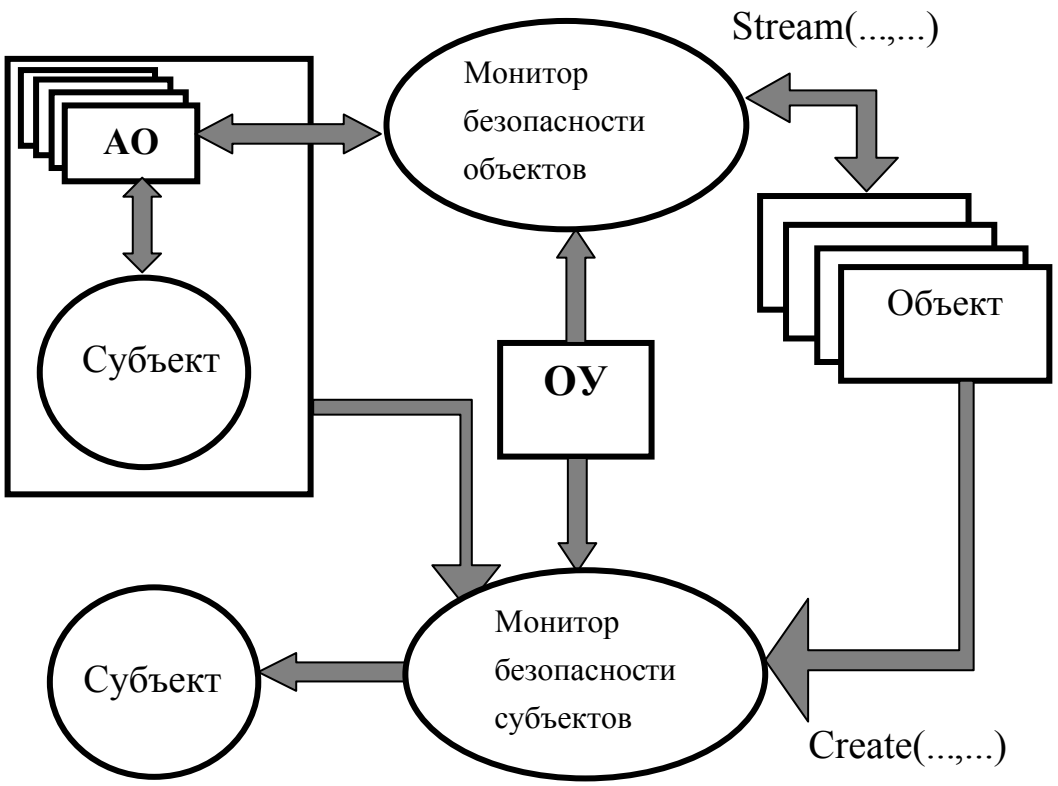


Рис. 4 – Ядро безопасности с учетом контроля порождения субъектов

Конструкция ОУ на схеме обозначает объект управления, т.е. объект, содержащий информацию о разрешенных значениях ***Stream*** (об элементах множества L или N) и ***Create*** (элементы множества E). Объект управления может быть ассоциирован (ассоциированный объект-данные) как с МБО, так и с МБС.

Перейдем к описанию практических методов построения изолированной программной среды. Целью рассмотрения практических подходов является иллюстрация утверждения о том, что достаточные условия гарантированной защищенности могут быть практически выполнены в реальных КС.

2.2 Метод генерации изолированной программной среды при проектировании механизмов гарантированного поддержания политики безопасности

Опираясь на базовую теорему ИПС, опишем метод субъектно-объектного взаимодействия в рамках ИПС для более конкретной архитектуры КС [5].

Из теоремы следует, что для создания гарантированно защищенной КС в смысле выполнения политики безопасности необходимо:

1. Убедиться в попарной корректности субъектов, замыкаемых в ИПС (либо убедиться в корректности любого субъекта относительно МБО и МБС).

2. Спроектировать и реализовать программно (или программно-аппаратно) МБС так, чтобы:

- для любого субъекта и любого объекта производился контроль порождения субъектов (т.е. чтобы реализация МБС соответствовала его определению;
- порождение любого субъекта происходило с контролем неизменности объекта-источника.

3. Реализовать МБО в рамках априорно сформулированной политики безопасности.

Надо заметить, что приводимые выше утверждения верны только тогда, когда описанная и реализованная ПБ не нарушает их условий (проверка данного факта зависит от модели ПБ и является отдельной весьма важной задачей).

Кроме того, необходимо обратить внимание на следующее. Объект управления, который является ассоциированным объектом МБС (обычно ассоциированный объект-данные), играет решающую роль в проектировании ИПС. При возможности изменения состояния объекта управления потенциально возможно «размыкание» программной среды, т.е. добавление к множеству разрешенных субъектов дополнительных, реализующих злоумышленные функции. С другой стороны, процесс управления безопасностью подразумевает возможность изменения объекта управления. Возможность изменения объекта управления (реализация потока ***Stream*** (субъект управления, АО объекты субъек-

та управления)->ОУ) должна присутствовать для выделенных субъектов.

Важную роль при проектировании ИПС играет свойство КС, заключающееся в поэтапной активизации субъектов из объектов различного уровня представления информации. Рассмотрим в таблице 1 иерархию уровней при загрузке операционной системы.

В таблице выделен термин «сектор» для обозначения представления объекта аппаратно-программного уровня. Он обозначает непрерывную последовательность элементов хранения (байт) на материальном носителе, характеризуемую местом расположения.

Термин «файл» обозначает абстрактный объект, построенный по списочной структуре из объектов «сектор». Объекты типа «файл» и «сектор» выделены исключительно исходя из типовой структуры объектов КС.

Таблица 1 – Иерархия уровней при загрузке ОС

Уровень	Субъект	Локализация	Представление информации	Через какие функции реализуются потоки
0	Субъект аппаратно-программного уровня	ПЗУ (Bios)	сектора	через микропрограммы ПЗУ
1	Субъект уровня первичной загрузки	Загрузчик ОС	сектора	через Bios или первичный загрузчик
2	Субъект уровня вторичного загрузчика (драйвер)	Драйверы ОС	сектора	через Bios или первичный загрузчик
3	Субъект уровня ОС	Ядро ОС	файлы	через драйверы
4	Субъект пользовательского уровня	Прикладные приложения	файлы	через ядро ОС

С учетом иерархической структуры представления объектов можно говорить о том, что в начальные этапы активизации КС декомпозиция на субъекты и объекты динамически изменяется. Следовательно, основная теорема ИПС может быть применима только на отдельных интервалах времени, когда уровень представления объектов постоянен и декомпозиция фиксирована. Можно утверждать, что ИПС, действующую от момента активизации до момента окончания работы КС, невозможно сформировать в начальный момент активизации КС.

Пусть в КС выделяется конечное число уровней представления объектов U={0,…,R}, где R – максимальный уровень представления объекта.

С точни зрения базовой теоремы ИПС имело бы смысл говорить о некотором «стационарном состоянии КС, когда в отображениях ***Stream*** и ***Create*** участвуют только объекты уровня R. Тогда реализация МБС может быть значительно упрощена (в том смысле, что все аргументы-объекты операции ***Create*** имеют тот же уровень). Необходимо обратить внимание на то, что такое требование, с одной стороны, может накладывать ограничительные условия на свойства прикладного ПО (невозможность инициирования потоков, включающих объекты уровня менее R, прикладными программами, а с другой стороны, быть следствием проектировочных решений реализации субъекта, локализованного в ядре операционной системы (примером является ОС Windows NT 4.0, запрещающая операции ниже уровня «файл» со стороны субъектов прикладного уровня).

Практическая реализация всех операционных систем позволяет выделить две фазы их работы: активизация субъектов с ростом уровня представления объектов (фаза загрузки или начальная фаза) и фаза стационарного состояния (когда уровень представления объектов не увеличивается). Конечно, необходимо сделать оговорку, касающуюся возможности реализации потоков к объектам нижнего уровня (операционные системы типа DOS, в которых возможна операция с любым объектом нижнего уровня (сектор) из программ прикладного уровня).

Тогда практическая реализация ИПС может состоять из двух этапов: предопределенное выполнение начальной фазы, включающее в себя момент активизации МБС (и МБО), и работа

в стационарной фазе в режиме ИПС (возможно, с контролем неизменности объектов-источников).

Введем понятие последовательности активизации компонент КС. Смысл вводимых понятий и формулируемых ниже утверждений состоит в необходимости приведения субъектов КС в одно и то же состояние после активизации первичного субъекта аппаратно-программного уровня или, иначе говоря, в задании предопределенной последовательности активизации субъектов КС.

Обозначим: ZL – последовательность пар (i,j)t (t=0,1,2,…,**l**-1 – моменты времени) длины **l**, такие, что ***Create(Si,Oj)***[t]->**Sm**[t+**l**].

Обозначим также:

Sz – множество всех субъектов, включенных в последовательность Z_L;

O_Z – множество всех объектов, включенных в последовательность Z_L.

Для многопотоковых КС можно рассматривать несколько (возможно, независимых друг от друга) последовательностей Z_Lи соответственно множеств Sz и O_Z.

Определение 14. Состоянием КС в момент времени t называется упорядоченная совокупность состояний субъектов.

Напомним, что каждый объект есть слово в априорно определенном языке, а понятие состояния субъекта сформулировано выше.

Утверждение 3 (*условие одинакового состояния КС*).

Состояние КС в моменты времени tx1 и tx2 (tx1 и tx2 исчисляются для двух отрезков активности КС от нулевого момента активизации КС to1 и to2 – например, включения питания аппаратной части) одинаково, если:

1) tx1=tx2;

2) тождественны субъекты Si[to1] и Si[to2];

3) неизменны все объекты из множества O_Z;

4) неизменна последовательность Z_L.

Доказательство по принципу математической индукции.

Верность утверждения при t=1 следует из определения тождественности субъектов.

Пусть утверждение верно для t=k<1.

Тогда в момент времени k+1 могут быть порождены только тождественные субъекты, поскольку тождественны активизирующие субъекты (по предположению индукции) и по условию утверждения неизменны элементы множества O_Z.

Длина l последовательности Z_L определяется:

1. По признаку невозможности управления субъектами, принадлежащими множеству Sz, со стороны пользователя (в противном случае последовательность активизации субъектов может быть изменена).

2. По признаку доступности для контроля неизменности всех объектов из множества O_Z.

3. По признаку невозрастания уровня представления информации (в данном случае имеется в виду, что существует момент времени tx такой, что для любого t>tx объект-аргумент O_j операции ***Create(Si,Oj)*** принадлежит одному уровню представления).

Необходимо заметить, что последовательность Z_L локализуется в некотором объекте либо совокупности объектов (например, для DOS последовательность активизации субъектов предопределена содержанием файлов AUTOEXEC.BAT и CONFIG.SYS) и неизменность последовательности Z_L тождественна неизменности указанных объектов, для ОС Windows NT последовательность активизации компонент определена содержанием соответствующих ключей реестра (registry).

Пусть в последовательности Z_L можно выделить z_i такое, что для любого z_k при k>i отображения ***Create*** и ***Stream*** используют только объекты уровня R. Другими словами, с момента времени i наступает стационарная фаза функционирования КС.

В этих условиях, а также при попарной корректности субъектов и действии МБС с контролем неизменности объектов-источников на уровне R с момента времени m>k верно:

Утверждение 4 *(достаточное условие ИПС при ступенчатой загрузке).*

При условии неизменности Z_L и неизменности объектов из O_Z в КС с момента времени установления неизменности Z_L и O_Z действует ИПС. Доказательство не приводим.

Утверждение 5 *(требования к субъектному наполнению ИПС).*

Для того, чтобы ИПС поддерживалась в течение всего времени активности КС, достаточно, чтобы в составе программного обеспечения, могущего быть инициированным в ИПС, не было функций порождения субъектов и прекращения их работы, кроме заранее предопределенных при реализации МБС, и не существовало возможностей влияния на среду выполнения (под средой выполнения понимается множество ассоциированных объектов) любого процесса, а также инициирования потоков к объектам логического уровня менее R.

Утверждение 6 *(достаточное условие чтения реальных данных).*

Если субъект, обслуживающий процесс чтения данных (т.е. указанный субъект инициируется запрашивающим данные субъектом и участвует в потоке) содержал только функции тождественного отображения данных на ассоциированные объекты-данные любого субъекта, инициирующего поток чтения, и целостность объекта-источника для этого субъекта зафиксирована, то при его последующей неизменности чтение с использованием порожденного субъекта будет чтением реальных данных.

2.3 Формирование и поддержка изолированной программной среды

Предположим, что в КС работают N субъектов-пользователей, каждый i-й из которых характеризуется некоторой персональной информацией Ki, не известной другим пользователям и хранящейся на некотором материальном носителе. Существует также выделенный субъект − администратор системы, который знает все Ki. Администратор КС присваивает i-му пользователю полномочия, заключающиеся в возможности исполнения им только заданного подмножества программ

$$Ti=\{Pi1, Pi2, \ldots Pit\}.$$

Несанкционированным доступом является использование имеющихся на жестком диске ПЭВМ программ либо субъектом, не входящим в N допущенных, либо i-м пользователем вне подмножества своих полномочий Ti. Субъект, пытающийся проделать данные действия, называется злоумышленником. НСД осу-

ществляется обязательно при помощи имеющихся на ПЭВМ или доставленных злоумышленником программных средств (в данном случае не рассматривается возможность нарушения целостности аппаратных средств компьютера). НСД может носить непосредственный и опосредованный характер. При непосредственном НСД злоумышленник, используя некоторое ПО, пытается непосредственно осуществить операции чтения или записи (изменения) интересующей его информации. Если предположить, что в Ti нет программ, дающих возможность произвести НСД (это гарантирует администратор при установке полномочий), то НСД может быть произведен только при запуске программ, не входящих в Ti. Опосредованный НСД обусловлен общностью ресурсов пользователей и заключается во влиянии на работу другого пользователя через используемые им программы (после предварительного изменения их содержания или их состава злоумышленником). Программы, участвующие в опосредованном НСД, будем называть разрушающими программными воздействиями (РПВ) или программными закладками. РПВ могут быть внедрены i-м пользователем в ПО, принадлежащее j-му пользователю только путем изменения программ, входящих в Tj. Следовательно, система защиты от НСД должна обеспечивать контроль за запуском программ, проверку их целостности и активизироваться всегда для любого пользователя. Выполнение контроля целостности и контроля запусков ведется на основе Ki для каждого пользователя. При этом внедренный в КС защитный механизм должен обеспечивать следующее:

- в некоторый начальный момент времени требовать у субъекта предъявления аутентифицирующей информации и по ней однозначно определять субъекта и его полномочия Ti;

- в течение всего времени работы пользователя i должен обеспечивать выполнение программ только из подмножества Ti;

- пользователь не должен иметь возможности изменить подмножество Ti и/или исключить из дальнейшей работы защитный механизм и его отдельные части.

Положим, что в ПЗУ (BIOS) и операционной среде (в том числе и в сетевом ПО) отсутствуют специально интегрированные в них возможности НСД. Пусть пользователь работает с программой, в которой также исключено наличие каких-либо

скрытых возможностей (проверенные программы). Потенциально злоумышленные действия могут быть такими:

1. Проверенные программы будут использованы на другой ПЭВМ с другим BIOS и в этих условиях использоваться некорректно.

2. Проверенные программы будут использованы в аналогичной, но не проверенной операционной среде, в которой они также могут использоваться некорректно.

3. Проверенные программы используются на проверенной ПЭВМ и в проверенной операционной среде, но запускаются еще и не проверенные программы, потенциально несущие в себе возможности НСД.

Тогда НСД в КС гарантированно невозможен, если выполняются условия:

У1. На ПЭВМ с проверенным BIOS установлена проверенная операционная среда.

У2. Достоверно установлена неизменность DOS и BIOS для данного сеанса работы.

У3. Кроме проверенных программ в данной программно-аппаратной среде не запускалось и не запускается никаких иных программ, проверенные программы перед запуском контролируются на целостность.

У4. Исключен запуск проверенных программ в какой-либо иной ситуации, т.е. вне проверенной среды.

У5. Условия У1-4 выполняются в любой момент времени для всех пользователей, аутентифицированных защитным механизмом.

При выполнении перечисленных условий программная среда называется изолированной (далее будем использовать термин ИПС – изолированная программная среда). Функционирование программ в изолированной программной среде (ИПС) существенно ослабляет требования к базовому ПО. В самом деле, ИПС контролирует активизацию процессов через операционную среду, контролирует целостность исполняемых модулей перед их запуском и разрешает инициирование процесса только при одновременном выполнении двух условий – принадлежности к разрешенным и неизменности.

В таком случае от базового ПО требуется только:

1. Невозможность запуска программ помимо контролируемых ИПС событий.

2. Отсутствие в базовом ПО возможностей влиять на среду функционирования уже запущенных программ (фактически это требование невозможности редактирования оперативной памяти).

Все прочие действия, являющиеся нарушением Условий 1-3, в оставшейся их части будут выявляться и блокироваться. Таким образом, ИПС существенно снижает требования к ПО в части наличия скрытых возможностей.

Основным элементом поддержания изолированности среды является контроль целостности. При этом возникает проблема чтения реальных данных, так как контроль целостности всегда сопряжен с чтением данных (по секторам, по файлам и т.д.). В процессе чтения РПВ может навязывать вместо одного сектора другой или редактировать непосредственно буфер памяти. С другой стороны, даже контроль самого BIOS может происходить «под наблюдением» какой-либо дополнительной программы («теневой BIOS») и не показать его изменения. Аналогичные эффекты могут возникать и при обработке файла. Таким образом, внедренное в систему РПВ может влиять на процесс чтения-записи данных на уровне файлов или на уровне секторов и предъявлять системе контроля некоторые другие вместо реально существующих данных. Этот механизм неоднократно реализовывался в STELTH-вирусах. Однако верно утверждение: если программный модуль, обслуживающий процесс чтения данных, не содержал РПВ и целостность его зафиксирована, то при его последующей неизменности чтение с использованием этого программного модуля будет чтением реальных данных. Из данного утверждения следует способ ступенчатого контроля.

Алгоритм ступенчатого контроля для создания ИПС на примере DOS

При включении питания ПЭВМ происходит тестирование ОП, инициализация таблицы прерываний и поиск расширений BIOS. При их наличии управление передается на них. После отработки расширений BIOS в память считывается первый сектор дискеты или винчестера (загрузчик) и управление передается на него, код загрузчика считывает драйверы DOS, далее выполня-

ются файлы конфигурации, подгружается командный интерпретатор и выполняется файл автозапуска.

С учетом этого механизма для реализации ИПС предварительно фиксируется неизменность программ в основном и расширенных BIOS, далее, используя функцию чтения в BIOS (для DOS int 13h), читаются программы обслуживания чтения (драйверы DOS), рассматриваемые как последовательность секторов, и фиксируется их целостность. Далее, используя уже файловые операции, читаются необходимые для контроля исполняемые модули (командный интерпретатор, драйверы дополнительных устройств, .EXE и .COM – модули и т.д.). При запуске ИПС таким же образом и в той же последовательности выполняется контроль целостности. Этот алгоритм можно обобщить на произвольную операционную среду. Для контроля данных на i-м логическом уровне их представления для чтения требуется использование предварительно проверенных на целостность процедур i-1-го уровня. В случае описанного механизма загрузки процесс аутентификации необходимо проводить в одном из расширений BIOS (чтобы минимизировать число ранее запущенных программ), а контроль запуска программ включать уже после загрузки DOS (иначе DOS определяет эту функцию на себя). При реализации ИПС на нее должна быть возложена функция контроля за запуском программ и контроля целостности.

Реализация ИПС с использованием механизма расширения BIOS

Рассмотрим 2 этапа – этап установки ИПС и этап эксплуатации ИПС. Предположим существование N пользователей, каждый i-й из которых характеризуется некоторой персональной информацией Ki, не известной другим пользователям и хранящейся на некотором материальном носителе (например, устройстве типа Touch Memory). Существует также администратор КС, который знает все Ki и единолично проводит этап установки. Пользователи же участвуют только в этапе эксплуатации.

Процесс установки ИПС состоит из следующих действий:

1. В ПЭВМ устанавливается плата, включающая в себя устройства и программы ПЗУ данного устройства, реализующие:

- чтение Ki;

- идентификацию пользователя с номером i по введенному Ki;

- чтение массива данных, содержащего множество доступных для выполнения пользователем i задач Pi1, Pi2, ..., Pim и информации Mi1, Mi2, ... Mim, фиксирующей целостность файлов Fi1, ..., Fim каждой задачи.

Описанное устройство должно активизироваться сразу после включения питания, отработки процедур самотестирования и инициализации системы прерываний.

Для ПЭВМ типа IBM PC для этой цели необходимо использовать механизм расширения BIOS.

2. Администратор определяет для пользователя i набор задач и соответствующих задачам исполняемых файлов {Pit, Fit}, t=1,...,mi; i=1, ..., N, где mi - число разрешенных к запуску задач для i-го пользователя.

3. Администратор формирует (и заносит на носитель) или считывает с носителя для i-го пользователя его Ki и вычисляет значения для последующего контроля целостности Mir=f(Ki, Fir, Pir), где f – функция фиксации целостности (хэш-функция).

4. Администратор проделывает действия 2 и 3 для всех N пользователей.

5. Администратор устанавливает в программную среду модуль активизации ИПС и фиксирует его целостность. Фиксируется также целостность файлов операционной среды Foc (в которые входят файлы DOS, драйверы и сетевое ПО).

Процесс эксплуатации состоит из следующих действий.

1. Включение питания и активизация расширенного BIOS:

а) идентификация пользователя по его Ki (при успехе п. б);

б) проверка целостности всех включенных в ПЭВМ BIOS (при положительном исходе п. в);

в) чтение по секторам файлов DOS и проверка их целостности;

г) чтение как файлов Рипс, так и сетевого ПО (с помощью функций операционной среды) и проверка его целостности;

д) активизация сетевого ПО;

е) активизация процесса контроля Рипс;

ж) запуск избранной задачи i-го пользователя.

2. Работа в ИПС.

Запуск каждого процесса Ps сопровождается проверками:

а) принадлежит ли Fs к множеству разрешенных для i (Ti), если да, то п. б), иначе запуск игнорируется;

б) совпадает ли G=f(Ki,Fs,Ps) с M=f(Ki,Fs,Ps), вычисленной администратором;

в) при положительном исходе проверки б) задача запускается.

Легко видеть, что условия изолированности среды (У1-5) в данном случае выполнены.

Пункт У1 гарантируется при установке системы администратором.

Пункты У2, У4 и У5 обеспечиваются платой (загрузка собственной DOS и сетевой среды с дискеты невозможна, поскольку расширенный BIOS активен раньше и направляет загрузку на винчестер; пользователь допускается к работе только при проверке Ki).

Пункт У3 реализован программным модулем контроля запусков и контроля целостности задач, входящим в состав ИПС. Кроме того, в данном случае реализован механизм ступенчатого контроля, обеспечивающий чтение реальных данных.

Глава 3. БЕЗОПАСНОЕ ВЗАИМОДЕЙСТВИЕ В КС

3.1 Введение

Создание защищенной КС может происходить с учетом существования механизмов безопасности, в связи с чем возникает задача сопряжения субъектов обеспечения ПБ и поддержания гарантий ПБ в различных элементах КС [5]. Для создания КС с гарантированно выполненной ПБ необходимо реализовать МБО, поддерживающий эту ПБ, и МБС, гарантирующий ПБ, и замкнуть каким-либо образом субъекты КС в ИПС. Практически в любой КС этого можно достигнуть, если субъекты (МБО и МБС) спроектированы и реализованы в виде исполняемых модулей конкретной операционной среды. Однако весьма часто в ОС уже присутствуют средства типа МБО с реализацией некоторой ПБ, несколько реже присутствуют и МБС. С точки зрения оптимизации трудозатрат на реализацию защитных механизмов целесообразно максимально использовать средства, которые уже реализованы в КС, в необходимых случаях усиливая и дополняя их.

В этом случае в рамках целостного представления о защите возникает проблема сопряжения штатных и дополненных средств защиты.

Поставленная проблема имеет ряд аспектов. Первый из них – техническое сопряжение субъектов ОС, осуществляющих штатную аутентификацию с дополняемыми субъектами, получившими информацию от модулей аутентификации. Второй аспект связан с гарантиями передачи параметров между модулями аутентификации и модулями реализации и гарантирования ПБ (МБО и МБС) без нарушения условий корректности субъектов. Третий – с организацией необходимых структур данных (описание объектов), обеспечивающих хранение данных для работы модулей аутентификации и сопряжения с другими объектами КС без нарушения ПБ.

Итак, рассмотрим вопросы:

1. Формализация процедур аутентификации пользователей КС и описание ее характеристик.

2. Формализация процедур сопряжения субъектов (для решения задач передачи параметров от модулей аутентификации к модулям реализации и поддержания ПБ).

3. Описание процедур сопряжения различных аутентифицирующих объектов.

4. Формализация и описание процедур использования внешних субъектов для поддержания защищенности КС.

5. Методика анализа попарной корректности субъектов.

3.2 Процедура идентификации и аутентификации

Учитывая, что пользователь КС только опосредованно работает с объектами и субъектами КС через средства ввода и отображения информации, постулируем наличие как минимум двух аутентифицирующих пользователя объектов – внешнего аутентифицирующего объекта, не принадлежащего КС, и внутреннего, принадлежащего КС, в который переносится информация из внешнего объекта [5]. Полагаем также наличие субъекта переноса информации от внешнего к внутреннему объекту (например, драйвер клавиатуры). Например, символьный пароль для входа в систему находится в «памяти пользователя», затем он путем набора на клавиатуре переносится в буфер программы запроса пароля (объект оперативной памяти ПЭВМ).

Поскольку предполагается выполнение процедур как идентификации, так и аутентификации, предположим, что i-й аутентифицирующий объект содержит два информационных поля IDi – неизменяемый идентификатор i-го пользователя, который является аналогом имени и используется для идентификации пользователя, и Ki – аутентифицирующая информация пользователя, которая может изменяться и служит для аутентификации.

Совокупную информацию в аутентифицирующем объекте будем называть *первичной аутентифицирующей информацией* i-го пользователя. Описанная структура соответствует практически любому устройству, служащему для опознания пользователя, например, упомянутый носитель сенсорной памяти типа Touch Memory (TM) имеет 8 байт неперезаписываемого неповторяющегося серийного номера, который однозначно характеризует конкретную TM и некоторый объект перезаписываемой памяти, ко-

торый может содержать Ki. Аналогично для носителей типа пластиковых карт выделяются неизменяемая информация первичной персонализации пользователя, соответствующая IDi, и объект в файловой структуре карты, содержащий Ki.

Очевидно, что внутренний аутентифицирующий объект не должен существовать в КС длительное время (больше времени работы конкретного пользователя). Далее, для постоянного хранения необходимо использовать некую преобразованную информацию от первичной информации. Рассмотрим типовые схемы аутентификации.

Схема 1

В КС выделяется объект следующей структуры – эталон для идентификации и аутентификации (предположим, что в системе зарегистрировано n пользователей).

В табл. 1 обозначено Ei=F(IDi,Ki), F – функция, для которой можно качественно описать свойство «невосстановимости» Ki по Ei и IDi.

«Невосстановимость» Ki описывается некоторой пороговой трудоемкостью To решения задачи восстановления аутентифицирующей информации по Ei и IDi, ниже которой не должна опускаться ни одна оценка трудоемкости нахождения Ki для всех известных алгоритмов данной задачи.

Таблица 1 – Объект-эталон для схемы 1

	Информация для идентификации	**Информация для аутентификации**
1	ID1	E1
2	ID2	E2
…	…	…
n	IDn	En

Алгоритм идентификации и аутентификации:

1. Пользователь предъявляет свой идентификатор (имя) ID.

2. Если ID не совпадает ни с одним Idi, зарегистрированным в КС, то идентификация отвергается – пользователь не допущен к работе, иначе идентификация проходит.

3. У пользователя субъектом аутентификации запрашивается аутентификатор K.

4. Субъектом аутентификации вычисляется Y=F(IDi,K).

5. Субъектом аутенфикации производится сравнение Ei и Y. При совпадении фиксируется событие «пользователь аутентифицирован», информация о пользователе передается в МБО, считываются необходимые для реализации заданной ПБ массивы данных, в противном случае аутентификация отвергается.

Схема 2

В КС выделяется объект следующей структуры (табл. 2).

Таблица 2 – Объект-эталон для схемы 2

	Информация для идентификации	**Информация для аутентификации**
1	ID1, S1	E1
2	ID2, S2	E2
…	…	…
n	IDn, Sn	En

В таблице обозначено Ei=F(Si, Ki),

Si – случайный вектор, заданный при создании пользователя, F – функция, для которой можно качественно описать свойство «невосстановимости» Ki по Ei и Si.

Алгоритм идентификации и аутентификации:

1. Пользователь предъявляет свой идентификатор (имя) ID.

2. Если ID не совпадает ни с одним IDi, зарегистрированным в КС, то идентификация отвергается.

3. По IDi выделяется Si.

4. У пользователя субъектом аутенфикации запрашивается аутентификатор K.

5. Субъектом аутентификации вычисляется Y=F(Si,K).

6. Субъектом аутентификации производится сравнение Ei и Y. При совпадении фиксируется факт успешной аутентификации, информация о пользователе передается в МБО, считываются необходимые для реализации заданной ПБ массивы данных.

Вторая схема защиты применяется в ОС Unix. В качестве идентификатора применяется имя пользователя (запрошенное по

Login), в качестве Ki – пароль пользователя (запрошенный по Password), функция F представляет собой алгоритм шифрования DES. Эталоны для идентификации и аутентификации содержатся в файле Etc/passwd.

Поскольку выбор аутентифицирующей информации происходит из небольшого множества осмысленных слов, то для повышения надежности защиты к реализованным защитным мерам операционной среды необходимо добавить аппаратную аутенфикацию.

3.3 Реализация механизмов безопасности на аппаратном уровне

3.3.1 Защита на уровне расширений Bios

Защита ресурсов ПЗВМ на аппаратном уровне может быть реализована с использованием механизмов расширений Bios. В ПЭВМ, реализованных на платформе Intel, первичная активизация вычислительных ресурсов компьютера производится кодом процессора, хранящемся в основном Bios. При включении питания код основного Bios «проецируется» в область памяти F000 и управление передается на точку входа, определенную производителем Bios. После этого код Bios производит тестирование оборудования, инициализацию векторов прерываний, активизацию видеосистемы и некоторые другие действия, зависящие от специфики Bios. В состав кода Bios входит типовая процедура поиска так называемых расширений Bios (Bios Extention). Расширение Bios – фрагмент исполняемого кода, оформленный по правилам, приводимым ниже, на который (в случае соблюдения этих правил) передается управление в ходе процедуры поиска расширений. Поиск расширений заключается в сканировании с шагом 512 байт области памяти с C000 до F000 с целью нахождения двухбайтовой сигнатуры 55AA. После нахождения этой сигнатуры анализируется следующий (третий, начиная с 55) байт, который указывает область расширения Bios в 512-байтных страницах (или блоках).

Если в указанной позиции находится число, отличное от 0, то вычисляется арифметическая байтовая контрольная сумма от

области памяти с байта 55 на длину, указанную в третьем байте. В случае совпадения этой суммы с 0 на четвертый (от первого байта 55) байт передается управление.

Если в теле кода, на который передано управление, окажется процедура RETF (с учетом состояния стека на момент вызова расширения), то произойдет возврат к основному Bios (т.е. к процедуре дальнейшего поиска расширений).

Таким образом, имеется механизм для реализации ряда защитных функций на аппаратном уровне ПЭВМ, т.е. на уровне, «хронологически» находящемся до уровня загрузки операционной системы. С учетом того, что объем расширения Bios не может быть очень большим, на этом уровне может быть реализован достаточно небольшой объем значимых для безопасности КС функций, а именно:

- идентификация и аутентификация пользователя (возможно, с использованием специфического аппаратного носителя;
- запрет несанкционированной загрузки ОС с избранных носителей (например с CD-ROM);
- контроль неизменности или целостности аппаратной или программной компоненты ПЭВМ.

Надо заметить, что первый расширенный Bios, код которого будет исполнен, – это расширение, проецируемое видеокартой (VideoBios). Оно размещено по адресу C000. Используя программы получения дампа памяти, можно убедиться в наличии указанных выше заголовков и команд.

Программирование пользователем расширенного Bios связано с решением ряда технических проблем. Первая из них связана с тем, что программирование в данном случае целесообразно на языке низкого уровня. Вторая связана с тем, что изменение состояний переменных программы при ее неизменном размещении в ПЗУ невозможно. Это предопределяет необходимость корректного перемещения кода в ОЗУ с передачей управления. Далее стоит упомянуть о том, что на этапе выполнения кода Bios доступен только ряд сервисных функций, которые могут быть использованы для программирования на низком уровне, – это сервис клавиатуры, реализованный в обработчиках 9h и 16h прерываний, видеосервис 10h прерывания и сервисы диска 13h прерывания.

Кроме того, корректное завершение фрагментов кода расширений представляет собой отдельную задачу. Как указывалось выше, возврат к выполнению основного Bios происходит по команде RETF. Однако если реализуемый пользователем код расширения содержит аварийные выходы (например, в случае неверной аутентификации пользователя), то корректное прерывание выполнения может быть выполнено через аппаратную перезагрузку компьютера.

Наконец, о том, каким образом можно реализовать расширения Bios. В настоящее время существует значительное количество сетевых карт с местом размещения ПЗУ или флеш, а также значительное число средств защиты (например, платы АККОРД), которые дают возможность перепрограммирования кода расширений Bios, в котором можно заложить необходимый механизм парольной идентификации и аутентификации пользователей (например, троекратный запрос пароля).

3.3.2 Защита на уровне загрузчиков операционной среды

Локализация защитных механизмов в структурах, связанных с организацией начальной загрузки операционных систем, позволяет решить ряд важных задач компьютерной безопасности. Это задачи, связанные с «ранней» идентификацией и аутентификацией пользователей (при отсутствии аппаратных средств защиты), защитой от несанкционированной загрузки операционной системы, а также получением специального вида загрузочных носителей.

Рассмотрим данные проблемы подробнее. Первая проблема возникает в том случае, когда процедуры идентификации и аутентификации не удается реализовать на этапе инициализации аппаратной компоненты компьютера (в частности, невозможно реализовать указанные процедуры в расширении Bios). При размещении процедур идентификации-аутентификации совместно с процедурами начальной загрузки удается выполнить идентификацию и аутентификацию на ранней стадии сеанса работы пользователя.

Вторая проблема связана с реализацией защиты от загрузки несанкционированных копий ОС. Для решения данной задачи

обычно используют тонкости обработки загрузки с внешних носителей либо преобразуют (например, шифруют) информацию на несъемных носителях компьютера.

В первом случае загрузка с внешних носителей операционной системы невозможна физически, во втором – даже при успешной загрузке с несанкционированной копии ОС информация недоступна.

Третья проблема связана с формированием загрузочных носителей (например, дискет), имеющих нестандартный вид, для их специального использования.

Решение указанных проблем сводится в общем случае к программированию модифицированного загрузчика (или загрузчиков) операционной системы. Для простоты рассмотрим технологию создания модифицированного загрузчика для гибкого магнитного диска.

Рассмотрим процесс загрузки ОС для компьютеров семейства Intel.

После выполнения всех процедур, реализованных в основном и расширенных Bios, считывается сектор с номером 1 с нулевой дорожкой поверхности чтения в дисководе А либо при его отсутствии – с дисковода 80h (в случае если в установке Setup установлена последовательность A:, C:). Считанный код размером 512 байт загружает его с адреса 0:7C00h в оперативную память, после чего управление передается на данный адрес. На дискете в этом месте находится программа начальной загрузки (BOOT-сектор), которая загружает в память драйверы DOS и передает им управление. На нулевой дорожке дискеты также находятся системные области File Allocation Table и Root Directory, которые формируют файловую структуру дискеты.

На жестком диске в первом секторе размещается Master Root Record, который адресует выполнение (по той же схеме) активного загрузчика операционной среды.

Таким образом, очевидно, что для модификации загрузчика необходимо в общем случае проделать следующие операции:

- заместить первичный код загрузчика собственным фрагментом;
- сохранить исходный код загрузочного сектора (в случае необходимости его выполнения);

➢ с учетом необходимости размещения первичного загрузчика по тому же адресу, что и модифицированного, обеспечить корректное перемещение модифицированного загрузчика в другую область памяти без потери управления.

Следовательно, необходимо разместить модифицированный загрузочный сектор на месте первичного (исходного) загрузчика и разместить первичный загрузчик (возможно, в преобразованном виде) в таком месте дискеты или жесткого диска, где будет обеспечена его гарантированная сохранность.

Для дискеты предлагается такой способ. Нулевая дорожка дискеты целиком копируется на место k-й дорожки. Исходная нулевая дорожка заполняется нулями (или модифицируется как-то иначе для получения нужного вида дискеты). На место загрузочного сектора устанавливается необходимая программа.

Предлагаемый способ позволяет исключить использование изготовленной дискеты без загрузки с нее. Дополнив DOS программами проверки целостности, можно добиться соблюдения всех требований изолированности программно-аппаратной среды.

3.4 Контроль и управление доступом

3.4.1 Произвольное управление доступом

Основной задачей контроля и управления доступом является ограничение операций, выполняемых зарегистрированными пользователями в системе [1]. Существует два основных механизма управления доступом – дискреционный (произвольный) и мандатный (нормативный). Требования различных стандартов информационной безопасности к управлению доступом рассмотрены в приложениях 1,2. Приведем анализ основных особенностей механизмов реализации этих требований в современных системах на примере ОС.

Требования произвольного управления доступом по «Оранжевой книге» появляются, начиная с класса С.

Основой реализации произвольного управления доступом является матрица прав доступа, строки которой соответствуют субъектам (пользователи, процессы), а столбцы – объектам (фай-

лы, каталоги и т.п.). В ячейках матрицы содержатся права доступа субъектов к объектам (рис. 3.1).

В зависимости от способа представления матрицы прав доступа в ОС различают несколько способов реализации произвольного доступа. Наиболее распространенными для ОС являются списки прав доступа, биты доступа, «парольная» защита.

«Парольная» защита осуществляется следующим образом: пользователь использует отдельный пароль для доступа к каждому объекту в системе.

Файлы → Пользователи ↓	F1	F2	F3	F4	F5
Лапин		R		R	
Котов	RW		R		
Волков		RW			
Майоров	C	C	C	C	C

R – права доступа пользователя по чтению;
W – права доступа пользователя по записи;
C – управление доступом для других пользователей.

Рис. 3.1

Использовние данного метода доставляет пользователю массу неудобств, связанных с запоминанием паролей.

В подавляющем большинстве современных защищенных ОС используются списки прав доступа (Access Control List – ACL) (рис. 3.2) и биты доступа (рис. 3.3).

При реализации ACL с каждым объектом ассоциируется список пользователей с указанием их прав доступа к объекту. При принятии решения о доступе, соответствующий объекту доступа ACL проверяется на наличие прав, ассоциированных с идентификатором пользователя, запрашивающего доступ, или его группы.

Основными недостатками метода являются большие временные затраты на обработку списков по сравнению с битами

защиты, а также необходимость разрешения противоречий между отдельными элементами списка.

Вследствие того, что многие ОС ведут свое происхождение от UNIX, они реализуют произвольный доступ с помощью механизмов битов защиты. При этом вместо списка пользователей, которым запрещен доступ к объекту, с объектом связываются биты защиты. Пример битов защиты приведен на рис. 3.3. В ОС UNIX биты защиты указывают права доступа для трех категорий пользователей: все пользователи (world), члены группы владельца (group) и владелец (owner). При этом биты защиты могут изменять только владелец объекта и администратор.

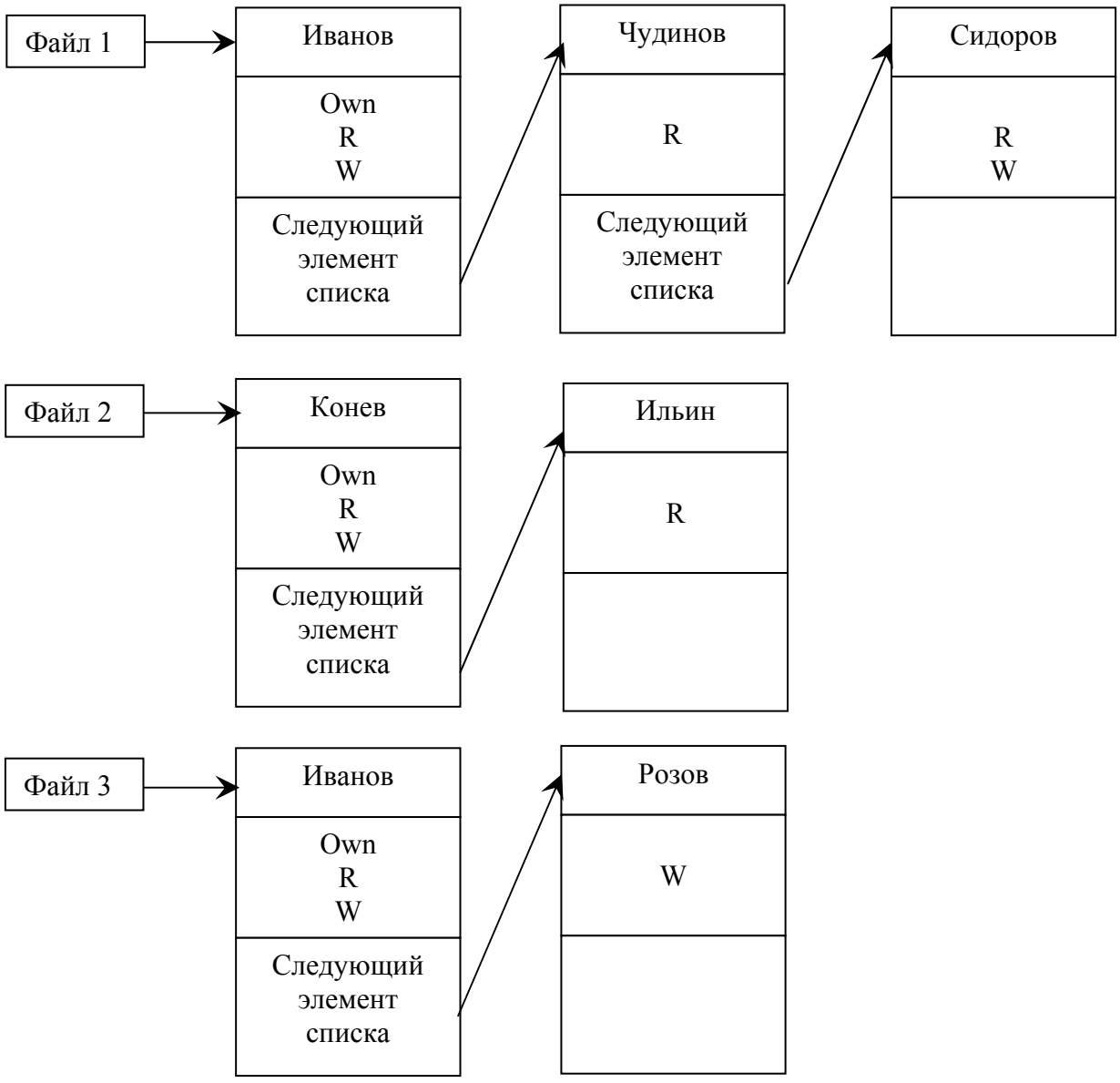


Рис. 3.2 – Пример списка контроля доступа к файлам

Рассмотрим типичный алгоритм проверки прав доступа к объекту при использовании механизма битов защиты. Субъект системы обычно ассоциирован с некоторым эффективным идентификатором (EUID), содержащим информацию о пользователе и группе, к которой он принадлежит.

При попытках доступа выполняются следующие действия:

1. Проверяется, является ли субъект собственником объекта. Для этого сравниваются значения эффективного идентификатора процесса и идентификатора владельца объекта. Если они равны, то сравниваются полномочия владельца объекта с запрашиваемым типом доступа. Если запрашиваемый тип доступа присутствует в соответствующем поле, то доступ предоставляется. Если идентификаторы не равны, то осуществляется переход ко второму шагу алгоритма.

2. Проверяется, входит ли субъект в группу владельца. Для этого сравниваются значения эффективного идентификатора процесса и идентификатора группы владельца объекта. Если они равны, то сравниваются полномочия группы владельца объекта с запрашиваемым типом доступа. Если запрашиваемый тип доступа присутствует в соответствующем поле, то доступ предоставляется. Если идентификаторы не равны, то осуществляется переход к третьему шагу алгоритма.

Владелец			Группа			Все пользователи		
Чтение	Запись	Выполн.	Чтение	Запись	Выполн.	Чтение	Запись	Выполн.
1	2	3	4	5	6	7	8	9

1…9 – номер бита

Рис. 3.3 – Биты защиты UNIX

3. В противном случае сравниваются полномочия, предоставленные всем пользователям системы с запрашиваемым типом доступа. Если запрашиваемый тип доступа присутствует в соответствующем поле, то доступ предоставляется.

Недостатком использования механизма битов защиты является неполная реализация произвольного контроля доступа, т.к.

доступ к объекту нельзя разрешить или запретить для отдельных пользователей.

В современных системах часто используются комбинации списков контроля доступа и битов защиты.

3.4.2 Нормативное управление доступом

В отличие от произвольного управления доступом, которое позволяет передавать права от одного пользователя другому, нормативное управление доступом полностью запрещает передачу прав доступа между пользователями. Это позволяет решить проблему «троянских коней» в защищенных информационных системах. Нормативное управление доступом основано на модели Белла-Лападула, которая описывает правила документооборота, принятые в правительственных учреждениях США.

Для защиты от угрозы целостности информации в защищенных информационных системах чаще всего используется модифицированный аналог модели Белла и ЛаПадула, так называемая нормативная модель целостности Биба. Нормативную модель целостности Биба часто называют инверсией модели Белла-Лападула. Это довольно точное название, поскольку основные правила этой модели просто переворачивают правила модели Белла-Лападула. Но при этом в модели рассматриваются уровни целостности, а не уровни безопасности.

В системах, не реализующих модель Биба, для защиты целостности доверенного программного обеспечения ТСВ системы (см. следующий параграф), в рамках нормативной модели Белла и ЛаПадула может использоваться следующий прием. ТСВ системы размещается на нижней ступени иерархии безопасности. Поскольку субъекты и объекты с высокой целостностью находятся внизу иерархии, а компоненты с низкой целостностью – наверху иерархии, то правила «запрет чтения с верхнего уровня» и «запрет записи на нижний уровень» имитируют нормативную модель целостности Биба в структуре модели Белла и ЛаПадула. Примеры реализации модели Белла и ЛаПадула приведены далее.

Пример реализации управления доступом и контроля за его осуществлением в операционной системе UTS MLS 2.1.5+.

Операционная система UTS MLS 2.1.5+ является доработкой традиционной UNIX до системы, защищенной по классу B. Управление доступом реализовано следующим образом. В описании объекта (в структуре i-node) операционной системы UTS MLS 2.1.5+ хранится информация о владельце объекта (его идентификатор – UID, указывающий на пользователя, создавшего объект), псевдогрупповая информация (Privs) и права доступа для всех пользователей (other) в системе (рис. 3.4).

Владелец	Псевдогрупповая информация	Все пользователи

Рис. 3.4 – Описатель объекта

Псевдогрупповая информация указывает на комбинацию группы субъекта с меткой безопасности нормативного контроля доступа. Стандартный механизм UNIX дает поддержку для произвольного разграничения доступа для групп (DGID). Однако в стандартной версии ОС UNIX отсутствует поддержка нормативного управления доступом. Для того чтобы обеспечить поддержку групп и нормативное управление доступом, а также совместимость с UNIX в UTS MLS 2.1.5+ вместо групп UNIX используется понятие псевдогрупповой информации, ассоциированной с каждым объектом и процессом.

Каждая комбинация группы и метки безопасности нормативного управления доступом является уникальной и отображается в единственную псевдогрупповую информацию. При этом меняется семантика поля GID. Вместо идентификатора группы в UTS MLS данное поле содержит псевдогрупповую информацию, указывающую на соответствующий элемент в таблице атрибутов безопасности (данная таблица поддерживается TCB ОС UTS). Каждый элемент таблицы атрибутов безопасности (рис. 3.5) содержит следующую информацию:

- Идентификатор псевдогрупповой информации в таблице (PTI).
- Группу пользователя.
- Метку безопасности нормативного управления доступом.

PTI	Метка нормативного контроля доступа		Группа (DGID)
	Уровень	Категория	

Рис. 3.5 – Формат записи таблицы атрибутов безопасности

Из приведенной таблицы атрибутов безопасности видно, что в ОС UTS метка нормативного контроля доступа является объединением иерархического (упорядоченного) уровня и иерархической категории. Данная пара определена для каждой метки в таблице атрибутов безопасности.

Нормативное управление доступом в ОС UTS позволяет использовать 256 упорядоченных уровней для задания уровня безопасности субъектов и объектов. Уровень 0 (SYSTEM) присвоен всем элементам, входящим в состав TCB, (что обеспечивает защиту его целостности), уровни 1-255 предназначены для пользователей. Всего поддерживается 1024 категории.

Рассмотрим алгоритм принятия решения о доступе (рис. 3.6).

1. Для нормативного контроля доступа псевдогрупповой идентификатор объекта (PTI) используется для получения информации об уровне безопасности объекта, извлекаемой из таблицы атрибутов безопасности. Данная информация сравнивается с помощью правил модели Белла-ЛаПадула с соответствующим уровнем безопасности субъекта, запрашивающего доступ. Кроме проверки соответствия уровня безопасности, система проверяет соответствие категорий в метке нормативного доступа.

Для произвольного контроля доступа прежде всего сравнивается эффективный идентификатор процесса (EUID), которым отмечается каждый процесс пользователя, с идентификатором владельца объекта (UID). Если они равны, то сравниваются полномочия владельца объекта с запрашиваемым типом доступа.

2. Если запрашиваемый тип доступа присутствует в соответствующем поле, то доступ предоставляется. Если идентификаторы не равны, то сравнивается информация о группе процесса с информацией о группе владельца (DGID), получаемой из таблицы атрибутов безопасности (Priv-таблицы). Если они равны, то сравниваются полномочия группы владельца объекта с запраши-

ваемым типом доступа. Если запрашиваемый тип доступа присутствует в соответствующем поле, то доступ предоставляется. Если нет, то доступ отклоняется.

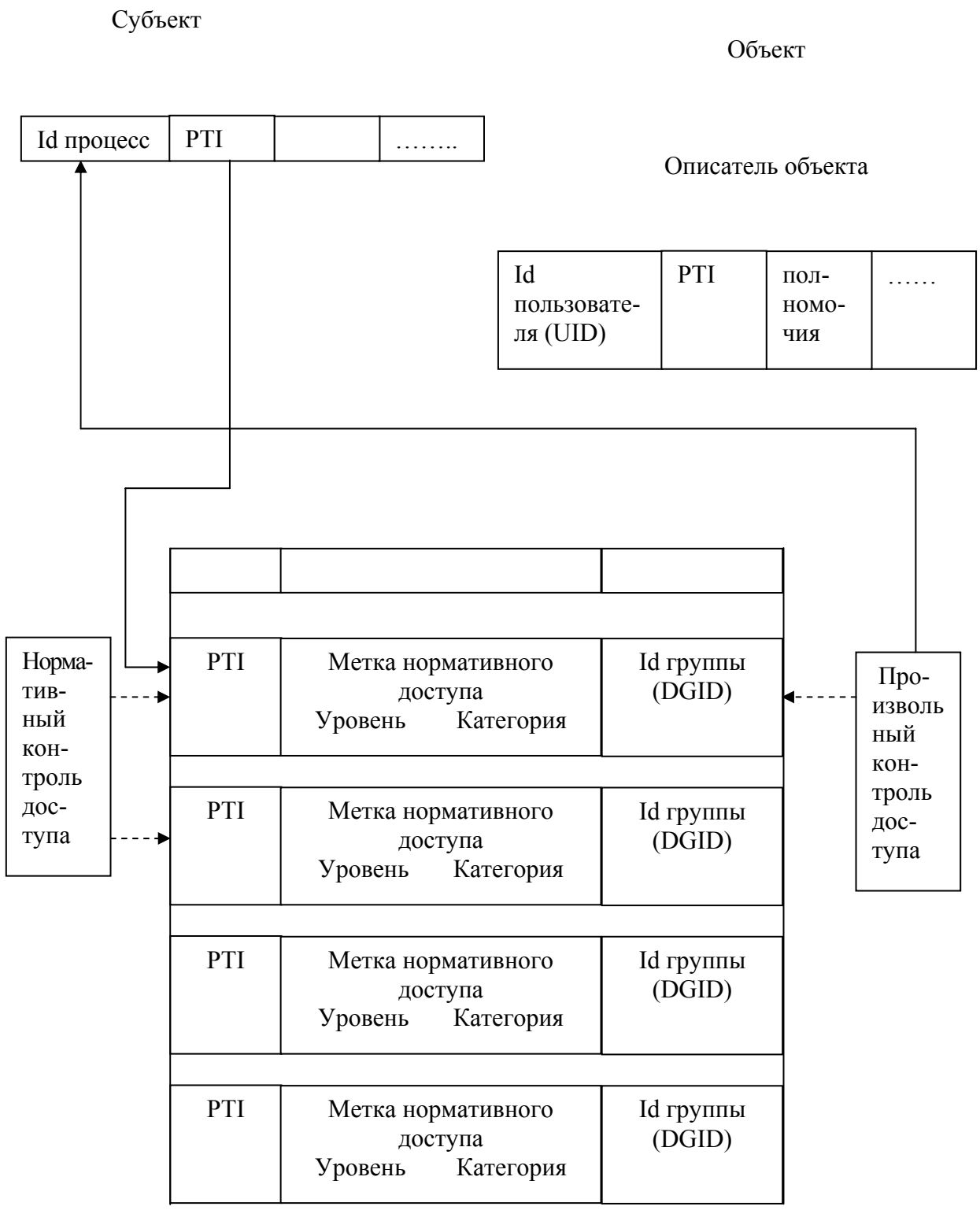


Рис. 3.6 – Механизм контроля доступа в UTS MLS 2.1.5+

Если и в данном случае совпадение отсутствует, то сравниваются полномочия, хранящиеся в поле «все пользователи» (other) с запрашиваемым типом доступа. Если запрашиваемый тип доступа присутствует в соответствующем поле, то доступ предоставляется. Если нет, то доступ отклоняется.

Субъекту предоставляется доступ к объекту только в том случае, если он одновременно разрешается механизмами нормативного и произвольного контроля доступа.

Диаграмма, характеризующая алгоритм принятия решения о доступе, приведена выше (рис. 3.6).

3.4.3 Диспетчер доступа комплекной системы защиты информации

Структура диспетчера доступа, реализованного в комплексной системе защиты информации (КСЗИ) «Панцирь» для ОС HP-UX 10.20, показана на рис. 3.7. Реализуются следующие механизмы управления доступом (каждый механизм реализуется своим диспетчером контроля доступа):

- авторизация пользователей и администратора безопасности осуществляется при доступе в систему (при запуске интерфейса настроек клиентской и серверной частей КСЗИ);
- разграничение прав произвольного доступа пользователей и процессов к объектам локальной файловой системы (томам, каталогам и файлам);
- разграничение прав произвольного доступа пользователей и процессов к разделяемым сетевым (по протоколу NFS) ресурсам файл-серверов локальной вычислительной сети – томам, каталогам, файлам;
- разграничение прав произвольного доступа пользователей и процессов к устройствам;
- разграничение прав произвольного доступа пользователей и процессов к сетевым ресурсам по протоколу TCP(UDP)/IP.

Верификация диспетчера доступа проводится в соответствии с требованиями к средствам вычислительной техники по 5 классу требований.

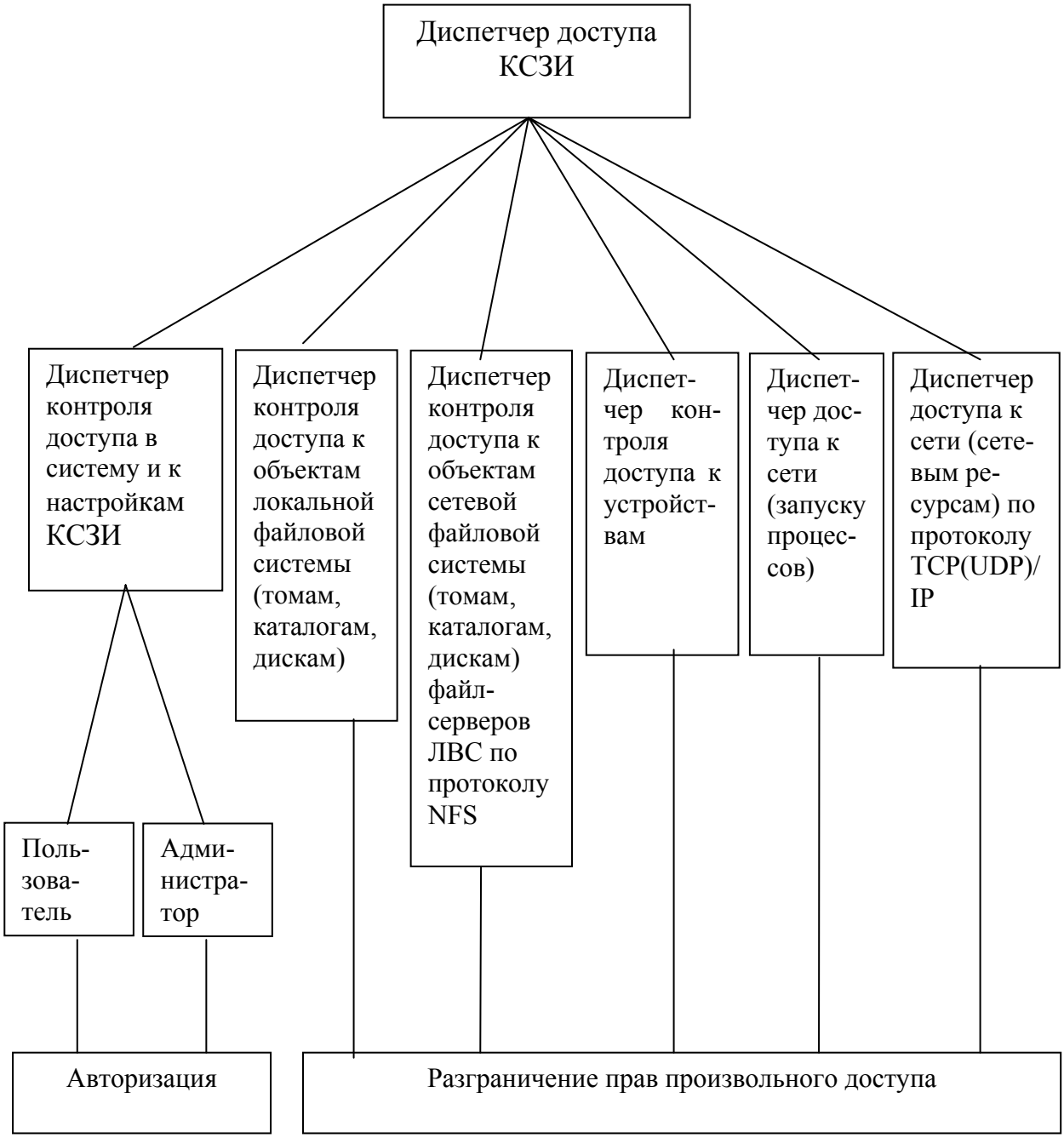


Рис. 3.7

Диспетчер доступа к системе обеспечивает идентификацию пользователей при запросах на доступ, проверяет подлинность идентификатора субъекта (осуществляет аутентификацию). Диспетчер доступа к системе располагает необходимыми данными для идентификации и аутентификации и препятствованию входа в систему неидентифицированного пользователя или поль-

зователя, чья подлинность при аутентификации не подтвердилась, а также обладает способностью надежно связывать полученную идентификацию со всеми действиями данного пользователя.

Диспетчер доступа к объектам файловой системы (к томам, каталогам, файлам) обеспечивает контроль доступа именованных субъектов (пользователей и процессов) к именованным объектам (файлам, каталогам, программам, томам). Для каждой пары (субъект-объект) в системе задано и недвусмысленное перечисление допустимых типов доступа (запись/чтение/исполнение/создание/удаление/переименование/изменение владельца/ изменение прав пользователя/создание ссылок на защищаемый объект), т.е. тех типов доступа, которые являются санкционированными для данного субъекта (индивида или группы индивидов, а также процессов) к данному ресурсу системы (к объекту).

КСЗИ содержит механизм, претворяющий в жизнь правила разграничения прав произвольного доступа на основе матрицы доступа. Контроль доступа применим к каждому объекту и каждому субъекту (индивиду или группе равноправных индивидов, процессов). Механизм, реализующий контроль произвольного доступа, предусматривает возможности санкционированного изменения правил разграничения доступа администратором безопасности после его авторизации. При этом предоставляется возможность сакционированного изменения списка пользователей системы, списка процессов (программ), имеющих доступ к объектам, и списка защищаемых объектов. Право изменять правила разграничения доступа предоставляется выделенным субъектам (администратору безопасности). Предусмотрены средства управления, ограничивающие распространение прав на доступ.

3.5 Безопасность компьютерной сети

3.5.1 Сканеры

Сканеры стали грозным оружием как нападения, так и защиты в Internet.

Сканер – это программа, предназначенная для автоматизации процесса поиска слабостей в защите компьютера, подклю-

ченного к сети в соответствии с протоколом TCP/IP [4]. Наиболее совершенные сканеры обращаются к портам TCP/IP удаленного компьютера и в деталях протоколируют отклик, который они получают от этого компьютера. Запустив сканер на своем компьютере, пользователь может найти бреши в защитных механизмах удаленного сервера.

Большинство сканеров предназначено для работы в среде операционной системы UNIX, хотя к настоящему времени такие программы имеются для любой ОС. Возможность запустить сканер на конкретном компьютере зависит от ОС, под управлением которой работает этот компьютер, и от параметров подключения к Internet. Есть сканеры, которые функционируют только в среде UNIX, а с остальными ОС оказываются несовместимыми. Другие отказываются нормально работать с устаревшими компьютерами и ОС и с медленным (до 14400 бит/с) доступом к Internet, осуществляемым по коммутируемым линиям. Такие компьютеры будут надоедать сообщениями о переполнении стека, нарушении прав доступа или станут просто зависать.

Критическим является и объем ОЗУ компьютера. Сканеры, которые управляются при помощи командной строки, как правило, более умерены в своих требованиях к объему ОЗУ. А самыми «прожорливыми» являются сканеры, снабженные оконным графическим интерфейсом пользователя.

Написать сканер не очень трудно. Для этого достаточно хорошо знать протоколы TCP/IP, уметь программировать на C или Perl и на языке сценариев, а также разбираться в программном обеспечении сокетов. Но в этом нет необходимости, так как предложения рынка сканеров превышают спрос на них.

Не следует переоценивать положительные результаты, которых можно достичь благодаря использованию сканера. Действительно, сканер может помочь выявить дыры в защите хост-машины, однако в большинстве случаев информацию о наличии этих дыр сканер выдает в довольно завуалированном виде, и ее надо еще уметь правильно интерпретировать. Сканеры не в состоянии сгенерировать пошаговый сценарий взлома исследуемой КС. Поэтому для эффективного использования сканеров на практике прежде всего необходимо научиться правильно интерпретировать собранные с их помощью данные, а это возможно только

при наличии глубоких знаний в области сетевой безопасности и богатого опыта.

Обычно сканеры создаются и используются специалистами в области сетевой безопасности. Как правило, они распространяются через сеть Internet, чтобы с их помощью системные администраторы могли проверять компьютерные сети на наличие в них изъянов. Поэтому обладание сканерами, равно как и их использование, вполне законно. Однако некоторые сканеры в процессе поиска брешей в защите сетей предпринимают такие действия, которые по закону могут квалифицироваться как несанкционированный доступ к информации, или как создание, использование и распространение вредоносных программ. И если следствием таких действий стало уничтожение, блокирование, модификация или копирование информации, хранящейся в электронном виде, то виновные в этом лица в соответствии с российским законодательством подлежат уголовному преследованию. А значит, прежде чем начать пользоваться первым попавшимся под руку бесплатным сканером для UNIX-платформ, стоит убедиться, не копирует ли случайно этот сканер заодно и какие-нибудь файлы с диска тестируемой им хост-машины (например, файл password из каталога /ETC).

3.5.2 Защита от анализаторов протоколов

В настоящее время технология построения компьютерных сетей Ethernet стала самым распространенным решением [4]. Сети Ethernet завоевали огромную популярность благодаря хорошей пропускной способности, простоте установки и приемлемой стоимости сетевого оборудования. Участки сети, для которых скорости передачи данных 10 Мбит/с недостаточно, можно легко модернизировать, чтобы повысить эту скорость до 100 Мбит/с или даже до 1 Гбит/с.

Однако технология Ethernet не лишена существенных недостатков. Основной из них – передаваемая информация не защищена. Компьютеры сети оказываются в состоянии перехватывать информацию, адресованную соседям. Основной причиной тому является принятый в сети Ethernet так называемый *широковещательный механизм обмена сообщениями*.

Компьютеры сети, как правило, совместно используют один и тот же кабель, который служит средой для пересылки сообщений между ними. Компьютер сети, желающий передать какое-либо сообщение по общему каналу, должен удостовериться, что этот канал в данный момент свободен. В начале передачи компьютер прослушивает несущую частоту сигнала, определяя, не произошло ли искажения сигнала в результате возникновения коллизий с другими компьютерами, которые ведут передачу одновременно с ним. При наличии коллизии компьютер прерывает передачу и «замолкает». По истечении некоторого случайного периода времени он пытается повторить передачу.

Если компьютер, подключенный к сети Ethernet, ничего не передает сам, он, тем не менее, продолжает «слушать» все сообщения, передаваемые другими компьютерами. Заметив в заголовке поступившей порции данных свой сетевой адрес, компьютер копирует эти данные в свою локальную память.

Существует два основных способа объединения компьютеров в сеть Ethernet. В первом случае компьютеры соединяются при помощи коаксиального кабеля. Кабель соединяется с сетевыми адаптерами Т-образным разъемом (сеть Ethernet 10Base2). В этой сети «все слышат всех». Любой компьютер способен перехватывать данные, посылаемые другим компьютером.

Во втором случае каждый компьютер соединен кабелем типа витая пара с отдельным портом центрального коммутирующего устройства – концентратора или коммутатора. В таких сетях, которые называются сетями Ethernet 10BaseT, компьютеры поделены на группы, именуемые доменами коллизий. Домены коллизий определяются портами концентратора или коммутатора, замкнутыми на общую шину. В результате коллизии возникают не между всеми компьютерами сети, а по отдельности – между теми из них, которые входят в один и тот же домен коллизий, что повышает пропускную способность сети.

В последнее время в крупных сетях стали появляться коммутаторы нового типа, которые не используют широковещание и не замыкают группы портов между собой. Вместо этого все передаваемые по сети данные буферизируются в памяти и отправляются по мере возможности.

Как уже отмечалось, сетевой адаптер каждого компьютера в сети Ethernet, как правило, «слышит» все, но обрабатывает и помещает в свою локальную память только те порции (так называемые кадры) данных, которые содержат его уникальный сетевой адрес. В дополнение к этому подавляющее большинство современных Ethernet-адаптеров допускают функционирование в особом режиме, называемом *беспорядочным*. При использовании данного режима адаптер копирует в локальную память компьютера все без исключения передаваемые по сети кадры данных.

Специализированные программы, переводящие сетевой адаптер в беспорядочный режим и собирающие весь трафик сети для последующего анализа, называются анализаторами протоколов. Администраторы сетей широко используют анализаторы протоколов для осуществления контроля за работой этих сетей и определения их перегруженных участков. К сожалению, анализаторы протоколов используются и злоумышленниками, которые с их помощью могут перехватить чужие пароли и другую конфиденциальную информацию.

Анализаторы протоколов представляют серьезную опасность. Присутствие их в сети указывает на брешь в защитных механизмах. Установить анализатор протоколов мог посторонний человек, который проник в сеть извне (например, если сеть имеет выход в Internet).

Специалисты в области компьютерной безопасности относят атаки на компьютеры при помощи анализаторов протоколов к так называемым атакам второго уровня. Это означает, что компьютерный взломщик уже сумел проникнуть в сеть и теперь стремится развить свой успех. При помощи анализатора протоколов он может попытаться перехватить регистрационные номера и пароли пользователей, их секретные финансовые данные. Имея в своем распоряжении достаточные ресурсы, компьютерный взломщик в принципе может перехватывать всю информацию, передаваемую по сети.

Анализаторы протоколов существуют для любой платформы. Они исследуют не конкретный компьютер, а протоколы. Поэтому анализатор протоколов может обосноваться в любом узле сети и оттуда перехватывать сетевой трафик.

В советах по защите от анализаторов протоколов нуждаются только те, кто желает дать отпор компьютерным взломщикам. В руках сетевого администратора анализатор протоколов является весьма полезным инструментом, помогающим находить и устранять неисправности, избавляться от узких мест, снижающих пропускную способность сети, и обнаруживать проникновение в нее компьютерных взломщиков.

Рекомендовать можно следующее:

- Обзаведитесь сетевым адаптером, который принципиально не может функционировать в беспорядочном режиме. Такие адаптеры существуют. Одни адаптеры не поддерживают беспорядочный режим на аппаратном уровне, а остальные просто снабжаются драйвером, не допускающим работу в беспорядочном режиме, хотя этот режим и реализован в них аппаратно.
- Учитывая, что спецификация PC99, подготовленная по инициативе корпораций Microsoft и Intel, требует безусловного наличия в сетевой карте беспорядочного режима, приобретите современный сетевой интеллектуальный коммутатор, который буферизует каждое отправляемое по сети сообщение в памяти и отправляет его по мере возможности точно по адресу. В результате надобность в «прослушивании» сетевым адаптером всего трафика для того, чтобы выбирать из него сообщения, адресатом которых является данный компьютер, отпадает.
- Не допускайте несанкционированной установки анализаторов протоколов на компьютеры сети. Для этого следует применять средства из арсенала борьбы с программными закладками.
- Шифруйте весь трафик сети.

3.5.3 Межсетевые экраны – эффективная технология сетевой защиты информации

Средства сетевой защиты в литературе фигурируют как firewall, брандмауэры и даже информационные мембраны. Но наиболее часто используется термин «межсетевые экраны» (МЭ). В общем случае для обеспечения сетевой защиты между двумя множествми информационных систем ставится экран или информационная мембрана, которые являются средством разграни-

чения доступа клиентов из одного множества систем к информации, хранящейся на серверах в другом множестве. В этом смысле МЭ можно представить как набор фильтров, пропускающих через себя весь трафик, анализирующих проходящую через них информацию и принимающих решение: пропустить информацию или ее заблокировать. Одновременно с этим производится регистрация событий и тревожная сигнализация в случае обнаружения угрозы.

Обычно экранирующие системы делают несимметричными. Для экранов определяются понятия «внутри» и «снаружи», причем в задачу экрана входит защита внутренней сети от потенциального враждебного окружения. Кроме того, МЭ может использоваться в качестве корпоративной открытой части сети, видимой со стороны Интернета. Так, например, во многих организациях МЭ используются для хранения данных с открытым доступом, как, например, информация о продуктах и услугах, файлах из баз FTP, сообщений об ошибках и т.п.

3.5.4 Современные требования к межсетевым экранам

1. Основное требование – это обеспечение безопасности внутренней (защищаемой) сети и полный контроль над внешними подключениями и сеансами связи.

2. Экранирующая система должна обладать мощными и гибкими средствами управления для простого и полного проведения в жизнь политики безопасности организации.

3. Межсетевой экран должен работать незаметно для пользователей локальной сети и не затруднять выполнение ими легальных действий.

4. Процессор МЭ должен быть быстродействующим, работать достаточно эффективно и успевать обрабатывать весь входящий и исходящий поток в пиковых режимах, чтобы его нельзя было блокировать большим количеством вызовов и нарушить его работу.

5. Система обеспечения безопасности должна быть надежно защищена от любых несанкционированных воздействий, поскольку она является ключом к конфиденциальной информации организации.

6. Система управления экранами должна иметь возможность централизованно обеспечивать проведение для удаленных филиалов единой политики безопасности.

7. Межсетевой экран должен иметь средства авторизации доступа пользователей через внешние подключения, что является необходимым в случае работы сотрудников организации в командировках.

3.6 Управление криптографическими ключами и хранение ключевой информации

3.6.1 Ключевая информация

Все криптографические системы основаны на использовании криптографических ключей. В симметричной криптосистеме отправитель и получатель используют один и тот же секретный ключ. Этот ключ должен периодически обновляться во избежание его компрометации. Процесс рассылки ключей между участниками информационного обмена в симметричных криптосистемах имеет сложный характер.

Несимметричная криптосистема использует два ключа – один открытый и второй секретный (личный). При обмене информацией необходимо пересылать только открытый ключ. Под ключевой информацией понимают совокупность всех действующих в автоматизированной системе обработки информации ключей [7].

В компьютерных системах должно быть налажено надежное управление ключевой информацией для предотвращения несанкционированного доступа к системе.

Управление ключами включает в себя следующие функции:

- генерация ключей;
- хранение ключей;
- распределение ключей.

Генерация ключей. Так как от криптографического ключа зависит безопасность криптографического алгоритма, то качественные криптографические ключи должны иметь достаточную длину и случайные значения битов.

Для генерации ключей используют аппаратные и прогаммные средства, способные формировать случайные значения ключей. В большинстве случаев применяют датчики псевдослучайных чисел (ПСЧ). Степень случайности должна быть высокой. Идеальными генераторами случайных чисел являются устройства, работающие на принципах естественных случайных процессов, например, на основе «белого шума».

В системах со средними требованиями защищенности допустимы программные генераторы ключей, которые формируют псевдослучайные числа как сложную функцию от текущего времени и (или) числа, введенного пользователем.

Для регулярной замены ключа необходимо проводить процедуру его модификации. Под модификацией ключа понимают генерирование нового ключа из предыдущего с помощью однонаправленной функции. Пользователи разделяют один и тот же ключ и одновременно вводят его значение в качестве аргумента в однонаправленную функцию, получая один и тот же результат. Затем они берут определенные биты из этих результатов и создают новое значение ключа.

Процедура модификации ключа обеспечивает его безопасность на уровне прежнего ключа. Но если злоумышленник завладеет прежним ключом, то он сможет его модифицировать.

Создание ключей для ассиметричных криптосистем с открытыми ключами намного сложнее.

Хранение ключевой информации. Под *хранением ключей* понимают организацию их безопасного хранения, учета и удаления. Злоумышленники проявляют особый интерес к ключам, так как ключи открывают доступ к конфиденциальной информации. Поэтому вопросам безопасного хранения ключей уделяется особое внимание. Секретные ключи никогда не должны записываться в явном виде на носителе, который может быть считан или скопирован.

Носители ключевой информации. Носитель, на котором записан ключ, может быть магнитным диском, устройством хранения ключей типа Touch Мемогу, пластиковой картой и т.д.

Применение магнитного диска (МД) в качестве носителя ключа позволяет временное изъятие МД из состава технических средств компьютерной системы. Особенно целесообразно исполь-

зование в качестве ключевых носителей съемных накопителей – гибких магнитных дисков, съемных магнитооптических носителей и т.д.

Основное преимущество МД по сравнению с другими носителями ключевой информации заключается в том, что оборудование для взаимодействия с МД (дисковод) входит в состав штатных средств компьютера. Другая важная особенность, определяющая широкое распространение МД, – стандартный формат хранения информации на дисках и стандартные программные средства доступа к дискам. Кроме того, из всех средств хранения ключевой информации гибкие магнитные диски имеют самую низкую стоимость.

Для обеспечения надежного хранения ключевой информации на МД применяют как минимум двукратное резервирование. Это обеспечивает защиту ключевой информации от ошибок при считывании с МД и от сбоев программной и аппаратной части.

Для исключения возможности перехвата ключевой информации в процессе ее чтения с МД используют хранение ключевой информации на МД в зашифрованном виде.

Устройство хранения ключей типа Touch Memory является относительно новым носителем ключевой информации. Носитель информации Touch Memory (ТМ) представляет собой энергонезависимую память, размещенную в металлическом корпусе, с одним сигнальным контактом и одним контактом земли. Корпус ТМ имеет диаметр 16,25 мм и толщину 3,1 или 5,89 мм (в зависимости от модификации прибора).

В структуру ТМ входят следующие основные блоки [7]:

• Постоянное запоминающее устройство (ПЗУ), которое хранит 64-разрядный код, состоящий из байтового кода типа прибора, 48-битового уникального серийного номера и 8-битовой контрольной суммы. Содержимое ПЗУ уникально и не может быть изменено в течение всего срока службы прибора.

• Оперативное запоминающее устройство (ОЗУ) емкостью от 128 до 8192 байт, которое содержит практически все модификации ТМ. В одной из модификаций оперативная память аппаратно защищена от несанкционированного доступа.

• Встроенная миниатюрная литиевая батарейка со сроком службы не менее 10 лет, которая обеспечивает питанием все блоки устройства.

Особенностью технологии хранения и обмена ключевой информации между носителем ТМ и внешними устройствами является сравнительно низкая скорость из-за последовательной передачи данных и высокая вероятность сбоя в тракте чтения-записи, обусловленная тем, что контакт устройства ТМ с устройством чтения осуществляется пользователем вручную без дополнительной фиксации (простое касание, что и определило название прибора ТМ – *память касания*). В связи с этим особое значение приобретают вопросы надежного обмена между программами обработки ключевой информации пользователей и носителем ТМ.

В устройстве ТМ конструктивно отработаны вопросы надежности функционирования и вопросы интерфейса со считывающим устройством на основе одного сигнального контакта. Для обеспечения достоверного чтения применяются корректирующие коды, для обеспечения достоверной записи в приборе предусмотрена технология буферизации. При проведении операции записи первоначально вектор передаваемой в ТМ информации помещается в буфер, далее выполняется операция чтения из буфера, затем прочтенная из буфера информация сравнивается с записываемой и в случае совпадения подается сигнал переноса информации из буфера в память долговременного хранения.

Таким образом, носитель ТМ является микроконтроллерным устройством без собственной вычислительной мощности и с ограниченным объемом хранимой информации, но с достаточно высокими надежностными характеристиками. Поэтому применение ТМ вполне обосновано в случае повышенных требований к надежности носителя ключа и небольшого объема ключевой информации, хранимой в ТМ.

Электронные пластиковые карты становятся в настоящее время наиболее распространенным и универсальным носителем конфиденциальной информации, который позволяет идентифицировать и аутентифицировать пользователей, хранить криптографические ключи, пароли и коды.

Интеллектуальные карты (смарт-карты), обладающие наибольшими возможностями, не только эффективно применяются для хранения ключевой информации, но и широко используются в электронных платежных системах, в комплексных решениях для медицины, транспорта, связи, образования и т.п.

3.6.2 Концепция иерархии ключей

Любая информация об используемых ключах должна быть защищена, в частности храниться в зашифрованном виде.

Необходимость в хранении и передаче ключей, зашифрованных с помощью других ключей, приводит к концепции *иерархии ключей.*. В стандарте ISO 8532 (Banking-Key Management) подробно изложен метод главных сеансовых ключей (master/session keys). Суть метода состоит в том, что вводится иерархия ключей: главный ключ (ГК), ключ шифрования ключей (КК), ключ шифрования данных (КД) [7].

Иерархия ключей может быть:

- двухуровневой (КК/КД);
- трехуровневой (ГКУКК/КД).

Самым нижним уровнем являются *рабочие или сеансовые КД*, которые применяются для шифрования данных, персональных идентификационных номеров (P1N) и аутентификации сообщений.

Для шифрования ключей с целью защиты при передаче или хранении, применяют ключи следующего уровня – *ключи шифрования ключей*, которые никогда не должны использоваться как сеансовые КД.

Таким образом, стандарт устанавливает шифрование различных типов ключей с помощью различных версий ключей шифрования ключей. В частности, ключи шифрования ключей, используемые для пересылки ключей между двумя узлами сети, называются *ключами обмена между узлами сети* (cross domain keys). В большинстве случаев в канале применяются два ключа для обмена между узлами сети, по одному в каждом направлении. Поэтому каждый узел сети будет иметь *ключ отправления* для обмена с узлами сети и *ключ получения* для каждого канала, поддерживаемого другим узлом сети.

На верхнем уровне иерархии ключей находится *главный ключ, мастер-ключ*, шифрующий КК, если требуется сохранить их на диске. Обычно в компьютере используется только один мастер-ключ. Для исключения перехвата мастер-ключ распространяется между участниками обмена неэлектронным способом. Значение мастер-ключа сохраняется длительное время (до нескольких недель или месяцев). Мастер-ключ компьютера создается случайным вы-

бором из всех возможных значений ключей и помещается в защищенный от считывания и записи блок криптографической системы. Должен существовать способ проверки на правильность значения ключа. Один из способов аутентификации мастер-ключа показан на рис. 3.7 [7].

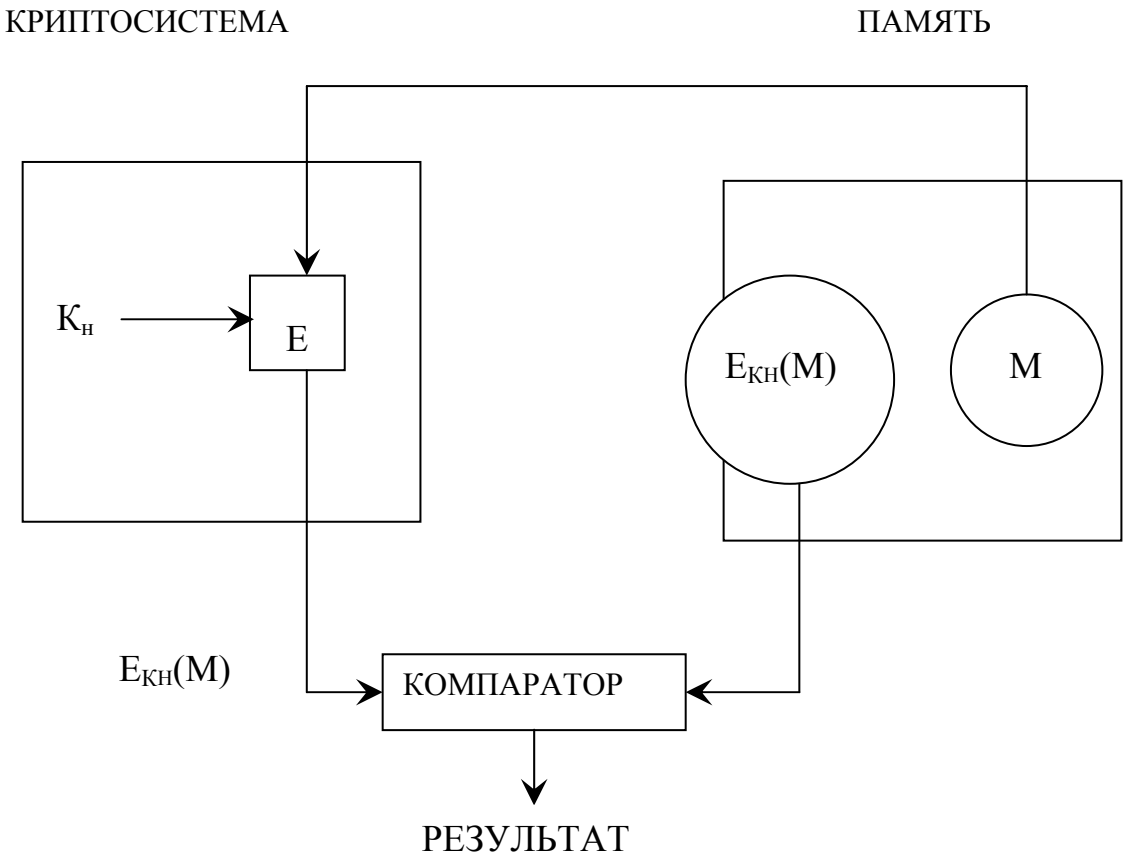


Рис. 3.7

Администратор, получив новое значение мастер-ключа $K_н$ хост-компьютера, шифрует некоторое сообщение M ключом $K_н$. Пара (криптограмма $E_{KH}(M)$, сообщение M) помещается в память компьютера. Когда требуется аутентификация мастер-ключа, берется сообщение M из памяти и подается в криптографическую систему. Полученная криптограмма сравнивается компаратором с криптограммой, хранящейся в памяти. При их совпадении считается, что ключ правильный.

Рабочие ключи (например, сеансовый) создаются с помощью генератора псевдослучайных чисел и могут храниться в незащищенном месте, поскольку такие ключи генерируются в форме крип-

тограмм (генератор выдает вместо ключа K_S его криптограмму $E_{KH}(K_S)$, получаемую с помощью мастер-ключа хост-компьютера). Расшифрование такой криптограммы осуществляется перед применением ключа K_S.

Схема защиты рабочего ключа показана на рис. 3.8. Чтобы зашифровать сообщение М ключом K_S, на соответствующие входы криптосистемы подаются криптограмма $Ек_н(K_5)$ и сообщение М. Криптографическая система сначала восстанавливает ключ K_5, а затем шифрует сообщение М, используя открытую форму сеансового ключа K_5.

Как видно, безопасность сеансовых ключей зависит от безопасности криптографической системы. Криптографический блок может быть спроектирован как единая СБИС и помещен в физически защищенное место.

Важным условием защиты информации является периодическое обновление ключевой информации в (АСОИ). При этом должны переназначаться как рабочие ключи, так и мастер-ключи. В особо ответственных АСОИ обновление ключевой информации (сеансовых ключей) желательно делать ежедневно. Вопрос обновления ключевой информации тесно связан с третьим элементом управления ключами – распределением ключей.

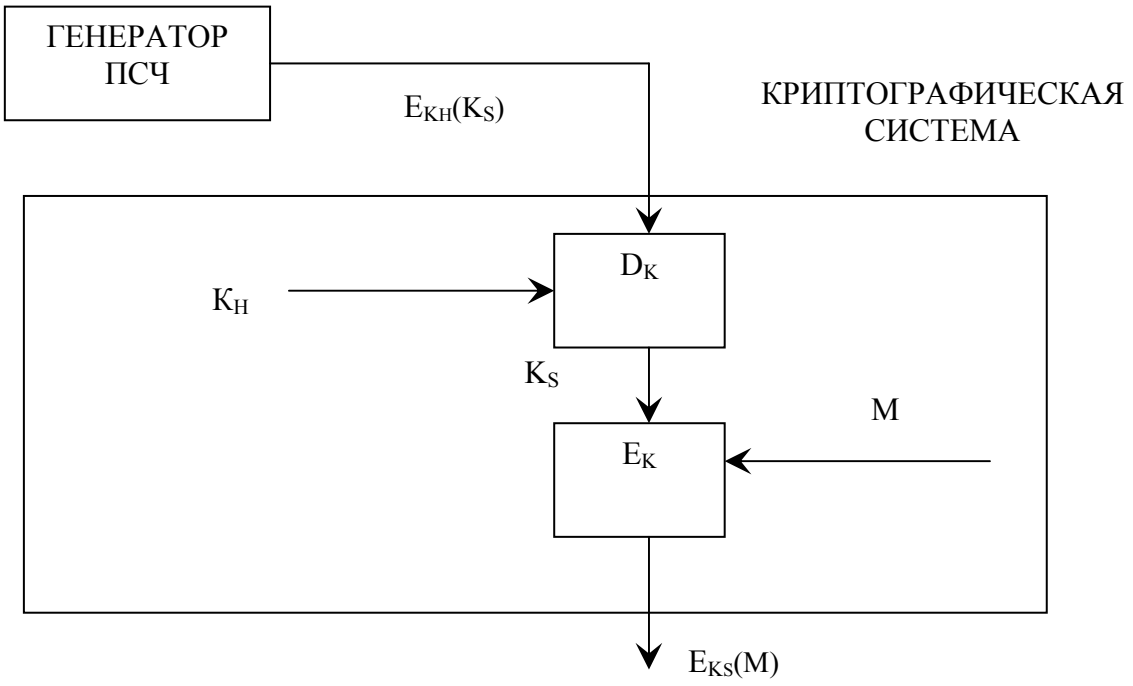


Рис. 3.8

3.6.3 Распределение ключей

Распределение ключей – самый ответственный процесс в управлении ключами. К нему предъявляются следующие требования:

- оперативность и точность распределения;
- скрытность распределяемых ключей.

Распределение ключей между пользователями компьютерной сети реализуется двумя способами [7]:

1) использованием одного или нескольких центров распределения ключей;

2) прямым обменом сеансовыми ключами между пользователями сети.

Недостаток первого подхода состоит в том, что центру распределения ключей известно, кому и какие ключи распределены, и это позволяет читать все сообщения, передаваемые по сети. Возможные злоупотребления существенно влияют на защиту. При втором подходе проблема состоит в том, чтобы надежно удостоверить подлинность субъектов сети.

В обоих случаях должна быть обеспечена подлинность сеанса связи. Это можно осуществить, используя механизм запроса-ответа или механизм отметки времени.

Механизм запроса-ответа заключается в следующем. Пользователь А включает в посылаемое сообщение (запрос) для пользователя В непредсказуемый элемент (например, случайное число). При ответе пользователь В должен выполнить некоторую операцию с этим элементом (например, добавить единицу), что невозможно осуществить заранее, поскольку неизвестно, какое случайное число придет в запросе. После получения результата действий пользователя В (ответ) пользователь А может быть уверен, что сеанс является подлинным.

Механизм отметки времени предполагает фиксацию времени для каждого сообщения. Это позволяет каждому субъекту сети определить, насколько старо пришедшее сообщение, и отвергнуть его, если появится сомнение в его подлинности. При использовании отметок времени необходимо установить допустимый временной интервал задержки.

В обоих случаях для защиты элемента контроля используют шифрование, чтобы быть уверенным, что ответ отправлен не злоумышленником и не изменен штемпель отметки времени.

Задача распределения ключей сводится к построению протокола распределения ключей, обеспечивающего:

• взаимное подтверждение подлинности участников сеанса;

• подтверждение достоверности сеанса механизмом запроса-ответа или отметки времени;

• использование минимального числа сообщений при обмене ключами;

• возможность исключения злоупотреблений со стороны центра распределения ключей (вплоть до отказа от него).

В основу решения задачи распределения ключей целесообразно положить принцип отделения процедуры подтверждения подлинности партнеров от процедуры собственно распределения ключей. Цель такого подхода состоит в создании метода, при котором после установления подлинности участники сами формируют сеансовый ключ без участия центра распределения ключей с тем, чтобы распределитель ключей не имел возможности выявить содержание сообщений.

3.6.4 Распределение ключей с участием центра распределения ключей

При распределении ключей между участниками предстоящего информационного обмена должна быть гарантирована подлинность сеанса связи. Для взаимной проверки подлинности партнеров приемлема *модель рукопожатия.* В этом случае ни один из участников не будет получать никакой секретной информации во время процедуры установления подлинности.

Взаимное установление подлинности гарантирует вызов нужного субъекта с высокой степенью уверенности, что связь установлена с требуемым адресатом и никаких попыток подмены не было. Реальная процедура организации соединения между участниками информационного обмена включает как этап распределения, так и этап подтверждения подлинности партнеров.

При включении в процесс распределения ключей центра распределения ключей (ЦРК) осуществляется его взаимодействие с одним или обоими участниками сеанса с целью распределения секретных или открытых ключей, предназначенных для использования в последующих сеансах связи.

Следующий этап – подтверждение подлинности участников. На этом этапе осуществляется обмен удостоверяющими сообщениями, чтобы иметь возможность выявить любую подмену или повтор одного из предыдущих вызовов.

Рассмотрим протоколы для симметричных криптосистем с секретными ключами и для асимметричных криптосистем с открытыми ключами. Вызывающий (исходный объект) обозначается через A, а вызываемый (объект назначения) – через B. Участники сеанса A и B имеют уникальные идентификаторы Id_A и Id_B соответственно.

Протокол аутентификации и распределения ключей для симметричных криптосистем. Рассмотрим в качестве примера протокол аутентификации и распределения ключей КегЬeros (по-русски – Цербер). Первоначально протокол КегЬeros был разработан в Массачусетском технологическом институте (США) для проекта Athena. Протокол КегЬeros спроектирован для работы в сетях TCP/IP и предполагает участие в аутентификации и распределении ключей третьей доверенной стороны. КегЬeros обеспечивает надежную аутентификацию в сети, разрешая законному пользователю доступ к различным машинам в сети. Протокол КегЬeros основывается на симметричной криптографии (реализован алгоритм DES, хотя возможно применение и других симметричных криптоалгоритмов). КегЬeros разделяет отдельный секретный ключ с каждым субъектом сети. Знание такого секретного ключа равносильно доказательству подлинности субъекта сети.

Основной протокол КегЬeros является вариантом протокола аутентификации и распределения ключей Нидхема-Шредера. В основном протоколе КегЬeros (версия 5) участвуют две взаимодействующие стороны A и B и доверенный сервер KS (КегЬeros Server). Стороны A и B, каждая по отдельности, разделяют свой секретный ключ с сервером KS. Доверенный сервер KS выполняет роль центра распределения ключей ЦРК.

Пусть сторона A хочет получить сеансовый ключ для информационного обмена со стороной B.

Сторона A инициирует фазу распределения ключей, посылая по сети серверу KS идентификаторы Id_A и Id_B:

$$A \rightarrow KS: Id_A, Id_B. \quad (1)$$

Сервер KS генерирует сообщение с временной отметкой T, сроком действия L, случайным сеансовым ключом K и идентифи-

катором Id_A. Он шифрует это сообщение секретным ключом, который разделяет со стороной B.

Затем сервер KS берет временную отметку T, срок действия L, сеансовый ключ K, идентификатор Id_B стороны B и шифрует все это секретным ключом, который разделяет со стороной A. Оба эти зашифрованные сообщения он отправляет стороне A:

$$KS \rightarrow A: E_A (T, L, K, Id_B), E_B (T, L, K, Id_A). \quad (2)$$

Сторона A расшифровывает первое сообщение своим секретным ключом, проверяет отметку времени T, чтобы убедиться, что это сообщение не является повторением предыдущей процедуры распределения ключей.

Затем сторона A генерирует сообщение со своим идентификатором Id_A и отметкой времени T, шифрует его сеансовым ключом K и отправляет стороне B. Кроме того, A отправляет для B сообщение от KS, зашифрованное ключом стороны B:

$$A \rightarrow B: Ек(Id_A,T), E_B(T, L, K, Id_A). \quad (3)$$

Только сторона B может расшифровать сообщения (3). Сторона B получает отметку времени T, срок действия L, сеансовый ключ K и идентификатор Id_A. Затем сторона B расшифровывает сеансовым ключом K вторую часть сообщения (3). Совпадение значений T и Id_A в двух частях сообщения подтверждают подлинность A по отношению к B.

Для взаимного подтверждения подлинности сторона B создает сообщение, состоящее из отметки времени T плюс 1, шифрует его ключом K и отправляет стороне A:

$$B \rightarrow A: E_к(T+1). \quad (4)$$

Если после расшифровки сообщения (4) сторона A получает ожидаемый результат, она знает, что на другом конце линии связи находится действительно B.

Этот протокол успешно работает при условии, что часы каждого участника синхронизированы с часами сервера KS. Следует отметить, что в этом протоколе необходим обмен с KS для получения сеансового ключа каждый раз, когда A желает установить связь с B. Протокол обеспечивает надежное соединение объектов A и B при условии, что ни один из ключей не скомпрометирован и сервер KS защищен.

Система КегЬегоs обеспечивает защиту сети от несанкционированного доступа, базируясь исключительно на программных ре-

шениях, и предполагает многократное шифрование передаваемой по сети управляющей информации.

Система КегЬегоs имеет структуру типа клиент-сервер и состоит из клиентских частей С, установленных на все машины сети (рабочие станции пользователей и серверы), и КегЬегоs-сервера KS, располагающегося на каком-либо (не обязательно выделенном) компьютере.

КегЬегоs-сервер, в свою очередь, можно разделить на две части: сервер (Ticket Granting Server) и сервер идентификации (AS). Информационными ресурсами, необходимыми клиентам С, управляет сервер информационных ресурсов RS (рис. 3.9).

Область действия системы КегЬегоs распространяется на тот участок сети, все пользователи которого зарегистрированы под своими именами и паролями в базе данных КегЬегоs-сервера.

Укрупненно процесс идентификации и аутентификации пользователя в системе КегЬегоs можно описать следующим образом. Пользователь (клиент) С, желая получить доступ к ресурсу сети, направляет запрос серверу идентификации AS. Последний идентифицирует пользователя с помощью его имени и пароля и выдает разрешение на доступ к серверу выдачи разрешений TGS, который, в свою очередь, по запросу клиента С разрешает использование необходимых ресурсов сети с помощью целевого сервера информационных ресурсов RS.

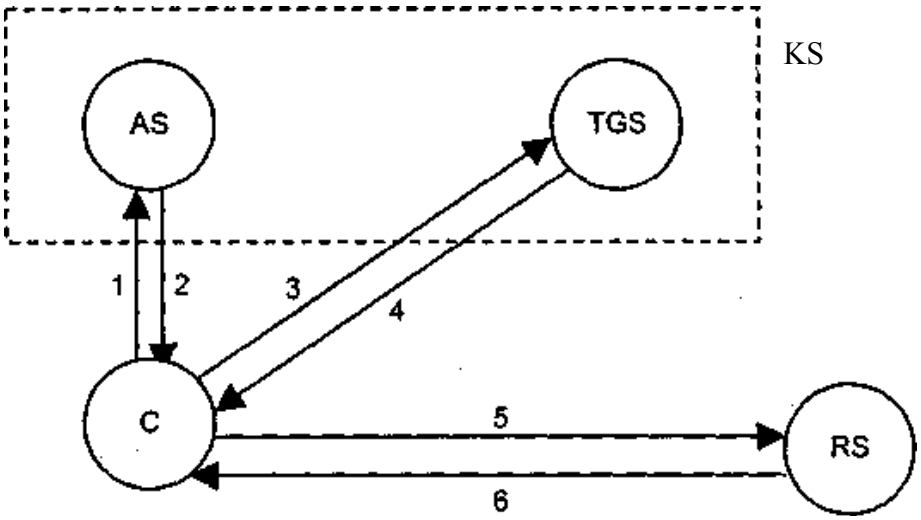
Данная модель взаимодействия клиента с серверами может функционировать только при условии обеспечения конфиденциальности и целостности передаваемой управляющей информации. Без строгого обеспечения информационной безопасности клиент не может отправлять серверам AS, TGS и RS свои запросы и получать разрешения на доступ к обслуживанию в сети. Чтобы избежать возможности перехвата и несанкционированного использования информации, КегЬегоs применяет при передаче любой управляющей информации в сети сложную систему многократного шифрования с использованием комплекса секретных ключей (секретный ключ клиента, секретный ключ сервера, секретные сеансовые ключи, клиент-сервер).

Протокол для асимметричных криптосистем с использованием сертификатов открытых ключей. В этом протоколе используется идея сертификатов открытых ключей.

Сертификатом открытого ключа С называется сообщение ЦРК, удостоверяющее целостность некоторого открытого ключа объекта. Например, сертификат открытого ключа для пользователя А, обозначаемый $С_А$, содержит отметку времени Т, идентификатор $Id_А$ и открытый ключ $К_А$, зашифрованные секретным ключом ЦРК $К_{ЦРК}$, т.е.

$$С_А = E_{КЦРК}(Т, Id_А, К_А).$$

Отметка времени Т используется для подтверждения актуальности сертификата и тем самым предотвращает повторы прежних сертификатов, которые содержат открытые ключи и для которых соответствующие секретные ключи несостоятельны.



Обозначения:

KS – сервер системы КегЬегоs;

AS – сервер идентификации;

TGS – сервер выдачи разрешений;

RS – сервер информационных ресурсов;

С – клиент системы КегЬегоs;

1: С→ AS: – запрос разрешить обратиться к TGS;

2: AS → С: – разрешение обратиться к TGS;

3: С →TGS: – запрос на допуск к RS;

4: TGS → С: – разрешение на допуск к RS;

5: С → RS: – запрос на получение информационного ресурса от RS;

6: : RS → С: – подтверждение подлинности сервера RS и предоставление информационного ресурса.

Рис. 3.9 – Схема и шаги протокола КегЬегоs

Секретный ключ $kц_{РК}$ известен только менеджеру ЦРК. Открытый ключ $Кц_{Р}к$ известен участникам А и В. ЦРК поддерживает таблицу открытых ключей всех объектов сети, которые он обслуживает.

Вызывающий объект А инициирует стадию установления ключа, запрашивая у ЦРК сертификат своего открытого ключа и открытого ключа участника В:

$$A \rightarrow \text{ЦРК: } Id_A, Id_B, \text{«Вышлите сертификаты ключей А и В'».} \quad (5)$$

Здесь Id_A и Id_B – уникальные идентификаторы соответственно участников А и В.

Менеджер ЦРК отвечает сообщением

$$\text{ЦРК} \rightarrow A: E_{КЦРК}(T, Id_A, KA), E_{КцРК}(T, Id_B, К_B). \quad (6)$$

Участник А, используя открытый ключ ЦРК $К_{ЦРК}$, расшифровывает ответ ЦРК, проверяет оба сертификата. Идентификатор Id_B убеждает А, что личность вызываемого участника правильно зафиксирована в ЦРК и $К_B$ – действительно открытый ключ участника В, поскольку оба зашифрованы ключом $К_{ЦРК}$.

Хотя открытые ключи предполагаются известными всем, посредничество ЦРК позволяет подтвердить их целостность. Без такого посредничества злоумышленник может снабдить А своим открытым ключом, который А будет считать ключом участника В. Затем злоумышленник может подменить собой В и установить связь с А, и его никто не сможет выявить.

Следующий шаг протокола включает установление связи А с В:

$$A \rightarrow B: C_A, E_{КА}(T), Ек_B(r_1). \quad (7)$$

Здесь C_A – сертификат открытого ключа пользователя А; $E_{КА}(T)$ – отметка времени, зашифрованная секретным ключом участника А и являющаяся подписью участника А, поскольку никто другой не может создать такую подпись; r_1 – случайное число, генерируемое А и используемое для обмена с В в ходе процедуры подлинности.

Если сертификат C_A и подпись А верны, то участник В уверен, что сообщение пришло от А. Часть сообщения $E_{КВ}(r_1)$ может расшифровать только В, поскольку никто другой не знает секретного ключа k_B, соответствующего открытому ключу $К_B$. Участник В расшифровывает значение числа r_1, чтобы подтвердить свою подлинность, посылает участнику А сообщение

$$B \rightarrow A: E_{КА}(r_1). \quad (8)$$

Участник А восстанавливает значение r_1 , расшифровывая это сообщение с использованием своего секретного ключа k_A. Если это ожидаемое значение r_1, то А получает подтверждение, что вызываемый участник действительно В.

Протокол, основанный на симметричном шифровании, функционирует быстрее, чем протокол, основанный на криптосистемах с открытыми ключами. Однако способность систем с открытыми ключами генерировать цифровые подписи, обеспечивающие различные функции защиты, компенсирует избыточность требуемых вычислений.

3.6.5 Прямой обмен ключами между пользователями

При использовании для информационного обмена криптосистемы с симметричным секретным ключом два пользователя, желающие обменяться криптографически защищенной информацией, должны обладать общим секретным ключом. Пользователи должны обменяться общим ключом по каналу связи безопасным образом. Если пользователи меняют ключ достаточно часто, то доставка ключа превращается в серьезную проблему.

Для решения этой проблемы можно применить два способа:

1) использование криптосистемы с открытым ключом для шифрования и передачи секретного ключа симметричной криптосистемы;

2) использование системы открытого распределения ключей Диффи-Хеллмана.

Второй способ основан на применении системы открытого распределения ключей. Эта система позволяет пользователям обмениваться ключами по незащищенным каналам связи. Интересно отметить, что система открытого распределения ключей базируется на тех же принципах, что и система шифрования с открытыми ключами.

Глава 4. ЗАЩИТА ОПЕРАЦИОННЫХ СИСТЕМ

4.1 Введение

В данной главе рассмотрены проблемы защиты программного обеспечения. Они охватывают широкий диапазон: от законодательных аспектов защиты интеллектуальной собственности до конкретных технических устройств. Последние помогают в тех ситуациях, когда очень трудно решить эти проблемы юридическими методами из-за высокой степени скрытности улики и несоизмеримости судебных издержек по сравнению с потерями от самих нарушений.

Установление авторского права на компьютерные программы изменило взгляд на защиту программных продуктов. Возникли две крайние точки зрения. Одну представляют изготовители программных продуктов, которые считают, что технические средства обеспечивают высокую степень защиты; другую – пользователи, для которых такие средства создают определенные трудности. Некоторые авторы считают, что авторское право должно быть единственным средством защиты.

Когда ряд фирм приняли авторское право в качестве единственного средства защиты программ, то оказалось коммерчески нецелесообразным продавать программы без соответствующей защиты на рынках тех стран мира, где авторское право не признается. Даже на территории стран, где соблюдается авторское право, необходима защита дешевых массовых программ; в противном случае продажа таких программ оказывается коммерчески невыгодной.

Кроме того, необходимо повысить степень скрытности защищаемой информации и методов обработки. В терминах информатики методы обработки, вероятно, должны быть реализованы в виде алгоритмов и соответствующих программных модулей, которые, хотя и являются объектом патентной защиты, если включены в состав вычислительной системы, тем не менее могут оказаться защищенными неадекватно, если содержат существенно новые элементы («ноу-хау»), которые представляют самостоятельный интерес для конкурентов.

Технические методы защиты, начиная от дешевых простых устройств и кончая мощными методами, включающими сложные способы шифрования, описаны ниже. Наша цель – рассмотреть все возможные виды компьютерного пиратства – от простых нарушений до злостных действий. Если для выявления нарушений требуются значительные усилия, это с большой вероятностью означает, что злоумышленник не является случайным лицом, однако обеспокоенность и постоянное чувство страха разоблачения делает его уязвимым по отношению к юридическим действиям, направленным на возмещение ущерба.

При чисто коммерческом отношении к оценке ущерба уменьшение числа посягательств на копирование программы, защищенной техническими средствами, должно быть сопоставлено с неприятием обычным покупателем того, что купленный товар имеет ограничения на его использование. Например, это может быть связано с тем, что пользователю запрещено создавать резервные копии программы; или один из портов компьютера должен быть выделен для подключения устройства защиты и тем самым ограничиваются его функциональные возможности; кроме того, защита может сделать программу более уязвимой и вызвать конфликт при взаимодействии с другими программами. Эти критические ситуации хорошо известны, и поэтому продолжаются поиски методов, которые могут удовлетворить таким противоречивым требованиям, как высокая степень защищенности и удобство применения. В идеале пользователь не должен «замечать» механизмов защиты, до тех пор пока он не попытается скопировать и выполнить программу на компьютере, на котором эти операции не разрешены.

Существует важное различие между методами защиты программного обеспечения и методами защиты компьютера и данных. Защита компьютера, основанная на использовании паролей и ограничении физического доступа к аппаратуре, совпадает с интересами законопослушного пользователя; в то же время, если программа успешно эксплуатируется и ее целостность не нарушена, пользователь не проявляет большой заинтересованности в защите прав автора программы.

Вычислительную установку разумно размещать на небольшой площади, и тогда упрощается контроль за ее защи-

той, а время и возможности на преодоление средств защиты оказываются весьма ограниченными. Программное обеспечение часто пересылается по почте, по каналам связи или просто продается в розницу, и это определяет способы его защиты. Программное средство, которое оказалось в руках пользователя, можно неограниченное время испытывать на преодоление механизмов защиты.

Когда говорят о превентивных мерах, обычно имеют в виду технические методы защиты, хотя и пассивные методы играют важную роль. Превентивные меры предполагают ограничения на использование программы, запрещение ее копирования или просмотра. Многие из них могут быть реализованы либо за счет создания условий защиты в самом компьютере, либо за счет включения защиты в программу, требующую, например, для своей работы разрешения от специального устройства. Однако такое устройство может оказаться экономически невыгодным из-за малого тиража его выпуска или из-за того, что его стоимость может увеличить существенно стоимость копии программы. Программа может быть сделана зависимой от характеристик конкретной вычислительной системы.

Другой важный механизм защиты – подтверждение подлинности программного кода. Он позволяет защитить от копирования, обеспечить конфиденциальность при выполнении финансовых сделок, в управленческой деятельности, при поддержании секретности, а также, например, в процессе автоматизированного проектирования, когда формальное доказательство целостности программы используется для разрешения входа в систему. Важно также обеспечить подтверждение подлинности и в тех случаях, когда в программу вносятся несущественные изменения, которые автор не в состоянии проконтролировать; это достигается тем, что такие изменения не учитываются. При этом программа должна сохранять совместимость с текущими версиями, имеющимися в продаже.

Предостережения обычно делаются с целью указать на неотвратимость возмещения ущерба по закону, но и простые словесные напоминания во многих случаях могут быть полезны. Лицензионная политика также позволяет достичь цели, но отношение к ней зависит от общественного мнения. Например, одноразовая продажа

программы с разрешением копирования может окончательно подорвать позицию конкурента. Некоторые пользователи считают, что они имеют право копировать другие программы в зависимости от того, получен или не получен автором гонорар при передаче программы в общественное пользование.

Одна из форм предостережения заключается в выводе на экран авторской этикетки (обращается внимание на юридическую ответственность). Аналогично включение в программу имени покупателя и вывод его на экран предостерегают покупателей от последующего копирования ее даже своим коллегам, поскольку имя покупателя будет присутствовать во всех копиях, которые будут сделаны и переданы другим лицам. Такие меры могут оказаться достаточными для законопослушного пользователя, но чтобы защититься от тех, кто не страдает угрызениями совести и способен выявить и уничтожить элементы защиты, в программу должны быть включены более мощные средства.

Пассивные методы могут во многих случаях помочь выявить улику, подтверждающую несанкционированное копирование, и это может сыграть свою роль в судебном разбирательстве. Однако следует определить цену такого пути, прежде чем последовать ему. Предпочтительно использовать улику, которую дают пассивные методы, для того чтобы вынудить компьютерного нарушителя (хакера), затеявшего игру, прекратить свою деятельность. Было бы достаточно, если бы нарушитель, которому предъявлены отличительные метки или другие улики в программе, прекратил дальнейшие попытки распространения и согласился возместить автору нанесенный ущерб.

Психология предостережения весьма важна при защите программного обеспечения. Хотя хакер может овладеть большим числом приемов раскрытия защиты, к услугам автора имеются разнообразные активные и пассивные методы защиты, использующие отличительные метки для ссылки на владельца авторского права или лицензию непосредственно в коде программы. Многообразие авторских приемов ставит компьютерного нарушителя в сложное положение, поскольку, хотя он и может выявить отдельные приемы, но не может быть абсолютно уверен, что все ключи и ловушки удалены из программы.

Для того чтобы обсуждение методов носило содержательный характер, эти методы классифицированы по категориям. Следует отметить, что использованные термины для обозначения категории не всегда связаны с устоявшейся терминологией.

Одно из направлений защиты – использование неожиданных или маловероятных приемов. Полезно уяснить, как можно комбинировать методы, чтобы создать прием, который из-за его неожиданности не может быть раскрыт хакером.

Выделены следующие категории средств защиты программного обеспечения:

- средства собственной защиты;
- средства защиты в составе вычислительной системы;
- средства защиты с запросом информации;
- средства активной защиты;
- средства пассивной защиты.

4.2 Средства собственной защиты

Собственная защита программ – это термин, определяющий те элементы защиты, которые присущи самому программному обеспечению или сопровождают его продажу и препятствуют незаконным действиям пользователя. Средства собственной защиты представлены на рис. 4.1.

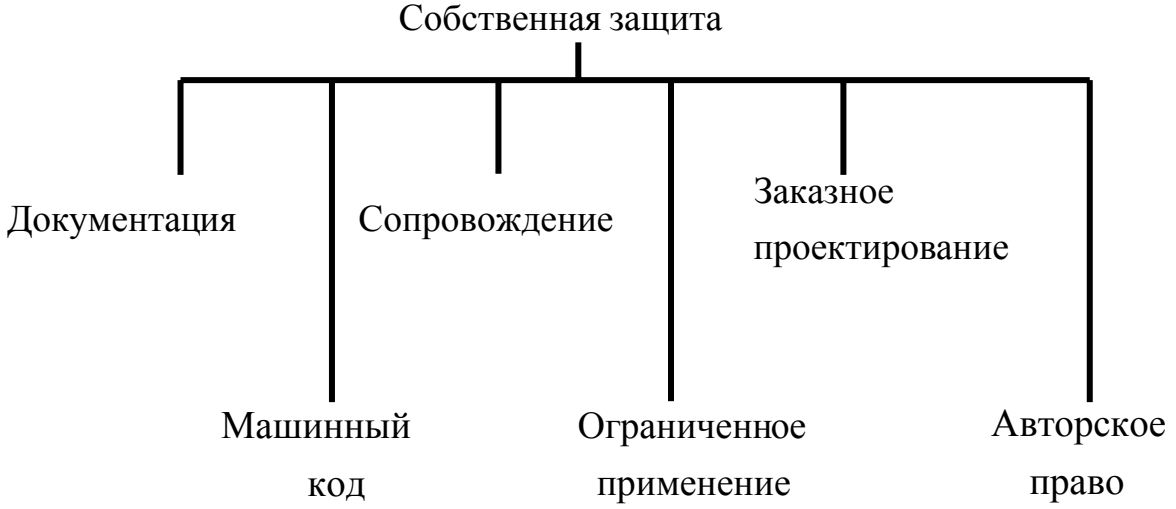


Рис. 4.1

Документация, сопровождающая любое программное обеспечение, является субъектом авторского права и может выполнять

функции защиты. Этому способствуют следующие факторы: ее репродуцирование стоит достаточно дорого, особенно если оригинал выполнен в цвете и не может быть качественно воспроизведен одноцветным копировальным устройством; обычно программы распространяются, будучи представленными в машинном коде, что затрудняет анализ их структуры и обеспечивает определенную степень защиты. В последнем случае весьма важно, чтобы сохранялось сопровождение программы со стороны разработчика, особенно в тех случаях, когда программа не полностью отлажена.

Ограниченное применение как способ защиты реализуется в том случае, когда программное обеспечение используется небольшим числом пользователей, каждый из которых известен по имени. Эта ситуация относительно легко контролируется в окружении, пользующемся доверием, хотя могут возникать проблемы с отдельными работниками, нанятыми на ограниченный срок. В этих случаях следует оговорить условия работы с программными средствами в заключаемом контракте.

Заказное проектирование предполагает разработку программного обеспечения для специальных целей. Если программа используется редко, то ее кража в коммерческих целях маловероятна; однако если кража произошла, то именно эти детали дают ключ к источнику несанкционированного копирования.

Рекомендуется также расставлять отличительные метки в стандартных программных модулях для того, чтобы идентифицировать программы, поставляемые добросовестным покупателям. Цена индивидуальной разметки каждой копии программы должна быть тщательно соразмерена с ожидаемой коммерческой прибылью.

4.3 Средства защиты в составе вычислительной системы

Эта категория средств защиты включает защиту дисков и аппаратуры, замки защиты, изменение функций штатных устройств. При использовании таких средств операционная среда вычислительной системы в отличие от штатного режима постоянно изменяется, поскольку выполнение программы зависит от определенных действий, специальных мер предосторожности и

условий, гарантирующих защиту. Перечень таких средств представлен на рис. 4.2.

4.3.1 Защита магнитных дисков

Методы защиты от копирования прежде всего были разработаны для IBM PC и подобных персональных компьютеров. Методы защиты гибких дисков используют два принципа: либо помешать копированию программы на другой диск, либо воспрепятствовать просмотру или операции обратного ассемблирования (реассемблирования).

О таких программах можно говорить как о «защищенных от копирования» и «защищенных от просмотра». Метод первого типа защищает программу от несанкционированного воспроизведения, а второго – от несанкционированной проверки.

Эти два типа защиты не взаимосвязаны: вполне возможно, что допускающая копирование программа защищена от просмотра и наоборот.

Весьма уязвимой является программа, размещенная на диске, защищенном от копирования, но допускающая просмотр. Если пользователь достаточно разобрался в программе и может обеспечить ее просмотр, то с большой долей вероятности он сможет найти способ переписать на незащищенный диск. В крайнем случае он может сделать распечатку и затем ввести программу заново. И наоборот, если программа не допускает просмотра, но находится на незащищенном диске, квалифицированный специалист может, используя стандартные средства проверки и изменения содержимого диска, обеспечить просмотр программы. По этой причине программные продукты обычно имеют оба вида защиты.

Защита дисков выполняется различными способами. Основная техника заключается в форматировании диска специальными способами, которые предохраняют операционную систему от копирования. Это – нестандартное определение форматов данных или каталогов, изменение размеров секторов, увеличение числа синхронизирующих битов и замена информационных заголовков. Поскольку время обращения к секторам различно, то программным способом можно определить время запаздывания

при чтении различных секторов, а поскольку при копировании с помощью стандартной DOS расположение секторов изменится, то запаздывания не будут более соответствовать запаздываниям исходной копии.

Рис. 4.2

Перечисленные методы становятся неэффективными при использовании систем побитового копирования. Побитовый копировщик – это электронная система копирования, которая осуществляет непосредственное считывание информации, бит за битом. Таким способом можно скопировать диск независимо от того, форматирован он или нет.

Средства защиты от побитовых копировщиков используют некоторые интересные приемы. Например, введение битов защиты, которые читаются по-разному в разное время и, таким образом, мешают верификатору копий; или запись исходной копии со скоростью ниже стандартной, что увеличивает плотность записи. В этом случае копирование на другой диск со стандартной скоростью вызывает увеличение длины записи и, следовательно, начало дорожки будет испорчено.

4.3.2 Защитные механизмы устройств вычислительной системы

Использование специальных характеристик аппаратуры для защиты программ всегда рассматривалось как весьма мощное, но дорогостоящее средство. Практический интерес представляет идея уникального диска, поскольку стоимость ее реализации достаточно низка. Принцип состоит в том, чтобы придать магнитной поверхности диска уникальность, запрещающую запись информации в некоторые секции дорожки. Сначала это достигалось механическим стиранием элементов поверхности, а позднее – использованием лазерного пучка. Таким образом, требуемый формат каждого диска оказывается уникальным. Это позволяет, сравнивая скорости чтения разных дисков, различать оригинальный диск от его копии.

В то время как специализированный компьютер экономически невыгоден для обеспечения защиты программ, специализированный микропроцессор на одном чипе оказывается пригодным для этих целей. Разработаны методы проектирования архитектуры микропроцессора, позволяющей защитить программу от считывания на шину данных для просмотра или копирования. Выполнение программы реализуется в чипе. В этом случае чип играет роль интеллектуального аппаратного модуля,

предназначенного для реализации наиболее важных процедур, требующих защиты, например, процедур шифрования и дешифрирования программ или данных, используемых исполняемой программой.

4.3.3 Замки защиты

Замки защиты используются для того, чтобы запретить доступ к программе, если при попытке обращения к ней не выполнены некоторые проверки. Например, можно осуществлять контроль предельного времени или даты использования согласно лицензии; при этом эталоном служат часы компьютера. Во многих системах установка часов доступна оператору, и это позволяет подстраивать часы под временной интервал действия лицензии. Защиту часов компьютера, т.е. предотвращение доступа к ним, можно выполнить, используя либо программный модуль, либо дополнительное оборудование в составе центрального процессора для контроля доступа к часам.

Контроль можно реализовать, если построить замок на основе уникального для каждого компьютера серийного номера. Такой замок защиты относят к классу «мобильных» (настраиваемых, уникальных) замков. В этом случае программа функционирует только на тех компьютерах, серийные номера которых включены в лицензию. Доступ к серийному номеру невозможен ни на одном компьютере, но хакеру это доставляет незначительные трудности – достаточно внести изменения в машинные коды объектного модуля, чтобы обойти или имитировать проверку.

Такая возможность обхода присуща многим методам контроля; данный недостаток можно устранить, если предусмотреть проверки в многочисленных и случайно назначаемых местах программы. Эти точки контроля должны быть проверены на подлинность перед переходом к счету, а еще лучше и во время счета, так чтобы никогда не возникали ситуации, когда вычисления идут без подтверждения проверки.

Интересный метод, который позволяет придать уникальную характеристику каждому компьютеру, состоит в записи с частичным разрушением памяти. Блоки динамической памяти в отличие от статической характеризуются тем, что данные в ней должны

периодически восстанавливаться путем регенерации. Данные исчезают, если регенерация, связанная с периодической перезаписью, по каким-либо причинам приостанавливается. Это свойство изменчивости структуры памяти можно использовать в целях идентификации. Если сигнал перезаписи прерывать на некоторое время, можно убедиться, что элементы памяти разрушаются специфическим образом для каждого конкретного модуля памяти. Условие функционирования программы можно связать с уникальной структурой памяти. Запись с частичным разрушением дает уникальный ключ защиты, который предохраняет программу от функционирования на другом компьютере.

Ключи защиты позволяют контролировать использование программного средства в течение заданных интервалов времени с последующим продолжением. Допустим, пользователь выплачивает периодическую (например, ежемесячную) арендную плату и получает на этот срок определенный пароль. Схема защиты запрещает доступ к программе, если пароль или ключ не будут соответствовать показаниям внутренних часов.

Другое применение таких ключей защиты заключается в том, что не период аренды, а ресурс программного продукта служит критерием. В этом случае необходимо определить единицу ресурса, которой может быть время функционирования программы в секундах или объем данных, извлеченных из базы данных. В этом методе ключевые слова присваиваются в соответствии с различными номерами устройств, так что пользователь может покупать блоки устройств в соответствии с работой, которая должна быть выполнена. В дальнейшем длительность использования может быть установлена заново и перезаписана в памяти.

4.3.4 Изменение функций

Существует ряд методов защиты, которые основаны на чередовании действий ключей или функций системы. Эти чередования могут предотвратить просмотр программного листинга или приостановить выполнение подпрограмм копирования. Изменения в визуальном представлении данных могут быть незаметными для пользователя, и он может не осознавать, что ин-

формация изменена или скрыта. Так, например, если изменить имена файлов, выводимых на экран монитора, то случайный пользователь не сможет вызвать такие файлы.

Функциональные особенности аппаратуры могут использоваться для защиты программ. Любая программа, размещенная в ПЗУ, будет отражать присущее этому устройству свойство, разрешающее только чтение информации. Попытки обойти это свойство программным способом будут безуспешны, если только не скопировать программу в память, допускающую запись, где незаконная копия может быть изменена.

4.4 Средства защиты с запросом информации

Включение защиты в программу связано с разработкой программ с запросом информации, т.е. требующих для своей работы ввода дополнительной информации, такой, как пароли, номера ключей и т.п.

4.4.1 Пароли

Обычные пароли не являются в полном смысле средствами защиты, они скорее относятся к механизмам управления доступом. Пароли обеспечивают сохранение целостности программного обеспечения в составе вычислительной системы, но для поддержания системы паролей требуется высокая дисциплинированность. Пароли должны быть просты для запоминания, чтобы не записывать их, и не должны быть столь очевидными, чтобы нарушитель мог угадать – они не должны быть связаны с адресом или названием фирмы. Вопросно-ответные системы, которые запрашивают место рождения, девичью фамилию и т.п., обеспечивают высокий уровень защиты, но требуют значительных ресурсов и времени работы вычислительной системы. Использование в качестве пароля отдельных элементов условного слова, например первой и пятой буквы, предотвращает ситуацию, когда целое слово могло бы быть случайно услышано. Так называемый одноразовый блокнот – более надежный механизм формирования паролей; в этом случае пароль является составным, наподобие листков блокнота, которые открываются одновременно (такой пароль

очень трудно определить). Однако защита с помощью пароля может оказаться неэффективной, если требуется хранение копии пароля, которая может быть похищена хакером.

Пароль в обычном смысле этого слова не является средством защиты программного обеспечения, поскольку законный пользователь, которому вручен пароль, может оказаться хакером. При формировании пароля можно прибегнуть к помощи специального устройства, которое генерирует последовательности чисел или букв в зависимости от данных, которые задает пользователь. Такое устройство называется преобразователем информации. В действительности вычислительная система генерирует последовательность случайных чисел и требует, чтобы пользователь в течение короткого промежутка времени присвоил ей некоторое число.

4.4.2 Сигнатуры

Сигнатура – уникальная характеристика компьютера или других устройств системы, которая может быть использована для защиты и проверена программным способом.

Уникальность гибких дисков проявляется прежде всего в форматировании. Уникальное форматирование позволяет закрепить за таким диском каталог файлов, требуемых для данной программы, чтобы установить нужную вычислительную среду. При этом копирование отдельных участков на ту же дискету гарантирует правильность, чего нельзя утверждать при копировании на другую дискету. Список испорченных секторов зависит от конкретной дискеты и отличается от списка для дубликата. К другим возможным сигнатурам относятся длина незаписанных участков магнитной ленты, неиспользованные дорожки на дискете и т.п. Техника частичного разрушения диска является примером, где сигнатура определяется уникальными характеристиками блока памяти компьютера.

В общем случае следует отыскивать такие характеристики аппаратуры или системы, которые не подвержены изменениям и сами не влияют на нормальное функционирование программного обеспечения. Если характеристики уникальны для данной вычис-

лительной системы, нормальное прохождение программы может быть выполнено только на ней.

4.4.3 Аппаратура защиты

Ожидалось, что производители компьютеров первыми проявят интерес к аппаратуре защиты и начнут встраивать ее в стандартные устройства. Но выяснилось, что никаких запросов на обеспечение компьютеров средствами защиты от пользователей не поступало. Другим потенциальным заказчиком должен быть изготовитель программных средств, особенно при продаже сложных и дорогих программных продуктов, когда защищенность прямо связана с получением дохода.

Принцип защиты программ с использованием аппаратуры защиты состоит в том, что при несанкционированном копировании программы из ПЗУ в оперативную память вырабатывается сигнал на самоуничтожение программы. Часть программного обеспечения обычно размещается в ПЗУ либо из-за недостатка памяти компьютера, либо из-за желания поставить под контроль операционной системы операцию копирования ПЗУ. Такая защита, возможно, и будет успешной от неискушенного нарушителя, но совершенно недостаточна для защиты от хакера.

Преобразователь информации, о котором мы кратко упомянули при обсуждении паролей, использует некоторые особенности преобразования данных. В одной из возможных реализаций преобразователя используется микропроцессор, генерирующий в соответствии с алгоритмом псевдослучайное число при нажатии некоторой клавиши клавиатуры. Если на вычислительной установке имеется такой же алгоритм, оператору достаточно задать правильное число, чтобы подтвердить требуемую последовательность.

Другие возможности были реализованы при использовании оптических устройств для выделения исходного образа из искаженных, которые поступают от компьютера. Это может выполнить только оператор, имеющий соответствующее оптическое устройство. Такое устройство можно построить на основе оптических материалов с двойным лучепреломлением, которые по-

зволяют получать наборы цветовых сигналов в соответствии с числами, вводимыми в устройство. Оптические системы, как правило, просты в изготовлении и имеют низкую стоимость. Достоинство описанных преобразователей информации состоит в том, что они независимы от назначения вычислительной системы и в контуре управления используют человека для оценки ответа от компьютера и задания уставок с помощью клавиатуры или другого устройства.

Электронные устройства защиты (ЭУЗ) обычно подсоединяются через стандартный интерфейс RS-232 и откликаются на запрос в виде некоторого числа или последовательности чисел. Недостаток этого устройства связан с тем, что необходимо управлять доступом к этому устройству из программы, и поэтому хакер может предусмотреть обход такого запроса. Обход может, например, заключаться в том, что включается обращение к подпрограмме, которая имитирует функцию ЭУЗ и затем возвращается к исполнению основной программы, обходя запрос. Можно также следить за линией связи и фиксировать числа, а затем генерировать таблицу для вторжения в программу. Для предотвращения таких попыток вторжения необходимо повторять запрос на доступ несколько раз и случайным образом. Кроме того, подлинность исходной программы должна подтверждаться в случайные моменты времени и предусматривать самоуничтожение программы при обнаружении обходов.

Устройства защиты с элементами интеллекта представляют собой одну из форм ЭУЗ с встроенным микропроцессором для реализации сложных алгоритмов защиты. В этом случае связь выхода ЭУЗ с входом оказывается непредсказуемой, обход устройства осуществить нельзя, если только этот обход не удалось встроить непосредственно в защищенную программу. Чтобы обойти эту защиту, надо скопировать программу из памяти микропроцессора. Действия против нарушителя – проектирование микропроцессора на кристалле со сверхвысокой степенью интеграции и встроенной памятью.

Альтернативные устройства используют генератор случайных чисел для реализации механизма подтверждения. Защищаемая программа сначала обращается к датчику случайных чисел для генерации некоторой случайной последовательно-

сти, которая и запоминается в программе; при подтверждении программа генерирует некоторую последовательность байтов, которая заставляет датчик случайных чисел воспроизвести последовательность, уникальную для данной программы.

Разработаны также устройства, которые обеспечивают защиту и подтверждение подлинности при взаимодействии двух терминалов. Такое устройство имеется на каждом терминале и предназначено для шифрования. Оно шифрует некоторое случайное число, сгенерированное на одном терминале, а использует его в диалоге с другим. На следующем шаге эта процедура осуществляется со второго терминала.

Средства непосредственной защиты основаны на уничтожении данных, если модуль, содержащий секретные данные, например ключи шифрования, подвергся взлому. Наиболее часто блокируется питание динамической памяти путем разрыва проводов, и информация исчезает. Возможные способы преодоления такой защиты засекречены. Очевидно, что эти методы широко используются в военных применениях, однако информация об этом недоступна для обычных коммерческих целей.

Программу, которую следует защитить, можно зашифровать, используя стандарт шифрования данных и зашифрованный открытый ключ. Особенность шифра открытого ключа состоит в том, что ключ к стандарту шифрования можно дешифрировать только секретным ключом, который находится в специальном аппаратном модуле, а защищенная программа может быть дешифрирована и выполнена только внутри этого модуля. Такие системы первоначально разрабатывались для перевода капиталов между банками и были реализованы в виде специализированной программы. Например, модуль защиты программного обеспечения, разработанный Национальной физической лабораторией, использует как симметричный, так и асимметричный шифры внутри модуля непосредственной защиты; это удобно при защите нескольких программ одним устройством.

Вероятно, неразумно ни с позиций функционирования, ни из экономических соображений размещать в устройстве защиты целые программы или пакеты программ. Должны быть зашифрованы лишь наиболее важные программные модули, которые будут выполняться внутри такого устройства, а вызываться

будут из незашифрованной главной программы, выполняющейся на центральном компьютере. Такое устройство может включить часы для контроля времени и даты и, поскольку оно защищено, такие часы могут фиксировать допустимую длительность работы, например, при аренде или пересылке программы пользователю для проведения вычислений в течение ограниченного времени.

Проблема, которую следует решить при выполнении нескольких программ в одном устройстве, – это отразить вторжение «троянского коня», т.е. включение в состав законной программы подпрограммы, которая оказалась бы вынужденной обратиться к данным, хранящимся в памяти микропроцессора. Новшество, предложенное в одном из патентов, состоит в том, чтобы в защищенную область памяти нельзя было обратиться, не вызвав сброса микропроцессора. В результате перезагрузки микропроцессор стартует заново и, следовательно, оказывается под управлением защищенной программы, размещенной в новой области памяти.

Следует отметить, что защита системы с открытым ключом требует сохранения регистра законных ключей с целью воспрепятствовать возможному появлению парного ключа. Если этого не предусмотрено, зашифрованное программное обеспечение может быть дешифрировано оставшимся ключом, принадлежащим нарушителю. Почему же регистр хранения открытого ключа обеспечивает необходимую защиту? Дело в том, что длина этих чисел, будучи порядка 150 десятичных цифр, позволяет сгенерировать невероятно большое число таких ключей. Число законных пар, размещаемых в регистре, существенно меньше и вероятность того, что нарушитель подберет пару, которая соответствует регистру защиты, чрезвычайно мала.

Специальная архитектура плат со сверхвысокой степенью интеграции позволяет защитить память, содержащую программу, от чтения через шину данных. Память для хранения программ – это либо программируемое (ППЗУ), либо стираемое программируемое постоянное запоминающее устройство (СППЗУ), которое размещено отдельно от памяти данных и недоступно со стороны входных и выходных портов (чтобы предотвратить прямой доступ к содержимому памяти). Плата

помещается в резервные гнезда компьютера. Считается, что преодоление такой защиты по трудоемкости аналогично технологии воспроизведения архитектуры микропроцессора с помощью шлифовки платы, как это делается при расслоении интегральной микросхемы.

Степень защиты, обеспечиваемая рядом описанных выше подсистем, поднимает интересные проблемы о потенциальных возможностях такой защиты. Секретный ключ обычно можно снова восстановить из записей и, вероятно, он будет разглашен при судебном разбирательстве. Однако можно организовать модуль так, чтобы секретный ключ порождался генератором случайных чисел в обстановке секретности и помещался в защищаемый модуль без участия человека и знания ключа. В этом случае невозможно проверить содержимое модуля. То же справедливо и по отношению к архитектуре специального микропроцессора.

4.5 Средства активной защиты

Средства активной защиты делятся на две группы: внутренние и внешние, используемые в составе компьютера и вне его соответственно. Средства защиты инициируются при возникновении особых обстоятельств – вводе неправильного пароля, указания неправильной даты или времени при запуске программы на выполнение или других подобных условий. Попытки получить доступ к точной информации без разрешения на это могут также служить инициирующим обстоятельством для приведения защиты в действие. Внутренние средства активной защиты характеризуются тем, что их обычно не рекламируют хакерам: они либо блокируют программу, либо уничтожают ее.

4.5.1 Внутренние средства активной защиты

Ключи защиты для блокирования выполнения программы могут быть настроены на любое недозволенное действие, которое будет обнаружено. Обычно это ключи, настроенные на дату, определенное время или на перечень разрешенных ресурсов; в наибольшей степени это относится к арендуемым лицен-

зионным программам, для которых период использования и требуемые ресурсы бывают однозначно определены. Проверка уровня авторских полномочий необходима, чтобы блокировать доступ к точной информации или другим ресурсам, запрещенным для использования. Реакция на несанкционированный доступ может быть реализована в виде предупреждения, дружеского напоминания либо служить поводом для организации наблюдения.

Инициализация наблюдения может начаться с регистрации в системном журнале использования терминала или реализовываться в виде подтверждения подлинности структуры программы; следует проверить, не подверглись ли средства защиты, включенные в программу, изменению или удалению.

Искажение программы представляет собой интересный прием изменения функций, хотя возможны и более решительные действия, например стирание памяти. Например, программы-вирусы вызывают постепенное разрушение программы.

4.5.2 Внешние средства активной защиты

В группе этих средств общепринятые сигналы тревоги, которые известны или неизвестны хакеру, приводят в состояние готовности средства защиты, что может быть вызвано различными ситуациями. Они могут быть активизированы при возникновении многих условий, уже описанных выше. Такие внешние факторы включают также использование ключевых слов, чтобы вызвать распечатку названия программы или имени ее владельца.

Распечатка авторской этикетки важна, поскольку большинство людей считают, что они действуют законно, и напоминание им о праве собственности владельца вызывает у них некоторую обеспокоенность. Хотя описанное и не является защитой от пиратства, это тем не менее способствует увеличению объема продаж, если число случайных копирований уменьшится. Общепринятые сигналы тревоги более сродни созданию среды защиты компьютера, когда требуется подтверждение подлинности операции, особенно при копировании.

Замечено, что чувство беспокойства может оказаться эффективным механизмом сокращения активности по крайней мере не-

которых хакеров. Законность совместного (коллективного) пользования программами должна быть подтверждена приобретением лицензии, что может служить эффективным методом борьбы с нарушителями авторского права.

Запуск распечатки этикетки или других деталей из защищенных участков программы осуществляется только при наличии ключевых слов. В то время как этикетка, появившаяся на листинге, может быть вырезана из него, защищенные данные, о которых мы еще будем говорить при обсуждении методов пассивной защиты не так просто, во-первых, найти, а во-вторых, декодировать для получения распечатки, используя подпрограмму, которая инициируется при вводе нужного ключевого слова или другой операцией (активная защита). Метод применим и в иных случаях, не обязательно связанных с анализом программного листинга.

4.6 Средства пассивной защиты

К средствам пассивной защиты относятся предостережения, контроль, а также методы, направленные на поиск улик и доказательство копирования, чтобы создать обстановку неотвратимости раскрытия.

4.6.1 Идентификация программ

Идентификация программы или отдельного модуля представляет интерес в том случае, когда другие методы защиты не приносят успех. Эти вопросы слабо освещены в литературе, за исключением обсуждения нескольких программных процедур и ряда отчетов о судебных тяжбах в США. 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. Оценка относительной частоты появления операторов или машинных команд – практический способ количественной оценки характеристики программы. Эта величина изменяется при внесении хакером изменений в программу, однако в большой программе для существенного изменения характеристики требуется выполнить значительную работу; к этому следует добавить возможность появления дополнительных ошибок или не согласующихся процедур, которые уменьшают надежность программы.

Для получения корреляционных характеристик, связанных с вставкой программного модуля в большую программу, требуются трудоемкие расчеты, хотя можно указать ряд важных признаков, которые указывали бы на целесообразность более детальных исследований.

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.

Каждое из них может служить убедительной уликой нарушения авторского права. Наоборот, отличительные метки относятся к таким признакам, которые не являются случайными, а вводятся специально, чтобы дать информацию об авторе или владельце авторского права. Другое использование идентификационных меток – выявление путей незаконного копирования или других злоумышленных действий. Термин «отличительная метка» относится к пассивным средствам защиты, которые при нормальном функционировании не «проявляют» себя по отношению к пользователю.

Одно из убедительных доказательств копирования – наличие скопированных ошибок. Маловероятно, чтобы в точном аналоге, который создан, как утверждается, независимо, содержались те же ошибки. В каждой программе остаются избыточные части, например подпрограммы, которые были необходимы для отладки в процессе проектирования программного продукта, а затем не были удалены. Таким образом, в любой программе содержится встроенная улика, которая тем или иным способом сохраняет следы разработки. Отсюда вытекает практический совет – сохранять документацию, которая сопровож-

дала процесс проектирования, чтобы потом иметь улику, подтверждающую авторское право.

Существует точка зрения, что убедительность улики повышается, если отличительная метка, содержащая информацию о владельце авторского права, закодирована. Известно много способов включения такой улики, особенно в программы на языках высокого уровня. Диапазон возможностей сокращается, если необходимо, чтобы отличительная метка сохранялась после компиляции в машинном коде программы. Эта проблема особенно актуальна при использовании оптимизирующих трансляторов. Использование закодированных отличительных меток – довольно распространенная практика, поскольку при этом они остаются доступными и в машинном коде. Существенно, что отличительные метки не являются в полной мере избыточными для того, кто организует контроль за данными и в состоянии отделить на их фоне действительно избыточные данные.

Очевидно, что можно разработать методы, которые позволят использовать закодированные в программе данные. Один из них связан с форматированием выходных данных в закодированной форме, что обусловлено необходимостью проверки кодирования при удаленной передаче, и это требует дополнительной работы.

Важная особенность отличительных меток заключается в том, что они неизвестны нарушителю. Поскольку в прошлом на программы покушались в основном бывшие служащие фирмы, существует организационная проблема, связанная с тем, чтобы отличительные метки были неизвестны руководству фирмы. Даже если существуют многочисленные версии программы, необходимо учитывать относительную несложность процедуры оценки корреляции двух программ, чтобы обнаружить различия в отличительных метках. Методы идентификации машинного кода с целью установления факта копирования довольно-таки надежны, но проблема с процедурами, встроенными в программу, значительно сложнее. Степень подобия процедур, которая обеспечивает правильное функционирование, представляет значительный интерес.

В то время как элементы интеллектуальной собственности, содержащиеся в программе, должны быть защищены, требование совместимости, особенно по диалоговому интерфейсу, имеет важное значение. Существует общий подход к проблеме диалогового взаимодействия (иногда определяемой как проблема «человек – машина»). Сообщество пользователей не должно изучать различные процедуры диалога при переходе к другим системам. Считается, что пользователю необходимо около трех месяцев, чтобы изучить приемы работы со сложной системой, например автоматизированного проектирования. О несогласованности процедур диалога в различных системах можно судить по тому, что более шести недель требуется для переобучения. Аналогичные проблемы возникают при использовании клавиатур с различающимся расположением клавиш.

4.6.2 Устройства контроля

Устройства регистрации событий, процедур или доступа к данным могут рассматриваться как часть общей системы защиты, причем как программ, так и данных. Подтверждение подлинности программы охватывает проблемы: от установления идентичности функционирования текущей программы и ее оригинала до подтверждения адекватности средств защиты. Сохранение выполняемой функции наиболее важно при выполнении финансовых сделок, а также для систем автоматизированного проектирования, когда целостность процедуры проектирования не должна быть нарушена. Последнее весьма важно, если используются устройства с низким уровнем защищенности, когда возможен обход проверок, связанных с защитой.

4.6.3 Водяные знаки

Использование водяных знаков как метода выявления подделки занимает особое место, поскольку препятствует созданию точной копии, которую пользователь не мог бы отличить от оригинала. В большинстве методов, предложенных для анализа проблемы идентификации программ, считается, что программы либо замаскированы, либо не полностью открыты для просмотра.

4.6.4 Психологические методы защиты

Эти 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). Во многих странах распространено мнение, что защита авторского права на программы усиливает психологическую обеспокоенность. 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.

Методы защиты программного обеспечения имеют широкий диапазон действия, и пользователь должен выбрать тот или иной механизм, учитывая стоимость его реализации. Фактор неудобства для покупателя может несколько снизить цену.

Стратегия, выбираемая изготовителем, будет зависеть от объема программных средств, которые следует защитить. Выбранный способ должен быть относительно дешев для изготовителя, чтобы можно было его включить в большое число программных продуктов, но одновременно он должен быть сложным и дорогостоящим для преодоления нарушителем, т.е. не должен относиться к одной-единственной программе. Угроза законного возмездия против квалифицированного хакера должна быть поддержана убедительными уликами.

Идеальные методы защиты должны позволять пользователю делать резервные копии для собственного использования и не должны ограничивать возможности компьютера (такие, как число строк ввода-вывода или выбор приоритета при работе в многозадачном режиме). Исключительно важное значение приобретают устройства зашиты, которые эффективны при работе в сетях.

Психологические и социальные факты должны способствовать защите и поддерживать в сознании нарушителя обеспокоенность, и это будет полезное дополнение к методам защиты, которыми мы располагаем.

4.7 Электронные ключи

Среди средств так называемых AAA (authentication, authorization, administration – аутентификация, авторизация, администрирование) важное место занимают программно-аппаратные инструменты контроля доступа к компьютерам – электронные замки, устройства ввода идентификационных признаков (УВИП) и соответствующее программное обеспечение (ПО) [9]. В этих средствах контроля доступа к компьютерам идентификация и аутентификация, а также ряд других защитных функций, выполняются с помощью электронного замка и УВИП до загрузки ОС.

По способу считывания современные УВИП подразделяются на контактные, дистанционные и комбинированные.

Контактное считывание идентификационных признаков осуществляется непосредственным взаимодействием идентификатора и считывателя.

При бесконтактном способе считывания идентификатор может располагаться на некотором расстоянии от считывателя, а сам процесс считывания осуществляется радиочастотным или инфракрасным методом.

УВИП могут быть электронными, биометрическими и комбинированными.

Электронные УВИП содержат микросхему памяти идентификационного признака.

Анализ новых технологий защиты информации показывает, что одним из наиболее мощных инструментов защиты ПО и БД от несанкционированного использования и нелегального копирования являются системы защиты на базе электронных ключей, в частности инструментальная система **Hardlock** с ее основным компонентом – электронным ключом Hardlock.

В целом система защиты Hardlock базируется на трех компонентах:

- электронном ключе Hardlock;
- криптокарте для программирования ключей Cripto-Programmer Card;

- программном обеспечении Hardlock Bistro, позволяющем быстро создать систему защиты для приложений и связанных с ними файлов данных.

Основой ключей Hardlock является заказной ASIC-чип (Application Specific Integrated Circuit) со встроенной EEPROM-памятью, разработанной компанией Aladdin. Чип имеет достаточно сложную внутреннюю организацию и специфический алгоритм работы, причем он программируется только с использованием специальной платы Cripto-Programmer Card, так как любой другой метод программирования ключей в настоящее время не обеспечивает безопасное хранение ключевой информации.

Каждый экземпляр этой платы является уникальным и позволяет задавать 43 680 вариантов работы алгоритма шифрования, которые могут быть использованы при программировании ASIC-чипа ключа. Логику работы чипа практически невозможно реализовать с помощью наборов микросхем, его практически невозможно воспроизвести, а содержащийся в его памяти микрокод – считать, расшифровать либо сэмулировать.

Использование подобного чипа позволяет работать на всех типах ПК. Последняя модель ключа Hardlock Twin может работать как с параллельным портом, так и с последовательным, позволяя подключать через него практически любые устройства, в том числе модемы, сканеры, принтеры и т.п.

В настоящее время ключи Hardlock выпускаются в следующих конфигурациях:

- внешний ключ на параллельный порт (Hardlock EYE);
- внешний ключ на параллельный и на последовательный порт (Hardlock Twin);
- внутренний ключ на шину ISA/MCA (Hardlock Internal);
- сетевой ключ (HL-Server) – внешний или внутренний;
- USB-порт (Hardlock USB);
- PC Card (Hardlock PCMCIA) – для лэптопов и ноутбуков.

Все модели ключей совместимы между собой и могут работать с большинством ОС, в различных сетях (с протоколами IPX, TCP/IP, Net-BIOS), осуществлять защиту 16- и 32-разрядных приложений (com, exe, dll) DOS и Windows и связанных с ними файлов данных в прозрачном режиме. При записи данные авто-

матически шифруются с использованием заданного аппаратно реализованного алгоритма, при чтении – расшифровываются. Симметричное шифрование производится блоками по 64 бита, причем для каждого нового блока ASIC генерирует новый сеансовый ключ длиной 48 бит.

Главным отличием ключей Hardlock от известных является высокий уровень защиты программ и аппаратное шифрование файлов данных.

Компания ALADDIN SOFTWARE SECURITY R.D. получила сертификат Госкомиссии РФ на продукт для защиты информации под названием Secret Disk. Этот продукт позволяет создавать на диске компьютера защищенные разделы, доступ к которым невозможен без установленного в USB-порт компьютера индивидуального электронного брелока. Перед началом работы секретный диск необходимо отпереть, вставив в порт электронный брелок, и ввести пароль, а по окончании – отключить. Если во время работы с конфиденциальными документами необходимо прерваться, не выключая компьютера, достаточно вынуть ключ. При этом экран гаснет, а клавиатура блокируется. Допускается применение ключа также при работе с документами, составляющими государственную тайну.

Дальнейшим шагом в развитии Secret Disk стал продукт Secret Disk Server, предназначенный для шифрования корпоративной информации на серверах Windows NT/2000.

Устройства ввода идентификационных признаков на базе идентификаторов **Proximity** (от английского слова proximity – близость, соседство) относится к классу электронных бесконтактных радиочастотных устройств. Они выпускаются в виде карточек, ключей, брелоков и т.п. Каждый из них имеет собственный уникальный серийный номер. Основными составляющими устройств являются интегральная микросхема для связи со считывателем и встроенная антена. В составе микросхемы находятся приемо-передатчик и запоминающее устройство, хранящее идентификационный код и другие данные. Внутри Proximity может быть встроена литиевая батарейка (активные идентификаторы). Активные идентификаторы могут считывать информацию на расстоянии нескольких метров. Расстояние считывания пассив-

ными идентификаторами (не имеющих батарейки) составляет десятки сантиметров.

Устройство считывания постоянно излучает радиосигнал, который принимается антенной и передается на микросхему. За счет принятой энергии идентификатор излучает идентификационные данные, принимаемые считывателем.

4.8 Технология защиты информации на основе смарт-карт

Появление новой информационной технологии смарт-карт (СК), основанной на картах со встроенным микропроцессором, позволило удобнее решать вопросы использования пластиковых денег. Однако уникальные возможности СК с микропроцессором, состоящие в высокой степени защиты от подделки, поддержке базовых операций по обработке информации, обеспечении высоких эксплуатационных характеристик, сделали СК одним из лидеров среди носителей конфиденциальной информации. Следует отметить отличительные особенности таких карт. СК содержит микропроцессор и ОС, которые обеспечивают уникальные свойства защиты, имеют контактное и бесконтактное исполнение (на рис. 4.3 показана схема бесконтактной СК). СК могут быть произведены только промышленным путем и, следовательно, не могут быть скопированы. Каждая СК имеет уникальный код, определенный на производстве, и если на другую карту будут записаны те же данные, что и на оригинале, то различия во внутренних параметрах дают возможность системе отличить одну карту от другой. СК может быть запрограммирована так, что она выходит из строя при попытке НСД. Данные шифруются с помощью различных алгоритмов, в том числе ГОСТ 28147 – 89 или DES и секретных ключей, которые содержатся на микросхеме карты. Если карта обнаруживает несоответствие введенного пользователем pin-кода – персонального идентификационного номера – со своим личным pin-кодом несколько раз подряд (обычно 3 раза), она может самоуничтожиться, т.е. стать непригодной для использования, записав в память вместо системных ключей случайно сгенерированные числа.

Таким образом, технология СК обеспечивает надежное хранение ключей и доступ к различным информационным ресурсам.

Персональные идентификаторы **iKey** компании Rainbow являются недорогими брелоками, которые могут использоваться на любой рабочей станции, имеющей универсальную последовательную шину (USB). Они обеспечивают надежность, простоту и безопасность в такой же степени, как и смарт-карты, но без сложностей и лишних затрат, связанных с с использованием считывателя [9]. iKey являются идеальным инструментом для контроля доступа к сетевым службам. iKey 2000 поддерживает и интегрируется со всеми основными прикладными системами, работающими по технологии PKI и используемыми в сетях отдельной организации, нескольких взаимодействующих организаций. Указанные системы включают Microsoft Internet Exolorer и Outlook, Netscape, Entrust, Baltimore, Xcert, Verisign и др. iKey 2000 разрабатывался для защиты цифровой идентичности в рамках инфраструктуры открытых ключей (PKI). iKey 2000 способен с помощью аппаратных средств генерировать и сохранять в памяти пары открытых ключей и цифровые сертификаты, а также производить цифровую подпись. Личный PKI-ключ недоступен компьютеру клиента.

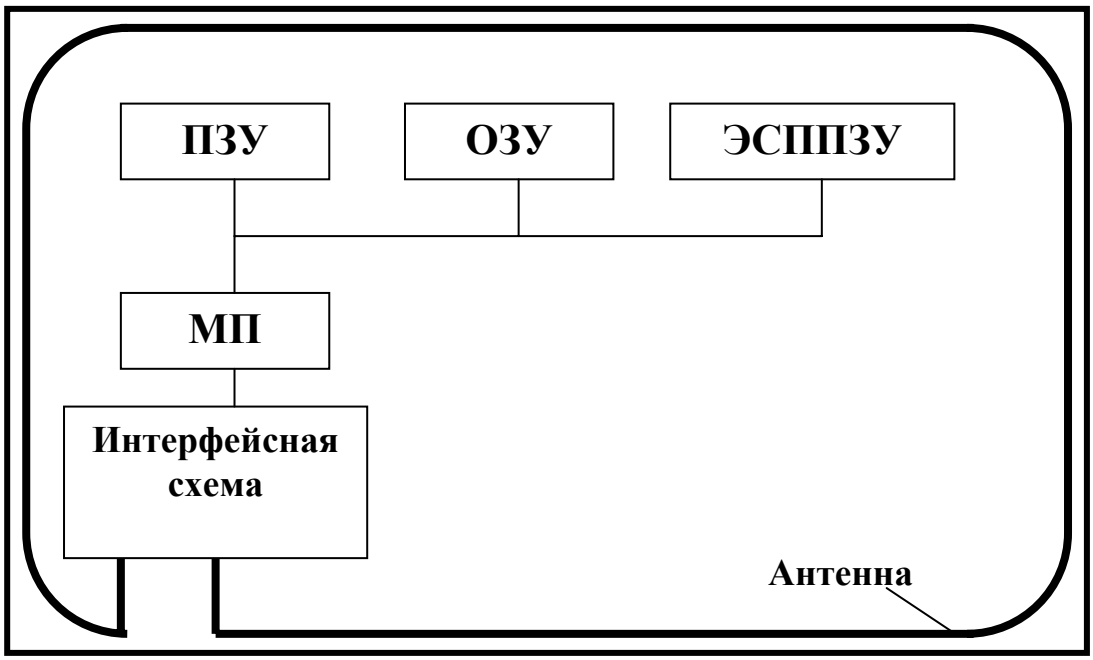


Рис. 4.3

iKey 2000 создает мощную систему защиты и криптографического кодирования непосредственно внутри аппаратного устройства. Для iKey 2000 пользователю поставляется программное обеспечение. Устройство содержит полный набор криптографических библиотек для броузеров Netscape и Internet Exolorer, а также для клиентов электронной почты. iKey 2000 действует одновременно как смарт-карта и считыватель, находящиеся в едином устройстве с конструктивом USB. Для активизации прикладной программы достаточно вставить iKey 2000 в USB-порт.

iKey 2000 реализует более простой метод обеспечения привилегий пользователя, чем пароли или чисто программные сертификаты. Чтобы запрограммировать ключ, администратору потребуется всего несколько минут. Потерянные ключи могут быть дезактивированы и изменены.

4.9 Создание защищенной операционной системы

Анализ архитектуры и характеристик сертифицированных защищенных систем показывает [1]:

1. В настоящее время имеются два принципиально различных метода к проектированию защищенных информационных систем. Первый метод предполагает разработку защищенной системы с нуля, и в этом случае для нее специально разрабатываются защищенные приложения. Вследствие значительной трудоемкости и стоимости такой подход приемлем для состоятельных производителей. Второй подход заключается в доработке системы-прототипа с целью улучшения ее характеристик защиты.

2. Принципы построения защищенных операционных систем изменялись с развитием сферы применения компьютерных систем. Защищенные ОС стали разрабатываться для рабочих станций и персональных компьютеров, для сетевых систем. Особое значение при этом имела ОС UNIX, которая наиболее часто служила основой для разработки защищенных ОС.

3. Для соответствия требованиям сертификации ОС должна обеспечивать произвольный и нормативный доступ. Для реализации произвольного доступа защищенные ОС используют как традиционный механизм битов защиты, так и списки контроля доступа (ACL). Для организации нормативного контроля доступа

в большинстве случаев используется модель Белла и Лападула. Обеспечение целостности данных в защищенных ОС обеспечивается применением нормативной модели Биба.

4. Так как при сертификации защищенные ОС подлежат формальному анализу, то защищенные системы должны разрабатываться на основе иерархического и модульного проектирования.

Все защищенные ОС в общем реализуют один и тот же набор защитных функций – управление доступом и контроль за его осуществлением, идентификация и аутентификация, аудит, прямое взаимодействие и т.д. Это связано с тем, что все разработчики ориентируются на соответствующие разделы стандартов информационной безопасности. Однако способы реализации этих функций, как и реализуемый ими уровень защиты, различаются от системы к системе даже в рамках одного класса требований [1].

Построению защищенной системы должно предшествовать определение класса требований стандарта информационной безопасности к данной системе. На современном уровне развития информационных технологий необходимо разрабатывать системы с жесткими требованиями по защищенности. В обязательном порядке должны быть реализованы произвольное и нормативное управление доступом, а структура ТСВ системы должна позволять применение формальных методов анализа.

Поэтому будем ориентироваться на требования безопасности уровня В3 «Оранжевой книги». Эти требования содержат поддержку формально определенной модели безопасности, предусматривают произвольное и нормативное управление доступом, метки безопасности, контроль доступа к субъектам и объектам. ТСВ в такой системе должна быть структурирована с целью исключения из нее подсистем, не отвечающих за функции защиты, и быть компактной для надежного тестирования и анализа. ТСВ должна быть минимизирована по сложности. Средства аудита должны содержать механизмы оповещения администратора о событиях, влияющих на безопасность системы. Требуется наличие средств восстановления работоспособности после сбоев и контроля скрытых каналов утечки информации.

С точки зрения обеспечения безопасности самой прогрессивной на сегодняшний день технологией построения ОС является технология микроядра. В отличие от традиционной архитектуры, в которой ОС представляет собой монолитное ядро, реализующее основные функции по управлению аппаратными ресурсами и организующее среду для выполнения пользовательских процессов, микроядерная архитектура распределяет функции ОС между микроядром и входящими в состав ОС системными сервисами (процессы, равноправные с пользовательскими приложениями).

Микроядро выполняет базовые функции операционной системы, на которые опираются эти системные сервисы и приложения. В итоге такие важные компоненты ОС как файловая система, сетевая поддержка и т.д. превращаются в независимые модули, которые функционируют как отдельные процессы и взаимодействуют с ядром и друг с другом на общих основаниях. Имевшее раньше место четкое разделение программного обеспечения на системные и прикладные программы размывается, т.к. между процессами, реализующими функции ОС, и прикладными процессами, выполняющими программы пользователя, нет никаких различий. Все компоненты системы используют средства микроядра для обмена сообщениями, но взаимодействуют непосредственно. Микроядро лишь проверяет законность сообщений, пересылает их между компонентами и обеспечивает доступ к аппаратуре.

Другое изменение в технологии построения ОС, связанное исключительно с внедрением технологии микроядра, это организация взаимодействий между процессами и ядром с помощью универсального механизма передачи информации – обмена сообщениями, пришедшего на смену технике системных вызовов. При этом десятки или даже сотни вызовов, различающихся числом и типом параметров, можно заменить несколькими типами сообщений, которые содержат компактные порции информации и могут передаваться от одного обработчика к другому.

На современном этапе развития ОС эта технология является самой перспективной, т.к. позволяет преодолеть самые заметные недостатки существующих систем – отсутствие мобильности, громоздкость, ресурсоемкость. Реализация многих традиционных

функций ОС за пределами ядра способствует построению на базе этого ядра операционных систем с недостижимым раннее уровнем модульности и расширяемости.

Чтобы иметь представление о базовом наборе понятий, необходимом для изложения деталей реализации средств защиты, рассмотрим некоторые понятия и основные принципы построения архитектуры современных микроядерных операционных систем на примере лежащего в основе Trusted Mach микроядра MK++[1].

4.9.1 Основные положения архитектуры микроядерных ОС

В основе архитектуры микроядерных ОС лежат следующие базовые концепции:

- минимизация набора функций, поддерживаемых микроядром, и реализация традиционных функций ОС (файловая система, сетевая поддержка) вне микроядра;
- организация синхронного и асинхронного взаимодействия между процессами через механизм обмена сообщениями;
- все отношения между компонентами строятся на основе модели клиент/сервер;
- применение объектно-ориентированного подхода при разработке архитектуры и программирования системы.

Минимизация функций микроядра дает возможность сконцентрировать в нем код, зависящий от аппаратной платформы, что позволяет повысить переносимость ОС до максимума. Таким образом, микроядро реализует только жизненно важные функции, лежащие в основе операционной системы, являющиеся базисом для всех системных служб, сервисов и прикладных программ.

Использование механизма передачи сообщений позволяет установить единый интерфейс для взаимодействия между всеми компонентами системы независимо от их уровня и назначения, что дает возможность строить все информационные связи в системе по модели клиент/сервер.

В модели клиент/сервер все компоненты рассматриваются либо как потребители (клиенты), либо как поставщики (серверы) некоторых ресурсов или сервисов. Стандартизированные протоколы предоставления сервиса или ресурсов позволяют серверу обслуживать клиентов независимо от деталей их реализации, что открывает перед разработчиками широкие возможности для построения распределенных систем. Инициатором обмена обычно является клиент, который посылает запрос на обслуживание серверу, находящемуся в состоянии ожидания запроса. Один и тот же процесс может являться клиентом по отношению к одним ресурсам и быть сервером для других. Данная модель успешно применяется не только при построении ОС, но и при создании программного обеспечения любого уровня. Применение модели клиент/сервер по отношению к ОС состоит в реализации не вошедших в состав ядра компонентов ОС, в виде множества серверов, каждый из которых предназначен для обслуживания определенного ресурса (например, управление памятью, процессами, контроль доступа и т.д.).

Наиболее полно раскрыть преимущества технологии клиент/сервер позволяет применение методов объектно-ориентированного проектирования и программирования. Если каждый сервер обслуживает только один тип ресурсов и представляет его клиентам в виде некоторой абстрактной модели, то такой сервер можно рассматривать как объект, т.к. он обладает всеми необходимыми для этого качествами. Объект должен обладать состоянием, поведением и индивидуальностью. Для каждого сервера существует четко определенная модель состояний и переходов между ними. И, наконец, «поведение» каждого сервера однозначно регламентируется протоколом его взаимодействия с клиентами. Соответственно, можно строить модель ОС, построенной по этим принципам, в виде иерархии серверов и моделей, представляемых ими ресурсов, а также описывать существующие между ними взаимосвязи с помощью объектных отношений наследования, использования и включения.

С точки зрения создания защищенных операционных систем, использование объектно-ориентированного подхода в сочетании с микроядром и технологией клиент/сервер позволяет разработчику реализовать взаимодействие субъектов и объектов, а

также контроль за информационными потоками с помощью ограниченного числа простых и понятных механизмов, что облегчает адекватность реализации модели безопасности и позволяет применять формальные методы анализа.

4.9.2 Микроядерная архитектура с точки зрения создания защищенных систем

Благодаря принципам, на которых основаны микроядерные ОС, их компоненты функционируют на основе очень небольшого и сравнительно простого набора абстракций, составляющих базис системы и компактно реализованных в микроядре. Можно сказать, что для защищенных систем такая архитектура является оптимальной, т.к. она позволяет достаточно просто и эффективно разрешить целый ряд вопросов, неизбежно возникающих при реализации защищенных систем:

1. Выявление потоков информации в системе. Поскольку все взаимодействия осуществляются исключительно посредством механизма передачи сообщений, очевидно, что, контролируя потоки сообщений, можно быть уверенным в том, что контролируются все информационные потоки в системе;

2. Определение субъектов и объектов взаимодействия. Как уже говорилось, все задачи в микроядерных системах связаны между собой отношениями клиент/сервер. Соответственно, субъектом взаимодействия всегда является задача-клиент, а объектом – ресурс, обслуживаемый задачей-сервером.

3. Размещение подсистемы контроля доступа. Поскольку единственным механизмом взаимодействия является передача сообщений, очевидно, что функция контроля за информационными потоками должна быть возложена на ту часть ядра системы, которая реализует этот механизм. Контроль и управление доступом к ресурсам и объектам могут быть реализованы, как в составе серверов, отвечающих за обслуживание этих ресурсов и объектов, так и на уровне всей системы в целом. Так как многие сервера обслуживают однотипные ресурсы и объекты (файлы, устройства и т.д.), то контроль доступа к ним с целью унификации реализуется на глобальном уровне, с помощью Сервера имен,

который формирует глобальное пространство имен системы и организует взаимодействие между клиентами и серверами.

4. Минимизация объема программного кода, отвечающего за контроль доступа. Как следует из предыдущего пункта, в системе существует всего две процедуры, реализующие контроль за осуществлением доступа: на уровне передачи сообщений и глобальная система на уровне именованных объектов. Таким образом, объем программ, корректность функционирования которых критична для безопасности всей системы, сокращен до минимума.

5. Использование объектно-ориентированных технологий программирования. Контроль за потоками сообщений и доступом процессов к ресурсам из глобального пространства имен можно осуществлять на основе унифицированного набора свойств сообщений (источник, приемник) и ресурсов (идентификатор процесса, имя ресурса). За счет этого достигается абстрагирование системы защиты от специфики информационных взаимодействий, но в то же время сохраняется ее гибкость за счет возможности использования в задачах-серверах специализированных механизмов защиты, адаптированных и конкретизированных к ресурсам, которые обслуживают эти сервера.

6. Верификация и анализ защиты. Достигнутая с помощью применения описанных решений простота и компактность средств контроля за осуществлением доступа очевидным образом, за счет структуризации системы, способствует применению формальных методов верификации и анализа программного кода средств защиты.

4.9.3 Микроядро как основа для создания защищенной ОС нового поколения – МК++

Микроядро МК++ отвечает всем требованиям, предъявляемым к ОС нового поколения (многопоточность, расширяемость, мобильность и т.д.) [1]. Разработчики преследовали следующие цели:

- отработать технологии реализации современных требований к операционным системам;

- создать основу для разработки будущих поколений защищенных ОС, рассчитанных на достаточно высокий класс требований безопасности;
- обеспечить возможность работы микроядра и приложений в режиме реального времени;
- предусмотреть максимальное использование параллельности, как для приложений, так и для самих компонентов операционной системы в расчете на распределенные и массово-параллельные системы будущего;
- обеспечить переносимость микроядра с одной аппаратной платформы на другую;
- обеспечить совместимость с существующим программным обеспечением.

Рассмотрим МК++ только с точки зрения создания основы для защищенной ОС.

Поскольку разработчики МК++ стремились создать классическую микроядерную систему, в которой практически все традиционные для ОС функции вынесены за пределы ядра, само микроядро осуществляет только следующий набор функций:

- управление физической аппаратурой (оперативной памятью, процессорами, внешними устройствами);
- распределение ресурсов аппаратной платформы между процессами (время процессора, память и т.д.);
- изоляция процессов;
- организация взаимодействия между процессами;
- управление процессами (создание, уничтожение, переключение).

Ядро является своеобразным арбитром, роль которого сводится к поддержанию некоторого набора «правил игры» внутри операционной системы, все остальные традиционные функции ОС должны быть реализованы вне ядра.

Для описания микроядерной архитектуры необходимо перечислить набор основных понятий, используемых в микроядерной технологии построения ОС:

Задача. В микроядерных системах это понятие заменяет традиционное для ОС понятие *процесс*. Задача представляет собой обобщение понятия процесс и обозначает набор ресурсов,

который образует среду для выполнения *потоков* (см. далее). Эта среда включает в себя:

- изолированное от других задач адресное пространство;
- среду выполнения прикладного процесса;
- атрибуты безопасности;
- средства взаимодействия с ядром;
- средства взаимодействия с другими задачами.

Каждая задача имеет свое собственное пространство имен *портов*. Задача может выступать в роли потребителя ресурсов (клиент) или предоставлять определенные ресурсы другим задачам (сервер). Одна и та же задача может являться одновременно и сервером, и клиентом, потребляя ресурсы, контролируемые одними задачами, и предоставляя свои ресурсы другим. Доступ к ресурсам, как и взаимодействие с другими задачами и ядром, осуществляется только с помощью обмена сообщениями через порты.

Поток (Thread) – логически связанный поток выполняемых команд. Каждый поток выполняется в контексте какой-либо *задачи* и может осуществлять непосредственный доступ только к ее среде. Потоки являются основной единицей вычислений и единственными активными элементами в системе. Поток представляет собой последовательность команд, выполняемых в рамках задачи. Его единственным атрибутом является состояние процессора. Все потоки внутри задачи совместно используют адресное пространство и наследуют атрибуты безопасности задачи.

Порт – однонаправленный информационный канал, с помощью которого задачи обмениваются информацией друг с другом и с ядром, осуществляя операции посылки и получения *сообщений*. Задачи могут получить доступ к портам только при наличии у них прав на посылку/прием сообщений.

Сообщение – логически связанный набор данных, предаваемый через порт за одно обращение. Для осуществления контроля доступа ядро снабжает все сообщения специальной меткой, идентифицирующей отправителя сообщения.

Пример взаимодействия между сервером, клиентом и микроядром с помощью портов и сообщений показан на рис. 4.4.

Рассмотрим внутреннюю структуру МК++, назначение и основные функции составляющих его компонентов. Как видно из рис. 4.5, структура МК++, представляет собой иерархию, в которой каждый уровень опирается на сервисы, реализуемые нижестоящими уровнями, и, в свою очередь, обслуживает уровни, лежащие выше.

Для того чтобы пояснить, как функционирует микроядро, рассмотрим в общих чертах тот сервис, который реализуется компонентами МК++ и некоторыми задачами, обеспечивающими важные для функционирования системы функции.

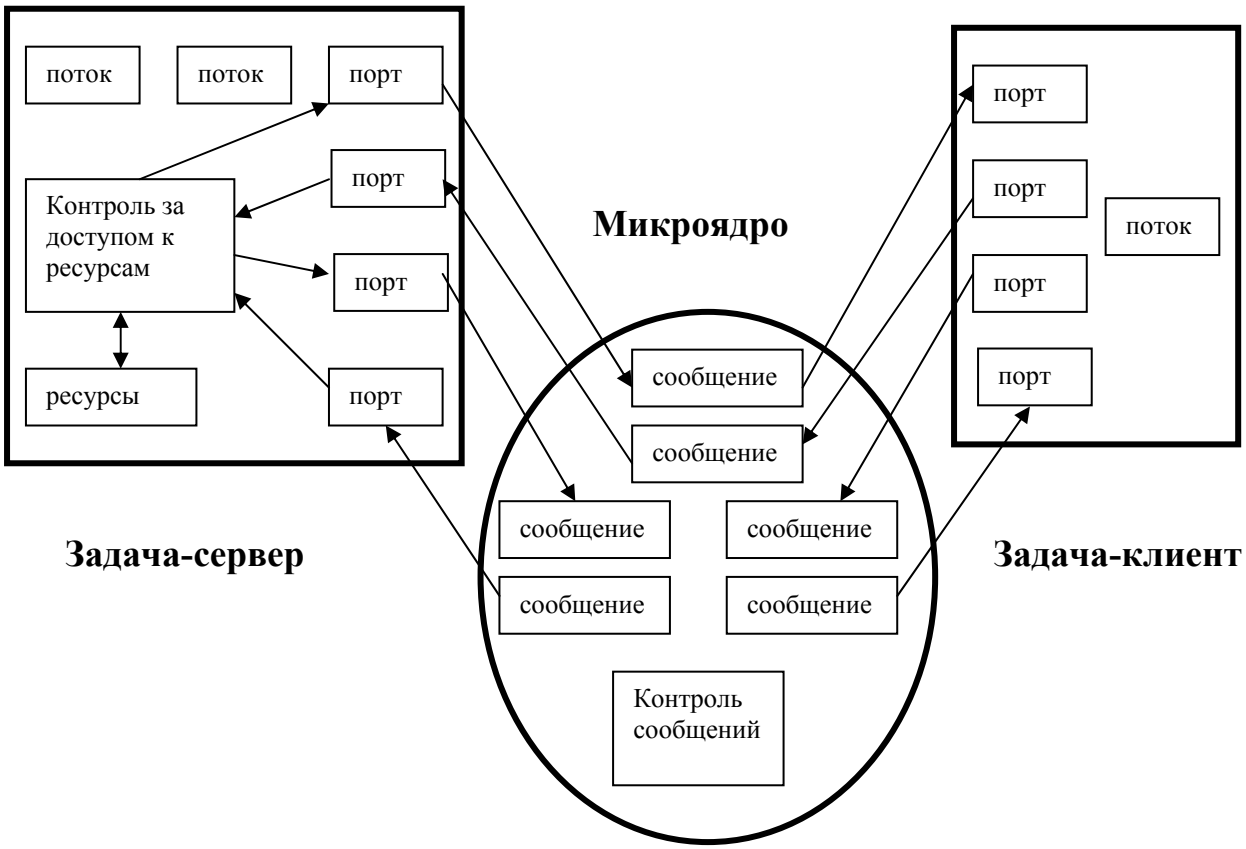


Рис. 4.4

1. Подсистема управления процессорами представляет собой самую низкоуровневую и в наибольшей степени зависимую от аппаратной платформы подсистему во всей структуре микроядра. Подсистема управления процессорами контролирует все переходы между различными режимами работы процессоров, в том числе изменение уровня привилегий и переходы между различными кольцами защиты, а также переключение с одного по-

тока команд на другой, и также обработку прерываний и исключений. Данный уровень скрывает от вышестоящих все особенности аппаратной архитектуры, такие как порядок сохранения и восстановления регистров, формат таблиц трансляции адресов, способы управления приоритетами прерываний и т.д. Таким образом, подсистема управления процессорами реализует следующие функции:

- управление состоянием процессоров;
- обработка аппаратных прерываний и исключений;
- синхронизация совместной работы нескольких процессоров;
- предоставление процессоров в распоряжение потоков и контроль за использованием процессорного времени.

2. Управление ресурсами микроядра. Служебные структуры (сообщения, порты и т.д.) требуют выделения определенных ресурсов (памяти) и при этом должны быть доступны как для приложений, так и из микроядра, что влечет за собой необходимость размещения их в адресном пространстве микроядра. Подсистема управления ресурсами памяти микроядра отвечает за выделение памяти в адресном пространстве микроядра, ее освобождение и контроль за использованием.

3. Подсистема организации взаимодействий реализует для вышележащих уровней сервис, позволяющий посылать и принимать элементы данных. Данная подсистема включает в свой состав универсальные механизмы, которые могут быть использованы для передачи абстрактных элементов (непосредственно или с буферизацией в очереди) от некоторого источника к некоторому приемнику. Тип элементов не имеет значения. Самым распространенным примером использования этого сервера является механизм передачи сообщений, обслуживающий информационный обмен между задачами и ядром, однако этот сервис может быть использован для других целей.

4. Подсистема управления физической памятью отвечает за порядок распределения и использования физической памятью системы. Кроме того, данная подсистема выполняет машинно-зависимые операции трансляции адресов для различных адресных пространств.


Загрузчик
Сервер свопинга
Интерфейс микроядра
Управление ресурсами микроядра
Управление портами
Управление адресными пространствами задач
Виртуальные устройства
Реальное время
Управление виртуальной памятью
Идентификация объектов микроядра
Управление вводом-выводом и таймерами
Управление физической памятью
Организация взаимодействий
Управление ресурсами микроядра
Управление процессорами
Аппаратная платформа

Рис. 4.5

5. Подсистема ввода-вывода выполняет низкоуровневый ввод/вывод информации, специфичный для каждого конкретного устройства. Кроме того, эта подсистема управляет аппаратными таймерами, а также осуществляет обработку прерываний от внешних устройств.

6. Подсистема идентификации ставит в соответствие каждой сущности уровня микроядра (задачи, потоки, порты и т.д.) уникальный идентификатор.

7. Идентификатор никогда не используется повторно, даже если объект, на который он ссылается, уничтожен. Эта подсистема обеспечивает механизм взаимодействий на уровне всей системы. Все манипуляции с объектами микроядра опираются на этот сервис.

8. Подсистема виртуальной памяти отвечает за отображение виртуальных адресных пространств ядра и задач в физическую память и обеспечивает инициализацию и очистку объектов памяти.

9. Подсистема реального времени обеспечивает функции работы с сигналами и таймерами и осуществляет доступ к аппаратным часам.

10. Подсистема управления виртуальными устройствами поддерживает абстракции, представляющие их на уровне задач. Данная подсистема ответственна за присваивание имен и реализацию операций открытия/закрытия для всех устройств.

11. Подсистема управления адресными пространствами организует виртуальные адресные пространства задач и пользуется сервисом виртуальной памяти для манипулирования областями памяти, находящимися в адресном пространстве задач. Эта подсистема также осуществляет низкоуровневое управление доступом к пространствам памяти.

12. Подсистема управления портами отвечает за организацию пространства имен портов для задач и осуществляет отображение имен портов во внутренние идентификаторы микроядра с помощью подсистемы идентификации. Именно эта подсистема реализует контроль за информационными потоками и определяет, обладает правом приема сообщений из порта и отправки в него.

13. Подсистема управления ресурсами обеспечивает высокоуровневые средства управления ресурсами ядра, в частности

задачами и потоками. Управление состоит в отслеживании событий, связанных с ресурсами, и манипуляциях с ними, например, создание, завершение и приостановка выполнения потоков. Специальный механизм данной подсистемы позволяет реализовать различные политики управления ресурсами ядра.

14. Интерфейс микроядра определяет границу между выполнением потоков в контексте ядра и в пользовательском контексте. Подсистема управления процессорами осуществляет переключение процессора таким образом, что вход в контекст микроядра из прикладной задачи и выход из него возможны только через интерфейс микроядра.

15. Сервер загрузки находится вне микроядра и представляет собой обычную пользовательскую задачу, создаваемую процессом инициализации ОС. Целью этой задачи является загрузка и запуск серверов, реализующих системные сервисы, в том числе входящих в состав ТСВ.

16. Сервер свопинга также является обычной задачей. Он обеспечивает сохранение на диске областей виртуальной памяти, вытесненных из физической памяти.

В микроядре отсутствуют средства, отвечающие за реализацию политики безопасности и управления доступом к информационным ресурсам системы. Однако МК++ включает два механизма, необходимые для реализации этих функций, а именно: изоляцию задач и контроль за передачей сообщений. Все остальные механизмы защиты, опирающиеся на этот сервис, могут быть реализованы в составе серверов. В совокупности микроядро и эти серверы образуют ТСВ системы.

Глава 5. ПРОГРАММНО-АППАРАТНЫЕ КОМПЛЕКСЫ ЗАЩИТЫ ИНФОРМАЦИИ

5.1 Программно-аппаратный комплекс «Аккорд – 1.95»

5.1.1 Общие сведения

Программно-аппаратный комплекс средств защиты информации от несанкционированного доступа (ПАК СЗИ НСД) «Аккорд – 1.95», далее комплекс «Аккорд», предназначен для применения на ПЭВМ типа IBM PC в целях защиты ПЭВМ и информационных ресурсов от НСД и обеспечения конфиденциальности информации, обрабатываемой и хранимой в ПЭВМ при многопользовательском режиме ее эксплуатации.

Комплекс разработан ОКБ САПР при участии фирмы «Инфокрипт» на основании лицензии Государственной технической комиссии при Президенте РФ (Гостехкомиссии России) от 02.06.95 N 56.

Комплекс «Аккорд» состоит из программно-аппаратных средств «Аккорд АМДЗ» и ПО разграничения доступа «Аккорд 1.95-00».

В настоящее время комплекс «Аккорд-1.95» выпускается в трех основных версиях в зависимости от модификации аппаратных средств (контроллеров):

версия 2.0 – контроллер «Аккорд – 4++»;

версия 3.0 – контроллер «Аккорд – 5»;

версия 4.0 – контроллер «Аккорд – 4.5»; «Аккорд – СБ/2».

Все модификации:

Могут использоваться на ПЭВМ с процессором 80386 и выше, объемом RAM 640 Кбайт и более.

Для установки необходим свободный слот:

ISA – для контроллеров «Аккорд – 4++», «Аккорд – 4.5»;

PCI – для контроллера «Аккорд – 5»; «Аккорд – СБ/2».

Используют для идентификации персональные ТМ-идентификаторы DS 199X с объемом памяти до 64 Кбит.

Используют для аутентификации пароль до 12 символов.

Блокируют загрузку с FDD, CD ROM, ZIP Drive.

Предусматривают регистрацию от 16 до 32 пользователей.

Имеют аппаратный датчик случайных чисел (ДСЧ).

Имеют возможность применения съемника, использующего внутреннее подключение к контроллеру (внутренний съемник).

Обеспечивают контроль целостности программ, данных и системных областей жестких дисков.

Имеют внутреннюю энергонезависимую память для хранения данных о зарегистрированных пользователях и журнала регистрации событий.

Допускают изменение встроенного ПО (технологический режим) без замены платы контроллера.

Обеспечивают режим доверенной загрузки ОС (выполнение процедур идентификации/аутентификации пользователя, контроль целостности аппаратной части ПЭВМ, системных файлов, программ и данных до загрузки ОС на аппаратном уровне).

Особенности этих модификаций приведены в таблице 1.

Таблица 1

Особенности различных типов контроллеров	«Аккорд-4++»	«Аккорд-4.5»	«Аккорд-5», «Аккорд СБ/2»
Тип используемой системной шины	ISA	ISA	PCI
Установка реле управления физическими линиями (5В, 300 Ма)	Не предусмотрена	Возможна установка 1-го или 2-х реле по заказу	Возможна установка 1-го или 2-х реле по заказу
Возможность перепрограммирования всех элементов без изменения аппаратной части	+	+	+
Установка таймера реального времени с собственным источником питания	Не предусмотрена	Возможна установка по заказу	Возможна установка по заказу
Установка датчика случайных чисел для криптографических применений	Не предусмотрена	Производится для всех контроллеров данного типа	Производится для всех контроллеров данного типа

Продолжение табл. 1

Особенности различных типов контроллеров	«Аккорд-4++»	«Аккорд-4.5»	«Аккорд-5», «Аккорд СБ/2»
Установка интерфейса RS 232 для считывателя smart-карт	Не предусмотрена	Не предусмотрена	Возможна установка по заказу

5.1.2 Технические и организационные сведения

Для установки комплекса «Аккорд» требуется следующий минимальный состав технических и программных средств:

- IBM PC AT совместимую ПЭВМ (с процессорами 80386 и старше);
- наличие на ПЭВМ HDD;
- наличие свободного слота на материнской плате ПЭВМ (ISA, PCI);
- операционная система MS DOS v.3.10 и выше.

Объем дискового пространства, необходимого для установки программных средств комплекса, составляет от 700 Кб до 1,2 Мб в зависимости от модификации программных средств комплекса.

Количество идентификаторов, используемых в комплексе «Аккорд», определяется заказчиком при поставке.

Комплекс «Аккорд» проверен на совместимость практически со всем доступным разработчику программно-аппаратным обеспечением ПЭВМ как зарубежного, так и отечественного производства. Совместимость обеспечивается правильной установкой и настройкой комплекса.

Для эффективного применения комплекса и поддержания необходимого уровня защищенности ПЭВМ и информационных ресурсов необходимы:

- физическая охрана ПЭВМ и ее средств, в том числе проведение мероприятий по недопущению изъятия контроллера комплекса;
- наличие администратора службы безопасности информации (СБИ) – привилегированного пользователя, имеющего осо-

бый статус и абсолютные полномочия (супервизора). Администратор СБИ планирует защиту информации на предприятии (учреждении, фирме и т.д.), определяет права доступа пользователям в соответствии с утвержденным Планом защиты, организует установку комплекса в ПЭВМ, эксплуатацию и контроль за правильным использованием ПЭВМ с внедренным комплексом, в том числе учет выданных ТМ-идентификаторов, осуществляет периодическое тестирование средств защиты комплекса.

- использование в ПЭВМ технических и программных средств, сертифицированных как в Системе ГОСТ Р, так и в ГСЗИ.

Применение комплекса «Аккорд» совместно с сертифицированными программными средствами криптографической защиты информации (СКЗИ) и/или программными средствами защиты информации от НСД (СЗИ НСД) позволяет значительно снизить нагрузку на организационные меры защиты информации, определенные условиями применения этих средств. При этом класс защищенности не снижается.

5.1.3 Особенности защитных функций комплекса

Комплекс «Аккорд» – это простой, но чрезвычайно эффективный комплекс технических средств, используя который можно надежно защитить информацию на компьютере без переделки ранее приобретенных программных средств.

Защитные функции комплекса реализуются применением:

1. Дисциплины защиты от НСД к ПЭВМ, включая идентификацию пользователя по уникальному ТМ-идентификатору и аутентификацию с учетом необходимой длины пароля и времени его жизни, ограничением времени доступа субъекта к компьютеру.

2. Процедур блокирования экрана и клавиатуры в случаях, в которых могут реализовываться угрозы информационной безопасности.

3. Дисциплины разграничения доступа к ресурсам АС (ПЭВМ), определяемой атрибутами доступа, которые устанавливаются администратором БИ в соответствие каждой паре «субъект доступа – объект доступа» при регистрации пользователей.

4. Дисциплины применения специальных процедур печати, управления стандартными процедурами печати, процедурами ввода/вывода на отчуждаемые носители информации.

5. Контроля целостности критичных с точки зрения информационной безопасности программ и данных (дисциплины защиты от несанкционированных модификаций).

6. Средств функционального замыкания информационных систем за счет использования средств защиты комплекса.

7. Других механизмов защиты в соответствии с требованиями нормативных документов по безопасности информации.

Комплекс «Аккорд» может применяться в произвольной и функционально замкнутой программной среде, обеспечивая при этом класс защищенности АС (ПЭВМ) 1В по классификации, надежно гарантируя при этом:

- защиту от несанкционированного доступа к АС (ПЭВМ) и ее ресурсам;
- разграничение доступа к ресурсам, в т.ч. к внешним устройствам, в соответствии с уровнем полномочий пользователей;
- защиту от несанкционированных модификаций программ и данных, внедрения разрушающих программных воздействий (РПВ);
- контроль целостности программ и данных;
- функциональное замыкание информационных систем с исключением возможности несанкционированного выхода в ОС, загрузки с FDD и несанкционированного прерывания контрольных процедур с клавиатуры.

Отметим, что в комплексе «Аккорд» используются и некоторые дополнительные механизмы защиты от НСД к ПЭВМ (АС). Так, в частности, для пользователя администратор БИ может установить:

- время жизни пароля и его минимальную длину, практически исключая тем самым возможность быстрого его подбора;
- временные ограничения использования ПЭВМ установкой интервала времени по дням недели (с дискретностью 30 мин), в котором разрешена работа для данного пользователя;
- параметры управления экраном – гашение экрана через заранее определенный интервал времени (в случае, если в течение указанного интервала действия оператором не выполнялись). Возможность продолжения работы предоставляется только после

проведения повторной идентификации по персональному ТМ-идентификатору пользователя;

- целесообразного с точки зрения критичности информационной безопасности объема конфиденциальной информации, выводимого на внешние устройства ПЭВМ;

- подачу соответствующих звуковых и визуальных сигналов при попытках несанкционированного доступа к ПЭВМ (АС) и ее ресурсам.

Предусмотрено подключение подсистемы криптографической защиты информации, которая позволяет пользователю зашифровать/расшифровать свои данные с использованием индивидуальных ключей, хранящихся в персональном ТМ-идентификаторе. Поставка криптографических систем защиты информации (в соответствии с действующим законодательством) и библиотеки программ для программирования работы с контроллером комплекса «Аккорд» оговаривается при заказе комплекса.

5.1.4 Построение системы защиты информации на основе комплекса

Построение системы защиты информации с использованием комплекса «Аккорд» и ее взаимодействие с программно-аппаратным обеспечением ПЭВМ показаны на рис. 5.1.

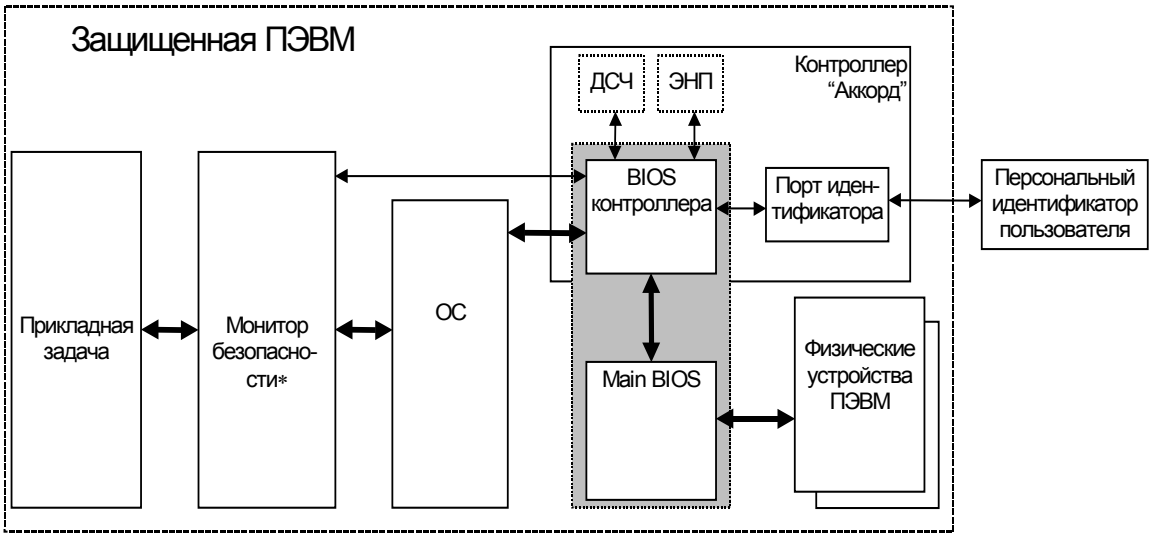


Рис. 5.1

Защита информации с использованием средств комплекса основана на обработке событий, возникающих при обращении прикладных программ или системного ПО к ресурсам ПЭВМ. При этом средства комплекса перехватывают соответствующие программные и/или аппаратные прерывания, в случае возникновения контролируемого события (запрос прерывания) анализируют запрос и в зависимости от соответствия полномочий субъекта доступа (его прикладной задачи), установленных администратором БИ ПРД, либо разрешают, либо запрещают обработку этих прерываний.

Комплекс «Аккорд» состоит из собственно средств защиты ПЭВМ от НСД и средств разграничения доступа к ее ресурсам, которые условно можно представить в виде четырех взаимодействующих между собой подсистем (рис. 5.2.) защиты информации.

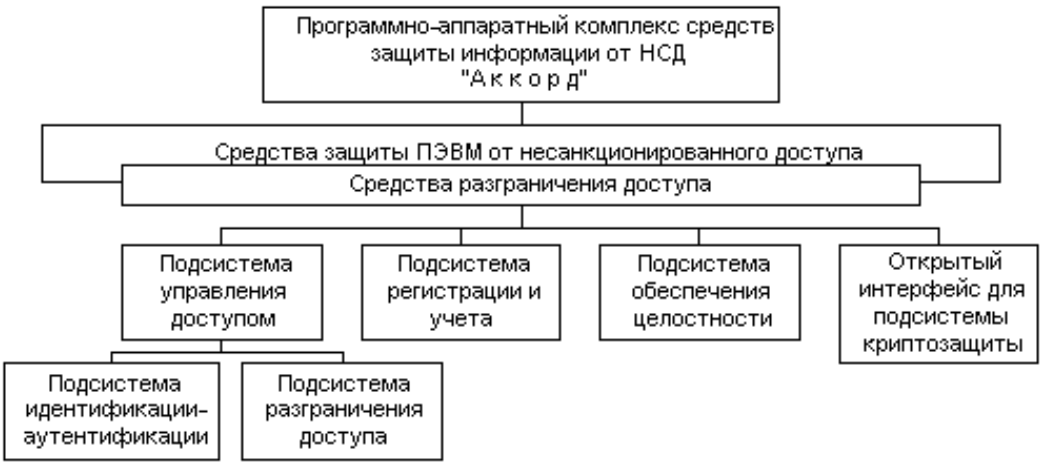


Рис. 5.2

5.1.4.1 Подсистема управления доступом

Предназначена для защиты ПЭВМ от посторонних пользователей, управления доступом к объектам доступа и организации совместного их использования зарегистрированными пользователями в соответствии с установленными правилами разграничения доступа. Под посторонними пользователями понимаются все лица, не зарегистрированные в системе (не имеющие зарегистрированного в конкретной ПЭВМ ТМ-идентификатора). Защита от посторонних пользователей обеспечивается процедурами иден-

тификации (сравнение предъявленного ТМ-идентификатора с перечнем зарегистрированных на ПЭВМ) и аутентификации (подтверждение подлинности) с защитой от раскрытия пароля. Для идентификации (аутентификации) пользователей в комплексе «Аккорд» используются интеллектуальные персональные идентификаторы DS 199X («Touch memory» – «память касания»), отличающиеся высокой надежностью, уникальностью, наличием быстродействующей памяти, удобством пользования, приемлемыми массо-габаритными характеристиками и низкой ценой.

В комплексе «Аккорд» реализован принцип дискреционного управления доступом. Зарегистрированному пользователю устанавливаются права доступа по принципу регистрации «белого списка» разрешенных к запуску программ (задач), которые прописываются в ПРД. При запросе пользователя на доступ, обеспечивается однозначное трактование установленных ПРД, и, в зависимости от уровня полномочий пользователя, разрешается или запрещается запрошенный тип доступа.

5.1.4.2 Подсистема регистрации и учета

Предназначена для регистрации в системном журнале различных событий, происходящих в ПЭВМ. При регистрации событий в системном журнале регистрируются:

- дата и время события;
- пользователь, осуществляющий регистрируемое действие;
- действия пользователя (сведения о входе/выходе пользователя из системы, запусках программ, событиях НСД, изменении полномочий и др.).

Доступ к системному журналу возможен только администратору СБИ (супервизору).

В системный журнал заносятся сведения более чем о 200 событиях, а также осуществляется архивация занесенных данных.

5.1.4.3 Подсистема обеспечения целостности

Предназначена для исключения несанкционированных модификаций (как случайных, так и злоумышленных) программной среды, в том числе программных средств комплекса, обрабаты-

ваемой информации, обеспечивая при этом защиту ПЭВМ от внедрения программных закладок и вирусов. В комплексе «Аккорд» это реализуется:

- проверкой целостности назначенных для контроля системных файлов, в том числе КСЗИ НСД, пользовательских программ и данных;
- контролем обращения к операционной системе напрямую, в обход прерываний DOS;
- исключением возможности использования ПЭВМ без контроллера комплекса;
- механизмом создания замкнутой программной среды, запрещающей запуск привнесенных программ и исключающей несанкционированный выход в ОС.

При проверке на целостность вычисляется контрольная сумма файлов и сравнивается с эталонным (контрольным) значением, хранящимся в ТМ-идентификаторе пользователя. Эти данные заносятся при регистрации пользователя и могут изменяться в процессе эксплуатации ПЭВМ. В комплексе «Аккорд» используется сложный алгоритм расчета контрольных сумм (вычисление значения их хэш-функций), исключающий факт необнаружения модификации файла. Эталонное (контрольное) значение хэш-функции контрольной суммы хранится вне ПЭВМ, в ТМ-идентификаторе пользователя, и этим защищается от несанкционированной модификации. Защита от модификации программы расчета хэш-функций обеспечивается тем, что она хранится в микросхеме ПЗУ контроллера комплекса.

5.1.5 Состав комплекса

Комплекс «Аккорд» включает программные и аппаратные средства.

5.1.5.1 Аппаратные средства

Аппаратные средства содержат:

- одноплатный контроллер (ТУ РБ 28591037.001-95), устанавливаемый в свободный слот материнской платы ПЭВМ;

- контактное устройство-съемник информации (4012-003-11443195-97 93). Устанавливается обычно на передней панели ПЭВМ в отверстие, высверливаемое в заглушке на зарезервированном месте для дисковода, либо в другом подходящем месте (в зависимости от модификации съемника). Предусматривается установка внешнего съемника. При этом подключение осуществляется к задней планке контроллера посредством разъема RJ-11;

- интеллектуальный персональный идентификатор DS 199X («Touch memory» – «память касания») – ТМ-идентификатор. Представляет собой полупассивное микропроцессорное устройство, снабженное элементом питания, в виде «таблетки» диаметром 16 мм и толщиной 3–5 мм в удобной пластмассовой (металлической) оправке. Каждый ТМ-идентификатор обладает уникальным номером (48 бит), который формируется технологически и подделать который практически невозможно. Объем памяти, доступной для записи и чтения, составляет до 64 Кбит в зависимости от типа идентификатора. Срок хранения записанной информации, обеспечиваемый элементом питания, составляет не менее 10 лет.

Количество и тип ТМ-идентификаторов, модификации контроллера и контактного устройства оговаривается при поставке комплекса.

5.1.5.2 Программные средства

Программное обеспечение СЗИ «Аккорд 1.95-00»:

ACED32.EXE	Редактор прав доступа
ACRUN.EXE	Монитор безопасности
ACCORD.SYS	Драйвер п/с защиты ПЭВМ от НСД
TMDRV.EXE	Драйвер контроллера
ACSETUP.EXE	Установка подсистемы И/А
ACCORD.RES	Библиотека ресурсов
ACLOGPP.EXE	Препроцессор журнала
ACLOG.EXE	Работа с журналом

TMTEST.EXE	Диагностика идентификаторов
MEMSCAN.EXE	Анализ памяти для установки джамперов
CHECSUM.EXE	Вычисление контрольных сумм
ACNED.EXE	Редактор прав доступа в сети
ACCON.EXE	Консоль наблюдения
ACCIPX.EXE	Сетевой драйвер консоли
ACRIPX.EXE	Сетевой драйвер станции
ACSHEDNW.EXE	Диспетчер прав доступа (планировщик)

Программный интерфейс к контроллеру комплекса, включающий в себя объектные модули и модули заголовков для Borland Pascal v.7.0 и Borland C++ v.3.1, а также примеры использования интерфейса.

5.1.6 Принцип работы комплекса

Плата контроллера комплекса «Аккорд» устанавливается в свободный слот материнской платы ЭВМ, производится установка программного обеспечения на жесткий диск, настройка комплекса, в том числе установление прав разграничения доступа, и регистрация пользователей. При регистрации пользователя администратором СБИ определяются его права доступа: списки исполняемых программ и модулей, разрешенных к запуску данным пользователем, и список стартовых (исполняемых непосредственно после загрузки ОС) программ и др. С помощью утилиты ACED32.EXE в специальные файлы данных вносятся списки файлов, целостность которых будет проверяться при запуске ПЭВМ данным пользователем. Значение хэш-функции (контрольной суммы) этих файлов прописывается в память персонального ТМ-идентификатора. После регистрации пользователю выдается на руки персональный ТМ-идентификатор, о чем делается запись в журнал учета носителей информации. Особенностью и, несомненно, преимуществом комплекса «Аккорд» является проведение процедур идентификации, аутентификации и контроля целостности защищаемых файлов до загрузки операционной системы. Это обеспечивается при помощи ПЗУ, установ-

ленного на плате контроллера комплекса, которое получает управление во время так называемой процедуры ROM-SCAN. Суть данной процедуры в следующем. В процессе начального старта после проверки основного оборудования BIOS компьютера начинает поиск внешних ПЗУ в диапазоне от С 800:0000 до Е000:0000 с шагом в 2К. Признаком наличия ПЗУ является наличие слова АА55Н в первом слове проверяемого интервала. Если данный признак обнаружен, то в следующем байте содержится длина ПЗУ в страницах по 512 байт. Затем вычисляется контрольная сумма всего ПЗУ, и если она корректна, то будет произведен вызов процедуры, расположенной в ПЗУ со смещением. Такая процедура обычно используется для инициализации. В комплексе «Аккорд» в этой процедуре проводится идентификация и аутентификация пользователя, и при ошибке возврат из процедуры не происходит, т.е. загрузка выполняться не будет.

ПЗУ контроллера «Аккорд» использует прерывание int 13h (дисковый ввод/вывод). Если в ПЭВМ установлен контроллер диска, имеющий ПЗУ, которое участвует в процедуре ROM-SCAN (например, SCSI-контроллеры, контроллеры с аппаратной памятью и т.п.), то ПЗУ такого контроллера должно иметь более младший адрес, чем ПЗУ « Аккорда».

При установленном и инсталлированном комплексе «Аккорд» загрузка компьютера осуществляется в следующем порядке.

1. BIOS компьютера выполняет стандартную процедуру POST (проверку основного оборудования компьютера) и по ее завершении переходит к процедуре ROM-SCAN, во время которой управление перехватывает контроллер комплекса «Аккорд». На монитор выводится сообщение:

«Access system BIOS v.1.xx copyright OKB SAPR 1993.– 1995_ s/n.....»

2. Выводится окно с приглашением пользователю предъявить свой ТМ-идентификатор:

«Attach key, please...»

Это окно остается на мониторе до момента контакта ТМ-идентификатора пользователя и съемника информации.

3. Если идентификатор не зарегистрирован, то выводится сообщение:

«Access denied!»

и происходит возврат к п.2.

4. При легальном ТМ-идентификаторе выводится окно с приглашением пользователю ввести пароль для аутентификации:

«Password»

5. При неправильно введенном пароле выводится сообщение:

«Access denied!»

и происходит возврат к п.2.

6. При правильно введенном пароле выводится сообщение:

«Access granted!»

и продолжается процедура загрузки DOS и т.д.

Вся процедура идентификации и аутентификации занимает 7–10 секунд. Устойчивость ее зависит от длины пароля. Допускается установка пароля от 3 до 12 символов. При осуществлении контрольных процедур (идентификации и аутентификации пользователя, проверке целостности) драйвер ACCORD.SYS блокирует клавиатуру и загрузку ОС с диска А. При касании съемника информации осуществляется поиск предъявленного ТМ-

идентификатора в списке зарегистрированных на ПЭВМ идентификаторов. Обычно список хранится на диске С.

Если предъявленный ТМ-идентификатор обнаружен в списке, то производится контроль целостности защищаемых по перечню данного пользователя файлов. При проверке перечня файлов пользователя на целостность программой CHECKSUM.EXE вычисляется хэш-функция контрольной суммы этих файлов и сравнивается с эталонным (контрольным) значением, считываемым из предъявленного персонального ТМ-идентификатора. Для проведения процедуры аутентификации предусмотрен режим ввода пароля в скрытом виде – в виде символов <*>. Этим предотвращается возможность раскрытия индивидуального пароля и использования утраченного (похищенного) ТМ-идентификатора. При положительном результате указанных выше контрольных процедур появляется сообщение «Access granted!» («Доступ разрешен») на зеленом фоне и производится загрузка DOS. Если предъявленный пользователем идентификатор не зарегистрирован в списке (сообщение «Неизвестный идентификатор») или нарушена целостность защищаемых файлов (сообщение «Нарушение целостности»), загрузка DOS не производится. Для продолжения работы потребуется вмешательство администратора СБИ. Таким образом, контрольные процедуры (идентификация, аутентификация, проверка целостности системных файлов ОС) осуществляются до загрузки ОС, при этом обеспечивается защита от РПВ. В любом другом случае, т.е. при неподтверждении прав пользователя на работу с данной ПЭВМ, загрузка DOS не выполняется. При выполнении модифицированных администратором СБИ в процессе установки комплекса файлов CONFIG.SYS и AUTOEXEC.BAT производится блокировка клавиатуры, загрузка модуля ACRUN.EXE, осуществляющего контроль за использованием пользователем только разрешенных ему ресурсов и запускающего (на основании проведенной идентификации/ аутентификации) стартовую пользовательскую задачу. В процессе работы пользователя программа ACRUN.EXE препятствует любым видам НСД к файлам CONFIG.SYS и AUTOEXEC.BAT.

Механизм контроля целостности реализуется процедурой сравнения двух векторов для одного массива данных: эталонного (контрольного), выработанного заранее на этапе регистрации

пользователей, и текущего, выработанного непосредственно перед проверкой. Эталонный (контрольный) вектор вырабатывается на основе хэш-функций (контрольной суммы) защищаемых файлов и хранится в идентификаторе. В случае санкционированной модификации защищенных файлов осуществляется процедура перезаписи в идентификатор нового значения хэш-функции (контрольной суммы) модифицированных файлов, для чего на экране выдается сообщение «Прислоните ТМ-идентификатор» с последующим подтверждением успешной (неуспешной) перезаписи значения хэш-функции в персональный идентификатор пользователя. В процессе функционирования комплекса резидентная часть «монитора безопасности» проверяет файлы всех загруженных из файла CONFYG.SYS драйверов и обеспечивает оперативный контроль целостности исполняемых файлов перед передачей им управления. Тем самым обеспечивается защита от программных вирусов и закладок. В случае положительного исхода проверки управление передается ОС для загрузки файла на исполнение. При отрицательном исходе проверки запуск программы не происходит.

Кроме того, «монитор безопасности» ограничивает доступ к файлам ПО комплекса, которые расположены в каталоге C:\ACCORD, запрещая пользователю их переименование, уничтожение, изменение (запись и редактирование). Таким же образом защищены и файлы AUTOEXEC.BAT и CONFIG.SYS (поскольку удаление из них вызовов программной части комплекса может привести к возможности НСД). Для защиты от извлечения платы контроллера комплекса используется специальный механизм, обеспечивающий выполнение нормальной загрузки DOS только при наличии платы. При отсутствии платы загрузка DOS не осуществляется.

Работа программы с журналами регистрации приведена в приложении 3.

5.2 Программно-аппаратный комплекс Secret Net NT 4.0

5.2.1 Функциональные возможности системы

Автономный вариант системы защиты информации Secret Net NT 4.0 предназначен для защиты ресурсов рабочей станции локальной сети или неподключенного к сети компьютера и разработан научно-инженерным предприятием «ИНФОРМЗАЩИТА».

Система Secret Net NT 4.0 дополняет стандартные защитные механизмы ОС Windows NT функциями, обеспечивающими:

- идентификацию пользователей при помощи специальных аппаратных средств (Touch Memory, Smart Card, Smarty);
- дополнительно к избирательному (дискреционному) управлению доступом, реализованному в ОС Windows NT, полномочное (мандатное) управление доступом пользователей к конфиденциальной информации на локальных и подключенных сетевых дисках;
- оперативный контроль работы пользователей компьютера путем регистрации событий, связанных с безопасностью ИС, удобные средства просмотра и представления зарегистрированной информации;
- контроль целостности программ, используемых пользователями и операционной системой;
- возможность создания для любого пользователя замкнутой программной среды (списка разрешенных для запуска программ);
- простоту управления объектами благодаря использованию механизма шаблонов настроек.

5.2.2 Общая архитектура

Система Secret Net NT включает в себя следующие компоненты и подсистемы:

- ядро системы защиты (1);
- подсистема управления (4);
- подсистема криптографической защиты информации (5);
- база данных системы защиты (6);

- подсистема избирательного управления доступом (9);
- подсистема разграничения доступа к дискам (10);
- подсистема разграничения полномочного доступа (11);
- подсистема замкнутой программной среды (12);
- подсистема контроля целостности (14);
- подсистема контроля входа (16).

На рис. 1 приведена обобщенная структура автономного варианта системы защиты Secret Net NT, представлены основные элементы и взаимосвязи между ними.

5.2.3 Основные компоненты

Ядро системы защиты (1) представляет собой программу, которая автоматически запускается на защищенном компьютере при его включении и функционирует на протяжении всего времени работы компьютера. Ядро системы осуществляет управление подсистемами и компонентами системы защиты и обеспечивает их взаимодействие.

В процессе работы системы защиты ядро выполняет следующие функции:

- обеспечивает обмен данными между компонентами системы и обработку команд, поступающих от этих компонент;
- обеспечивает доступ других компонент системы к информации, хранящейся в базе данных системы защиты;
- осуществляет сбор сведений о состоянии компьютера;
- контролирует доступ пользователя к ресурсам компьютера;
- обрабатывает информацию, поступающую от компонент системы защиты, о событиях, происходящих на компьютере и связанных с безопасностью системы, и осуществляет их регистрацию в журнале безопасности ОС Windows NT.

Подсистема регистрации (3) является одним из элементов ядра системы и предназначена для управления регистрацией в журнале безопасности Windows NT (8) событий, связанных с работой ОС и Secret Net. Эта информация поступает от отдельных подсистем системы защиты, которые следят за происходящими в информационной среде событиями. Регистрация событий осуще-

ствляется системными средствами (ОС Windows NT) или средствами системы защиты Secret Net NT. Перечень регистрируемых событий устанавливается администратором с помощью подсистемы управления (4). Для просмотра журнала используется специальная программа подсистемы управления, обладающая развитыми средствами работы с журналами регистрации.

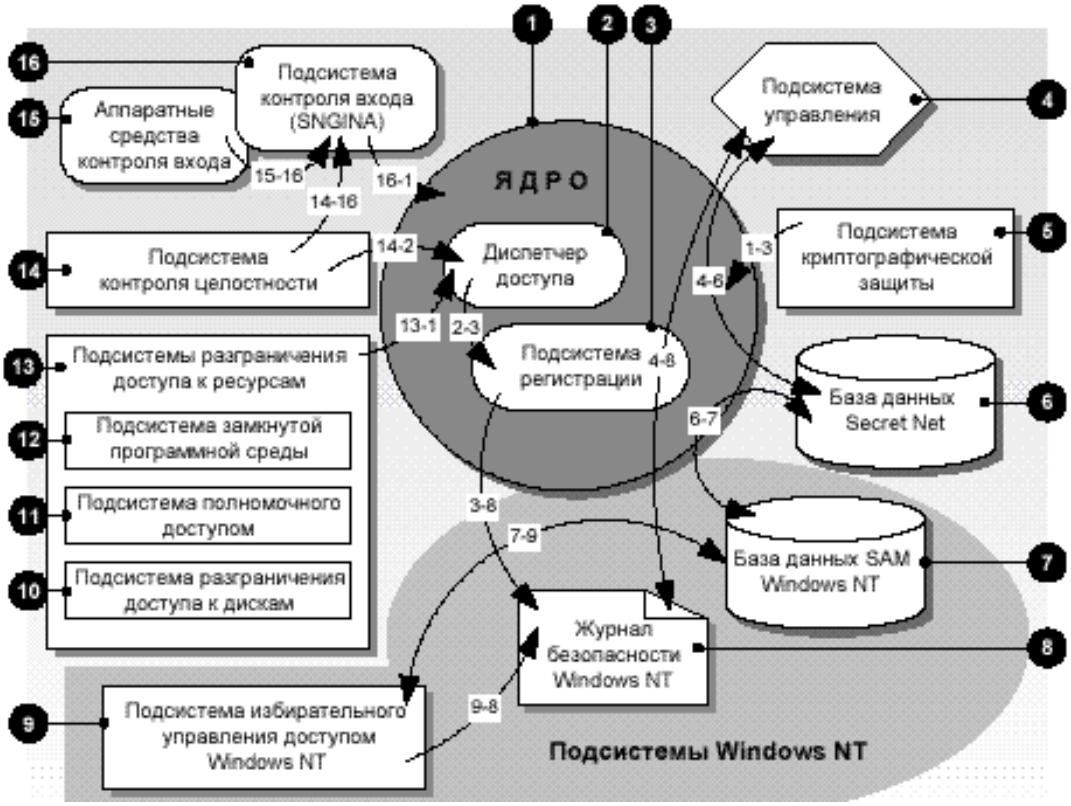


Рис. 1 – Архитектура автономного варианта системы Secret Net NT

Подсистема управления (4) располагает средствами для настройки защитных механизмов через управление общими параметрами работы компьютера, свойств пользователей и групп пользователей. В частности она обеспечивает:

- отображение и управление состоянием защищаемого компьютера;
- управление пользователями, настройками компьютера и сохранение относящихся к ним данных в БД системы защиты (6);
- получение информации из БД системы защиты;
- обработку и представление информации из журнала безопасности ОС Windows NT (8).

В состав подсистемы управления входит программа, предназначенная для просмотра журнала безопасности и подготовки отчетов. С ее помощью можно выполнить просмотр, отбор, сортировку, поиск записей, печать, экспорт журнала в другие форматы.

База данных Secret Net (6) предназначена для хранения сведений, необходимых для работы защищенного компьютера. БД Secret Net размещается в реестре ОС Windows NT и содержит информацию об общих настройках системы защиты, свойствах пользователей и групп пользователей.

Доступ подсистем и компонент системы защиты к данным, хранящимся в БД Secret Net, обеспечивается ядром системы защиты (1).

Первоначальное заполнение БД выполняется при установке Secret Net. Для этого используются данные, содержащиеся в БД безопасности Windows NT (политика безопасности, состав пользователей и групп пользователей и т.д.), и данные, устанавливаемые по умолчанию для Secret Net (значения общих параметров, некоторые свойства пользователей, набор шаблонов и т.д.).

Синхронизацию данных в БД безопасности Windows NT и БД Secret Net.обеспечивает ядро системы защиты (1). В дальнейшем информация, содержащаяся в БД, создается и модифицируется подсистемой управления (4) и другими подсистемами.

Подсистема избирательного управления доступом (9) обеспечивает разграничение доступа пользователей к ресурсам файловой системы, аппаратным ресурсам и ресурсам операционной системы компьютера.

Для управления доступом к ресурсам файловой системы, системному реестру и системным средствам управления компьютером используются стандартные средства ОС Windows NT, а непосредственное управление осуществляется с использованием интерфейса Secret Net NT. Для управления доступом к остальным ресурсам (дискам и портам) используются средства Secret Net NT.

Подсистема полномочного управления доступом (11) обеспечивает разграничение доступа пользователей к конфиденциальной информации, хранящейся в файлах на локальных и сетевых дисках. Доступ осуществляется в соответствии с категорией конфиденциальности, присвоенной информации, и уровнем допуска пользователя к конфиденциальной информации.

Подсистема полномочного управления доступом включает в себя драйвер полномочного управления доступом и компоненту управления допуском к ресурсам.

Компонента управления конфиденциальностью ресурсов включается в программу «Проводник» (Explorer). Из программы «Проводник» и осуществляется управление категориями конфиденциальности, которые присваиваются файлам, каталогам и дискам компьютера. Диски обязательно должны быть размечены для работы с файловой системой NTFS.

Драйвер полномочного управления доступом контролирует доступ пользователей к конфиденциальным ресурсам. Когда пользователь (или программа, запущенная пользователем) осуществляет попытку выполнить какую-либо операцию над конфиденциальным ресурсом, драйвер определяет категорию конфиденциальности ресурса и передает ее диспетчеру доступа (2), входящему в состав ядра системы защиты. Диспетчер доступа сопоставляет категорию конфиденциальности ресурса и уровень допуска данного пользователя к конфиденциальной информации. Также он проверяет, не противоречат ли действия пользователя с ресурсом другим настройкам системы (например, условиям копирования через буфер обмена). Если уровень допуска или настройки системы не позволяют выполнить операцию – диспетчер доступа передает драйверу запрещающую команду, и операция блокируется. При этом подсистема регистрации (3) ядра системы фиксирует в журнале попытку несанкционированного доступа.

Подсистема замкнутой программной среды (11) позволяет сформировать для любого пользователя компьютера программную среду, определив индивидуальный перечень программ, разрешенных для запуска.

Драйвер замкнутой программной среды контролирует запуск пользователем программ. Когда пользователь (программа,

запущенная пользователем) осуществляет попытку запуска какой-либо программы, драйвер передает диспетчеру доступа (2), входящему в состав ядра системы защиты, сведения о запускаемой программе. Диспетчер доступа проверяет, включена ли эта программа в персональный список программ, разрешенных для запуска. Если программа содержится в списке, диспетчер доступа передает драйверу разрешающую команду. Если пользователю запрещено запускать данную программу, диспетчер доступа передает драйверу запрещающую команду, и запуск программы блокируется. В этом случае подсистема регистрации (3) фиксирует в журнале безопасности попытку несанкционированного доступа.

Подсистема контроля входа (16) обеспечивает идентификацию и аутентификацию пользователя при его входе в систему. Подсистема включает в себя модуль идентификации пользователя, а также может содержать средства аппаратной поддержки, например, Secret Net TM Card или электронный замок «Соболь», если они установлены на компьютере, и программу-драйвер, с помощью которой осуществляется управление аппаратными средствами.

Подсистема контроля входа запрашивает и получает информацию о входящем в систему пользователе (имя, пароль, персональный идентификатор, личный ключ пользователя). Затем сравнивает полученную информацию с информацией, хранящейся в БД системы защиты. Предоставление информации из БД обеспечивает ядро системы защиты. Если в БД отсутствует информация о пользователе, процедура загрузки системы прекращается.

Для целей идентификации и аутентификации могут использоваться аппаратные средства. Для управления ими необходимы специальные программы-драйверы, которые обеспечивают обмен информацией между устройствами аппаратной поддержки и модулями системы защиты. Драйверы входят в комплект поставки и устанавливаются на компьютер вместе с системой Secret Net NT.

При загрузке компьютера подсистема контроля целостности (14) проверяет целостность системных файлов. Если целостность файлов не нарушена, подсистема контроля целостности передает

управление подсистеме опознавания пользователя. В случае нарушения целостности файлов загрузка системы может быть запрещена.

Подсистема контроля целостности (14) осуществляет слежение за неизменностью контролируемых объектов (файлов, ключей системного реестра и т.д.) с целью защиты их от модификации. Для этого определяется перечень контролируемых объектов. Для каждого из входящих в него объектов рассчитываются эталонные контрольные суммы. Вычисления проводятся с использованием хэш-функций (в соответствии с ГОСТ Р 34-10) или по оригинальному (быстрому) алгоритму собственной разработки. Эталонные контрольные суммы проверяемых объектов и информация об их размещении хранятся в пакетах контроля целостности.

Целостность объектов контролируется в соответствии с установленным расписанием. Подсистема контроля входа (16) передает подсистеме контроля целостности **Secret Net NT 4.0.** перечень контролируемых объектов и порядок их контроля при запуске компьютера.

Ядро системы (1) передает подсистеме контроля целостности расписание контроля, составленное администратором с помощью подсистемы управления (4). В соответствии с расписанием контроля, вычисляются контрольные суммы проверяемых объектов и сравниваются с ранее вычисленными их эталонными значениями.

Если выявляется нарушение целостности объектов, подсистема контроля целостности сообщает об этом диспетчеру доступа (2).

5.2.4 Защитные механизмы Secret Net NT 4.0

Система Secret Net NT дополняет операционную систему Windows NT рядом защитных средств, которые можно отнести к следующим группам:

5.2.4.1 Средства защиты от несанкционированного входа в систему:

- механизм идентификации и аутентификации пользователей (в том числе с помощью аппаратных средств защиты);
- функция временной блокировки компьютера на время паузы в работе для защиты компьютера от использования посторонним лицом;
- функция программной защиты от загрузки ОС с гибкого диска;
- аппаратные средства защиты от загрузки ОС с гибкого диска.

5.2.4.2 Средства управления доступом и защиты ресурсов:

- разграничение доступа пользователей к ресурсам компьютера с использованием механизмов избирательного и полномочного управления доступом;
- создание для любого пользователя замкнутой программной среды (списка разрешенных для запуска программ);
- средства криптографической защиты данных: шифрование информации, хранящейся в файлах на сетевых и локальных дисках; вычисление и проверка электронной цифровой подписи (ЭЦП).

5.2.4.3 Средства регистрации и оперативного контроля:

- политика регистрации, ведение журнала регистрации событий, имеющих отношение к безопасности системы, работа с журналами, управление временем хранения и удалением записей;
- контроль целостности файлов, управление расписанием контроля и выбор реакции на нарушение целостности;
- контроль аппаратной конфигурации компьютера.

Отличительной особенностью системы Secret Net NT является возможность гибкого управления набором защитных средств системы. Пользователь, имеющий привилегии на администрирование системы, может активизировать различные комбинации защитных механизмов системы, выбирая из них только необходимые и устанавливая соответствующие режимы их работы.

5.2.5 Механизмы контроля входа в систему

Защита от несанкционированного входа предназначена для предотвращения доступа посторонних лиц к защищенному компьютеру. К этой группе средств, как уже говорилось, могут быть отнесены:

- программные и аппаратные средства идентификации и аутентификации;
- функция временной блокировки компьютера;
- аппаратные средства защиты от загрузки ОС с гибкого диска.

5.2.6 Механизм идентификации и аутентификации пользователей

Идентификация и аутентификация пользователей выполняется при каждом входе пользователя в систему. При загрузке компьютера система Secret Net NT запрашивает у пользователя его идентификатор и пароль. Затем проверяется, был ли зарегистрирован в системе пользователь с таким именем и правильно ли указан его пароль. В качестве идентификаторов могут использоваться: уникальные имена и уникальные номера аппаратных устройств идентификации (персональных идентификаторов).

В Secret Net NT поддерживается работа с паролями длиной до 16 символов. Если пароль указан неверно, подается звуковой сигнал и в журнале безопасности регистрируется попытка несанкционированного доступа к компьютеру. При определенном числе неверных попыток ввода пароля происходит блокировка компьютера.

Идентификаторы пользователей (имена и номера аппаратных идентификаторов) хранятся в базе данных системы защиты в открытом виде, а пароли пользователей – в кодированном виде.

5.2.7 Аппаратные средства защиты от несанкционированного входа

Средства аппаратной поддержки в системах защиты предназначены для:

- запрета загрузки ОС со съемных носителей (гибких и компакт-дисков);

- идентификации пользователей системы защиты до загрузки ОС;
- поддержки различных аппаратных идентификаторов (например, Touch Memory), заменяющих ввод идентифицирующей информации с клавиатуры.

Работу системы защиты с аппаратными средствами обеспечивают специальные программы-драйверы, управляющие обменом информацией между устройством и модулями системы защиты.

В системе Secret Net NT предусмотрено несколько режимов идентификации и аутентификации с использованием аппаратных средств. Это дает возможность проводить их внедрение поэтапно. При «мягком» режиме работы любой пользователь может войти в систему либо предъявив персональный идентификатор, либо указав свое имя. При «жестком» режиме вход в систему любого пользователя разрешен только при предъявлении персонального идентификатора.

5.2.8 Функция временной блокировки компьютера

Функция временной блокировки компьютера предназначена для предотвращения использования компьютера посторонними лицами. В этом режиме блокируются устройства ввода (клавиатура и мышь) и экран монитора (хранителем экрана). Включить режим временной блокировки компьютера может сам пользователь, нажав определенную, заданную им, комбинацию клавиш.

Компьютер может быть заблокирован и автоматически после некоторого периода простоя. Длительность этого интервала устанавливается настройкой соответствующих параметров. Вывести компьютер из режима блокировки можно, только если вновь указать пароль или предъявить персональный идентификатор.

5.2.9 Механизмы управления доступом и защиты ресурсов

Система Secret Net NT включает в свой состав несколько механизмов управления доступом пользователей к ресурсам компьютера:

- механизм избирательного управления доступом;
- механизм полномочного управления доступом;

➢ механизм замкнутой программной среды.

Все ресурсы компьютера в системе Secret Net NT делятся на три типа:

Ресурсы файловой системы – локальные логические диски и размещающиеся на них каталоги и файлы.

Аппаратные ресурсы – локальные и сетевые принтеры, коммуникационные порты, физические диски, дисководы, приводы CD-ROM.

Ресурсы операционной системы – системные файлы, ключи системного реестра, системное время, диалоги настройки параметров системы.

Механизмы полномочного управления доступом и механизм замкнутой программной среды применяются только к ресурсам файловой системы.

5.2.10 Механизм избирательного управления доступом

Управление избирательным доступом к локальным ресурсам компьютера осуществляется на основании предоставления пользователям компьютера прав и привилегий.

В Secret Net NT для управления доступом к ресурсам файловой системы, системному реестру и системным средствам управления компьютером используются стандартные механизмы ОС Windows NT.

Примечание. Подробные сведения о механизме избирательного управления доступом в ОС Windows NT можно найти в документации к ОС. Для управления доступом к дискам и портам используются собственные механизмы системы Secret Net NT.

5.2.11 Механизм полномочного управления доступом

Система Secret Net NT включает в свой состав средства, позволяющие организовать полномочное (мандатное) управление доступом пользователей к конфиденциальной информации. Полномочное управление доступом осуществляется только по отношению к каталогам и распространяется на все файлы и подкаталоги, находящиеся в них. При организации полномочного управ-

ления доступом для каждого пользователя компьютера устанавливается некоторый уровень допуска к конфиденциальной информации, определяющий его права на доступ к конфиденциальным данным. Всем файлам и каталогам, находящимся на локальных дисках и подключенных сетевых дисках компьютера, назначается категория конфиденциальности. Используются три категории конфиденциальности информации: «Нет» (для общедоступной информации), «Конфиденциально», «Строго конфиденциально».

Доступ к конфиденциальным каталогам и находящимся в них файлам осуществляется следующим образом. Когда пользователь (программа, запущенная пользователем) осуществляет попытку доступа к конфиденциальному каталогу или находящемуся в нем файлу, диспетчер доступа Secret Net NT определяет категорию конфиденциальности данного ресурса. Затем категория конфиденциальности ресурса сопоставляется с уровнем допуска пользователя к конфиденциальной информации. Если текущий пользователь не превышает свой уровень допуска, система защиты санкционирует доступ к ресурсу. Иначе система защиты блокирует доступ к ресурсу.

При работе системы Secret Net NT в режиме полномочного управления доступом контролируются потоки конфиденциальной информации. Это позволяет, например, предотвратить копирование конфиденциальных документов в неконфиденциальные области дисков и запретить свободный доступ к принтерам и коммуникационному оборудованию. Печать конфиденциальных документов в этом случае осуществляется только стандартными средствами Secret Net и фиксируется в системном журнале.

5.2.12 Механизм замкнутой программной среды

Механизм замкнутой программной среды позволяет без использования системы атрибутов ограничить доступ пользователей к исполняемым файлам только теми программами, которые действительно необходимы ему для выполнения своих служебных обязанностей.

Режим замкнутой программной среды может быть активизирован избирательно для тех или иных пользователей. Преду-

смотрена возможность двух режимов работы этого механизма – «жесткого» и «мягкого». При «мягком» режиме пользователю разрешается запускать программы, не внесенные в список разрешенных для запуска, но при этом в системном журнале регистрируются соответствующие события несанкционированного доступа (НСД). При «жестком» режиме запуск любой программы, не внесенной в список разрешенных для запуска программ, будет блокироваться, а попытка запуска будет регистрироваться как событие НСД.

Примечание. В Secret Net NT средства защиты могут работать в двух режимах: «жестком» и «мягком». «Жесткий» режим является основным режимом работы системы защиты. Использование «мягкого» (технологического) режима облегчает настройку системы защиты при вводе ее в эксплуатацию. Анализируя случаи НСД, зарегистрированные при работе в этом режиме, администратор безопасности может, не ограничивая потребности пользователя в ресурсах, выявить и конкретизировать их.

Перечень программ, разрешенных для запуска, определяется индивидуально для каждого пользователя. Список может быть сформирован автоматически на основании сведений об используемых программах из системного журнала (в условиях «мягкого» режима работы) и отредактирован средствами специального редактора.

Сформированные списки разрешенных для запуска программ хранятся в файлах в подкаталоге UEL каталога, в который была установлена система Secret Net. Файлы имеют расширение .uel и имя, совпадающее с регистрационным номером пользователя в системе. Uel-файл – это обычный текстовый файл, содержащий в каждой строке полный путь к файлу программы, запуск которой разрешен.

5.2.13 Механизмы контроля и регистрации

Система Secret Net NT включает в свой состав следующие средства контроля:

- механизм регистрации событий;
- механизм контроля целостности.

5.2.14 Механизм регистрации событий

В процессе работы системы Secret Net NT события, происходящие на компьютере и связанные с безопасностью системы, регистрируются в журнале безопасности Windows NT. Относительно каждого события фиксируется следующая информация:

- дата и время, определяющие момент наступления события;
- идентификатор пользователя, действия которого привели к появлению события;
- краткая характеристика события;
- имя программы, работа которой привела к появлению события;
- ресурс, при работе с которым произошло событие.

В общей сложности в системный журнал заносятся сведения более чем о ста видах событий.

Механизм регистрации событий обладает гибкими возможностями управления. Для каждого пользователя можно определить индивидуальный режим регистрации. От общего объема регистрируемых событий зависит размер системного журнала и, соответственно, время записи и последующего анализа событий.

Для системного журнала может быть установлен предельный срок хранения регистрационных записей, по истечении которого устаревшие записи будут автоматически удаляться из журнала. Право на настройку режимов регистрации событий предоставляется пользователю посредством соответствующих привилегий на администрирование системы.

5.2.15 Механизм контроля целостности

Контроль целостности предназначен для слежения за изменениями характеристик выбранных объектов информационной среды. Объектами контроля могут быть: секторы дисков, файлы, каталоги, элементы реестра, ветви и настройки сервисов.

Каждый тип объектов имеет свой набор контролируемых данных. Так, например, файлы могут контролироваться на целостность: содержимого, прав доступа, атрибутов и существования.

Кроме того, для каждого из типов объектов могут использоваться различные алгоритмы контроля целостности.

В системе предусмотрена гибкая возможность выбора времени контроля. В частности, контроль может быть выполнен при загрузке ОС (средствами электронного замка «Соболь»), при входе или выходе пользователя из системы по заранее составленному расписанию. Кроме того, может быть проведен и немедленный контроль.

При обнаружении несоответствия предусмотрены следующие варианты реакции на возникающие ситуации подсистемы контроля целостности:

- регистрация изменений в системном журнале;
- оповещение администратора безопасности о произошедших изменениях;
- блокировка компьютера;
- отклонение или принятие изменений.

Для каждого типа контролируемых объектов на рабочей станции хранятся список имен объектов и задания для контроля тех или иных характеристик указанных объектов. Эта информация размещается в базе данных подсистемы контроля целостности, которая реализована в виде набора файлов определенного формата, расположенных в отдельном каталоге. База данных содержит всю необходимую информацию для функционирования подсистемы. Задание на контроль содержит необходимую информацию об эталонном состоянии объекта, порядке контроля характеристик и действий, которые надо выполнить при обнаружении изменений. Результаты контроля и обработки запросов фиксируются в системном журнале.

Подсистема контроля целостности взаимодействует с другими подсистемами через ядро системы защиты. Для просмотра и редактирования списков контроля целостности, режимов контроля и номенклатуры контролируемых объектов используется подсистема управления (4). Кроме того, подсистема контроля целостности самостоятельно выполняет контроль объектов и взаимодействует для выполнения различных действий со следующими подсистемами Secret Net:

- подсистемой контроля входа (16) – для оповещения о входе или выходе пользователя из системы;

- подсистемой аппаратной поддержки (15) – для получения доступа к аппаратным средствам контроля;
- подсистемой регистрации (3) – для записи сообщений в системный журнал;
- подсистемой криптографической защиты (5) – для выполнения криптографических операций при контроле целостности.

Подсистема контроля целостности используется в нескольких типичных случаях:

- для контроля в автоматическом режиме целостности объектов по установленному расписанию;
- для выполнения внеплановых проверок по инициативе администратора;
- для обработки запросов от программы управления с целью просмотра и изменения характеристик контролируемых объектов.

5.2.16 Контроль аппаратной конфигурации компьютера

Контроль аппаратной конфигурации компьютера предназначен для своевременного обнаружения изменений конфигурации и выбора наиболее целесообразного способа реагирования на эти изменения. Изменения аппаратной конфигурации компьютера могут быть вызваны выходом из строя, добавлением или заменой отдельных устройств.

Для эффективного контроля конфигурации используется широкий набор контролируемых параметров, с каждым из которых связаны правила обнаружения изменений и действия, выполняемые в ответ на эти изменения.

Сведения об аппаратной конфигурации компьютера хранятся в БД системы защиты. Первоначальные («эталонные») данные о конфигурации поступают от программы установки. Каждый раз при загрузке компьютера, а также при повторном входе пользователя система получает сведения об актуальной аппаратной конфигурации и сравнивает ее с эталонной.

Контроль конфигурации программных и аппаратных средств производится ядром системы Secret Net. По результатам контроля ядро принимает решение о необходимости блокировки компьютера. Решение принимается после входа пользователя и

зависит от настроек пользователя. Значение настроек пользователя определяет администратор безопасности.

Если было выполнено запланированное изменение конфигурации компьютера, то пользователь, обладающий административными привилегиями, может при помощи подсистемы управления обновить эталонные сведения о конфигурации.

5.2.17 Средства аппаратной поддержки Secret Net

В качестве средств аппаратной поддержки в Secret Net могут быть использованы следующие устройства:

Secret Net ROM BIOS – микросхема с расширением BIOS, устанавливается на сетевой карте компьютера в гнездо для микросхемы удаленной загрузки. Обеспечивает идентификацию с помощью электронных идентификаторов Touch Memory, считыватели которых подключены к COM -порту.

Secret Net Touch Memory Card – плата с разъемом для подключения считывателя Touch Memory или считывателя бесконтактных радиокарт Proximity, устанавливаемая внутри компьютера в разъем ISA. Обеспечивает идентификацию пользователей по электронным идентификаторам Touch Memory или картам Proximity.

Контроллер «Соболь» – плата с разъемом для подключения считывателя Touch Memory, аппаратным датчиком случайных чисел, 2-мя (4-мя) каналами физической блокировки устройств и внутренней энергонезависимой памятью. Устанавливается внутри компьютера в разъем ISA или PCI. Является основой системы Электронный замок «Соболь». В системе Secret Net может быть использован для идентификации пользователей по электронным идентификаторам Touch Memory, а также для генерации криптографических ключей.

Считыватель бесконтактных радиокарт Proximity – устройство, подключаемое к разъему Secret Net Touch Memory Card и устанавливаемое внутри корпуса компьютера. В системе Secret Net считыватель используется для идентификации пользователей по картам Proximity.

5.3 Порядок аттестации автоматизированных систем обработки информации

Аттестация автоматизированных систем обработки информации по требованиям информационной безопасности проводится после завершения ее создания и квалификации уровня безопасности перед вводом в эксплуатацию. Порядок проведения аттестации, контроль и надзор за аттестацией и эксплуатацией аттестованных объектов регламентирует Положение по аттестации объектов информатизации.

Под аттестацией объектов информатизации понимают комплекс организационно-технических мероприятий, в результате которых специальным документом – Аттестатом соответствия подтверждается, что объект соответствует требованиям стандартов или иных нормативно-технических документов по безопасности информации.

Наличие на объекте информатизации действующего Аттестата соответствия дает право обрабатывать конфиденциальную информацию в течение периода времени, установленного сроком действия Аттестата соответствия.

При аттестации объекта информатизации подтверждается его соответствие требованиям по защите информации от несанкционированного доступа, компьютерных вирусов, от утечки за счет побочных электромагнитных излучений и наводок при специальных воздействиях на защищаемый объект.

Контроль за эффективностью реализованных мер и средств защиты информации в автоматизированных системах учреждения возлагается на службу информационной безопасности учреждения.

ЛИТЕРАТУРА

1. Зегжда Д.П., Ивашко А.М. Основы безопасности информационных систем. – М.: Горячая линия – Телеком, 2000. – 452 с.: ил.

2. Закон Российской Федерации «Об информации, информатизации и защите информации» от 25.01.95 г.

3. РД. АС. Защита от НСД к информации. Классификация АС и требования по защите информации. – М.: Гостехкомиссия России, 1992.

4. Анин Б.Ю. Защита компьютерной информации. – СПб.: БХВ – Санкт-Петербург, 2000. – 384 с.: ил.

5. Щербаков А.Ю. Введение в теорию и практику компьютерной безопасности. – М.: Издатель С.В. Молгачева, 2001. – 352 с.: ил.

6. Защита программного обеспечения / Под ред. Д. Гроувера. – М., 1992. – 289 с.: ил.

7. Романец Ю.В., Тимофеев П.А., Шаньгин В.Ф. Защита информации в компьютерных системах и сетях / Под ред. В.Ф. Шаньгина. – 2-е изд., перераб. и доп. – М.: Радио и связь, 2001 с.: ил.

8. Чижухин Г.Н. Основы защиты информации в вычислительных системах и сетях ЭВМ: Учеб. пособие. – Пенза: Изд-во Пенз. гос. ун-та, 2001. – 164 с.

9. Бабенко Л.К., Ищуков С.С., Макаревич О.Б. Защита информации с использованием смарт-карт и электронных брелоков. – М.: Гелиос АРВ, 2003. – 352 с.: ил.

ПРИЛОЖЕНИЕ 1

Ранжированные функциональные требования «Федеральных критериев безопасности информационных технологий»

Данное приложение представляет собой согласно [1] ранжированный перечень функциональных требований, содержащийся в соответствующем разделе «Федеральных критериев». Оно отражает принципиальную суть требований и не претендует на перевод стандарта как руководящего документа.

Описание требований следует приведенным в разд. 2.8 требованиям в порядке возрастания уровня обеспечиваемой защиты. Описание каждого раздела требований начинается с его краткого описания и обзора предусмотренных уровней ранжирования, идентификаторы уровней сохранены в том виде, в каком они присутствуют в первоисточнике. Описание требований на каждом уровне приводится и виде дополнений и изменений по сравнению с предыдущем уровнем.

1. *Идентификация и аутентификация*

Идентификация и аутентификация принадлежат к основным компонентам политики безопасности. Отсутствие или низкая надежность процедур идентификации и/или аутентификации не позволяет противостоять атакам неавторизованных субъектов путем предотвращения их регистрации в системе и отказа в получении доступа к ее ресурсам. Надежность механизмов идентификации/аутентификации напрямую влияет на уровень безопасности всей системы в целом. При необходимости эти процедуры могут быть усилены совместным применением нескольких механизмов идентификации и аутентификации.

Требования идентификации и аутентификации ранжируются на основании уровня предоставляемых возможностей. Уровень **IA-1** включает только аутентификацию пользователей и предназначен для простейших систем, типа систем контроля доступа в помещения, в которых, кроме идентификации и аутентификации, реализована только функция регистрации входа. На уровне **IA-2** предполагается наличие специальных атрибутов (признаков), идентифицирующих конкретного субъекта и позволяющих выполнить его авторизацию (предоставить ему соответствующие права для работы в системе). Этот уровень наиболее широко ис-

пользуется в операционных системах, в которых существуют атрибуты, определяющие степень привилегированности субъектов и уровень конфиденциальности объектов. Данные возможности расширяются на уровне **IA-3** путем регламентации принципов обработки результатов аутентификации, а также требованиями к хранению, защите и предоставлению информации о результатах идентификации и аутентификации пользователей. Этот уровень используется в системах с четко определенной политикой управления доступом. На уровне **IA-4** происходит дальнейшее расширение требований, заключающееся в предоставлении возможностей установки специальных механизмов идентификации и аутентификации и их назначения для индивидуального пользователя и/или группы пользователей. На уровне **IA-5** требуется реализация нескольких независимых механизмов аутентификации пользователей.

Уровень **IA-1**. Минимальная идентификация и аутентификация.

1. ТСВ должна обеспечивать возможность идентификации каждого пользователя до начала выполнения им любых действий. ТСВ должна устанавливать индивидуальные полномочия пользователя в соответствии с его уникальным идентификатором. ТСВ должна обеспечивать возможность установления соответствия каждого регистрируемого в системе события с идентификатором инициировавшего его пользователя.
2. ТСВ должна использовать механизм аутентификации по паролю для проверки подлинности соответствия пользователя и его идентификатора.
3. ТСВ должна обеспечивать защиту данных, используемых для идентификации и аутентификации, с целью предотвращения доступа к ним неуполномоченных пользователей.

Уровень **IA-2**. Идентификация, аутентификация и авторизация.

1. *Без изменений.*

2. *Изменение.* Используемые для аутентификации данные должны включать информацию, необходимую как для проверки подлинности пользователя (например, пароль), так и для реали-

зации политики безопасности (атрибуты пользователей, имена групп и ролей, уровни привилегированности субъектов и конфиденциальности объектов, временные интервалы и т.д.). Эти данные должны использоваться ТСВ для контроля действий пользователя в соответствии с действующей в системе политикой безопасности.

3. *Без изменений.*

Уровень **IA-3**. Идентификация и аутентификация с контролем исключительных ситуаций.

1. *Без изменений.*

2. *Без изменений.*

3. *Дополнение.* ТСВ должна обеспечивать выполнение процедуры аутентификации вне зависимости от результата проведения процедуры идентификации (например, требовать пароль даже в случае ввода несуществующего имени пользователя).

ТСВ должна прекращать выполнение процедуры входа в систему в случае неудачного проведения идентификации и/или аутентификации пользователя последовательно произошедших заданное администратором число раз. В этом случае ТСВ должна оповестить администратора, зафиксировать данное событие в системном журнале и приостановить дальнейшее выполнение процедуры входа в систему на время, определенное администратором, или отказать пользователю в доступе вообще.

4. ТСВ должна обеспечивать хранение, защиту и предоставление информации о статусе и атрибутах всех активных пользователей, зарегистрированных в системе, и о бюджетах всех пользователей, существующих в системе.

Уровень **IA-4**. Назначение процедур идентификации и аутентификации.

1. *Дополнение.* ТСВ должна обеспечивать возможность идентификации каждого привилегированного субъекта.

2. *Дополнение.* ТСВ должна обеспечивать возможность внедрения новых механизмов аутентификации (например, на основе электронных карт или биометрических характеристик), дополняющих уже существующие. ТСВ должна обеспечивать возможность назначения различных механизмов аутентификации каждому пользователю в зависимости от его атрибутов.

3. *Без изменений.*

4. *Без изменений.*

Уровень **1А-5**. Комбинированное применение независимых механизмов идентификации и аутентификации.

1. *Без изменений.*

2. *Дополнение.* К каждому пользователю должны применяться два или более независимых механизма аутентификации. Аутентификация считается успешной, если все назначенные пользователю механизмы аутентификации подтвердили соответствие его идентификатора. ТСВ должна обеспечивать возможность назначения пользователю механизмов аутентификации в соответствии с атрибутам политики безопасности.

3. *Без изменений.*

4. *Без изменений.*

2. *Регистрация пользователя в системе*

Управление регистрацией пользователя в системе позволяет соблюсти требования политики аудита с помощью учета места, времени, способа и режима подключения пользователя к системе. Кроме того, управление регистрацией обеспечивает гарантию того, что зарегистрированный пользователь будет нести ответственность за выполняемые действия.

Процесс регистрации пользователя в системе должен управляться и контролироваться ТСВ. Должны быть четко определены условия, при которых возможно создание субъекта или субъектов, представляющих данного пользователя. Эти условия должны определяться на основании значений атрибутов пользователя, определяющих его статус, принадлежность к группе, возможные роли, степень полномочий, период разрешенного для работы времени, доступный ему сервис и ресурсы системы.

Требования к процедуре регистрации в системе ранжируются на основании обеспечиваемых возможностей защиты. Требования уровня **SE-1** включает в себя только простейшие возможности по управлению регистрацией в системе в соответствии с принадлежностью пользователя к определенной группе или выполнения им конкретной роли, а также степени его привилегированности. Этот уровень может использоваться в большинстве систем, поддерживающих выделенную самостоятельную процедуру регистрации пользователя. Системы, не реализующие от-

дельно такой процедуры в явном виде, используют для этой цели механизмы идентификации и аутентификации (по сути, идентификация и аутентификация могут считаться начальными процедурами управления регистрацией пользователя в системе). Уровень **SE-2** дополнен возможностями учета времени и места регистрации пользователя в системе. На уровне **SE-3** у пользователя появляются возможности блокировать (приостанавливать) и возобновлять сеанс работы с системой.

Уровень **SE-1**. Базовый механизм управления регистрацией пользователя в системе.

1. Перед началом выполнения процедуры регистрации пользователя в системе (процедуры *login)* ТСВ должна соответствующим сообщением предупреждать пользователя о недопустимости попытки неавторизованного проникновения в систему и о возможных последствиях подобных действий.

2. Перед началом сеанса работы (до разрешения регистрации) ТСВ должна выполнить процедуры идентификации и аутентификации пользователя. Если в ТСВ предусмотрена поддержка одновременного подключения нескольких независимых сеансов работы пользователя, то должен быть реализован механизм контроля количества одновременных подключений и не превышения их максимального числа.

3. ТСВ должна осуществлять регистрацию только в соответствии с подтвержденными аутентификацией атрибутами пользователя. Условия предоставления доступа к системе определяются на основании атрибутов пользователя в рамках политики безопасности. Если эти условия не определены явным образом, используются условия, принятые по умолчанию (например, успешная аутентификация пользователя).

4. ТСВ должна включать в себя защищенные механизмы, при помощи которых уполномоченный пользователь (администратор) может управлять атрибутами пользователей, используемыми при их регистрации в системе. Должны быть определены условия, при которых непривилегированный пользователь может ознакомиться с собственными атрибутами, но не изменить их.

5. После успешной регистрации пользователя в системе ТСВ должна предоставлять ему следующую информацию (без возможности ее модификации):

- место, время, режим, терминал или порт последней успешной регистрации пользователя;

- количество неуспешных попыток регистрации с использованием его идентификатора, зафиксированное между последним и текущим сеансом работы.

6. ТСВ должна обеспечивать блокирование (приостановку) или прерывание и завершение сеанса работы пользователя по истечению устанавливаемого администратором интервала отсутствия активности со стороны пользователя.

Уровень **SE-2**. Управление регистрацией в системе с учетом времени и места подключения.

1. *Без изменений.*

2. *Без изменений.*

3. *Дополнение.* ТСВ должна реализовывать механизм разрешения/запрещения регистрации в системе на основании контроля допустимого периода времени регистрации. Условия контроля периода времени должны быть определены для времени суток, дней недели и отдельных календарных дат.

ТСВ должна реализовывать механизм разрешения/запрещения регистрации в системе на основании контроля местоположения пользователя (используемых терминалов или портов). При необходимости должны быть также определены возможности блокировать (приостанавливать) и возобновлять сеанс работы с системой.

Уровень **SE-2**. Управление регистрацией в системе с учетом времени и места подключения.

1. *Без изменений*.

2. *Без изменений*.

3. *Дополнение.* ТСВ должна реализовывать механизм разрешения/запрещения регистрации в системе на основании контроля допустимого периода времени регистрации. Условия контроля периода времени должны быть определены для времени суток, дней недели и отдельных календарных дат.

ТСВ должна реализовывать механизм разрешения/запрещения регистрации в системе на основании контроля местоположения пользователя (используемых терминалов или портов). При необходимости должны быть также определены условия удаленного подключения пользователей по каналам связи.

4. *Без изменений.*

5. *Без изменений*

6. *Без изменений*

Уровень **SE-3**. Блокировка и восстановление сеанса работы пользователя.

1. *Без изменений*

2. *Без изменений*

3. *Без изменений*

4. *Без изменений*

5. *Без изменений*

6. *Дополнение.* ТСВ должна поддерживать механизм блокировки и возобновления сеанса работы пользователя по его команде или по истечению заданного администратором интервала отсутствия активности со стороны пользователя. Этот механизм должен обеспечивать:

- очистку экрана терминала для предотвращения возможности считывания с него информации и блокировку клавиатуры и других средств ввода информации;

- запрет на любые действия в системе с использованием заблокированных средств ввода-вывода информации на время приостановки сеанса пользователя;

- выполнение процедур идентификации и аутентификации пользователя перед возобновлением сеанса работы.

3. *Обеспечение прямого взаимодействия с ТСВ*

Функциональные требования, обеспечивающие прямое взаимодействие с ТСВ, ранжируются в зависимости от допустимой сферы применения и обеспечиваемых возможностей защиты. Минимальный уровень **ТР-1** предполагает наличие гарантированного канала взаимодействия пользователя с ТСВ, используемого для выполнения процедуры регистрации в системе. На уровне **ТР-2** прямое взаимодействие с ТСВ обеспечивается не только для процесса регистрации, но и для других процессов, требующих обеспечения гарантированно достоверного канала взаимодействия между пользователем, выполняющим действия, критичные с точки зрения безопасности (например, администрирование атрибутов безопасности), и ТСВ. На уровне **ТР-3** обеспечивается гарантированно достоверный канал взаимодействия

между пользователем и доверенными приложениями, позволяющими ему работать с критичной информацией.

Уровень **ТР-1.** Обеспечение прямого взаимодействия с ТСВ для процедуры регистрации в системе.

ТСВ должна обеспечивать гарантированно достоверный канал взаимодействия с пользователем в ходе процесса его идентификации и аутентификации во время регистрации в системе. Инициация прямого взаимодействия с ТСВ должна осуществляться только со стороны пользователя.

Уровень **ТР-2.** Обеспечение прямого взаимодействия пользователя с ТСВ.

Дополнение. ТСВ должна обеспечивать гарантированно достоверный канал для всех видов взаимодействий с пользователем (регистрация в системе, изменение атрибутов защиты и т.д.). Данный канал должен предусматривать возможность инициализации как со стороны пользователя, так и ТСВ, и должен быть изолирован от других аналогичных каналов, обладать уникальными характеристиками.

Уровень **ТР-3**. Обеспечение прямого взаимодействия пользователя с доверенными приложениями.

Дополнение. ТСВ должна обеспечивать гарантированно достоверный канал прямого взаимодействия пользователя с доверенными приложениями (например, для ввода или вывода критичной с точки зрения безопасности информации).

4. *Регистрация и учет событий в системе (аудит)*

Требования к регистрации и учету событий ранжируются в зависимости от предоставляемых возможностей отбора подлежащих регистрации событий, мощности средств анализа журнала событий и степени мониторинга действий пользователя. Требования к аудиту подразделяются на четыре группы:

- защита и управление доступом к системному журналу событий;
- определение множества подлежащих регистрации событий;
- фиксация и хранение зарегистрированных событий в журнале;
- анализ журнала событий и формирование отчетов.

Уровень **AD-1** включает минимальные требования к аудиту, которым должны следовать все системы в той мере, в какой они реализуют соответствующие функции защиты. На уровне **AD-2** эти требования усиливаются как за счет расширения множества типов регистрируемых событий, так и за счет добавления новых функций по управлению процессом регистрации и учета. На уровне **AD-3** появляются требования к наличию доверенных средств аудита, предоставляющих возможности выборочного анализа определенных типов событий, упрощающих взаимодействие с оператором за счет использования графического представления данных и т.п. Уровень **AD-4** характеризуется введением требования выявления критичных с точки зрения безопасности событий и объявления тревоги в случае их обнаружения. На уровне **AD**-5 требуется обеспечить такой же контроль в режиме реального времени (осуществлять в режиме реального времени обнаружение попыток нарушений безопасности).

Уровень **AD-1**. Минимальный аудит.

1. ТСВ должна обеспечивать возможность создания, хранения, ведения журнала аудита, содержащего регистрацию обращений к защищенным объектам. ТСВ должна обеспечивать защиту журнала от несанкционированного доступа, изменения или уничтожения. ТСВ должна предоставлять доступ к журналу только авторизованным пользователям.

2. ТСВ должна обеспечивать регистрацию в журнале аудита следующих типов событий:

- использование средств идентификации и аутентификации;
- создание и удаление объектов;
- доступ к объектам, помещение объектов в доступную пользователю область, запуск программ;
- действия, предпринятые операторами и администраторами, ответственными за безопасность.

Для поддержки в системе политики обеспечения работоспособности и контроля за распределением ресурсов должны регистрироваться попытки несанкционированных запросов на выделение ресурсов и попытки получения доступа к ресурсам, предоставленным другим субъектам.

При поддержке в системе нормативного управления доступом ТСВ должна иметь возможность осуществлять регистрацию

и учет изменений меток, классифицирующих уровень информации. Если нормативное управление доступом применяется для контроля потоков информации между субъектами, ТСВ должна иметь возможность регистрировать в журнале аудита события, которые потенциально могут использоваться для организации скрытых каналов передачи информации.

3. Для каждого регистрируемого события в журнал аудита заносятся:

- дата, время и тип события;
- идентификатор пользователя, инициировавшего событие;
- результат выполнения действия, соответствующего событию (успешное завершение или отказ).

При запросах на доступ к объектам или их удалении должны также регистрироваться имя и атрибуты объекта.

4. Администратор должен иметь возможность выбора регистрируемых событий и действий для каждого пользователя или объекта на основании соответствующих атрибутов политики безопасности.

Уровень **AD-2.** Базовые требования к аудиту.

1. *Без изменений.*

2. *Изменение.* ТСВ должна иметь возможность регистрировать следующие типы событий:

- использование механизмов идентификации, аутентификации и регистрации пользователя в системе;
- события, связанные с управлением доступом, относящиеся к определенному пользователю, субъекту, объекту или их атрибутам политики безопасности;
- создание, удаление субъектов и объектов, осуществление доступа, передача и отзыв прав доступа, изменение атрибутов политики безопасности, назначение и отзыв привилегий;
- действия, выполняемые операторами и администраторами, ответственными за безопасность, привилегированные операции, такие как модификация элементов ТСВ, настройка ТСВ, изменение параметров ТСВ и системных привилегий, изменение атрибутов пользователей, изменение состава и типов регистрируемых в журнале аудита событий.

Должен быть определен минимальный неизменяемый состав регистрируемых событий и их параметров. ТСВ должна содер-

жать средства защиты и управления множеством регистрируемых событий и их параметров и предоставлять доступ к ним только администратору.

3. *Без изменений.*

4. *Дополнение.* В ТСВ должны присутствовать защищенные средства запуска и остановки процесса аудита. Доступ к этим средствам, равно как и к средствам просмотра журнала аудита, должен быть разрешен только администратору.

ТСВ также должна включать средства управления аудитом, доступные только для администратора, которые позволяют осуществлять:

- создание и удаление журнала аудита, контроль и изменение его размеров;
- форматирование и упаковка записей журнала аудита;
- обеспечение целостности журнала аудита при сбоях и отказах системы.

Уровень **AD-3**. Развитые средства аудита.

1. *Без изменений.*

2. *Без изменений.*

3. *Без изменений.*

4. *Дополнение.* В ТСВ должны присутствовать специально разработанные средства контроля целостности журнала аудита, а также средства контроля целостности заданного множества регистрируемых событий.

1. Средства просмотра журнала аудита должны предоставлять авторизованному пользователю возможность ознакомления с данными аудита и их проверки. Данные средства должны быть защищены от несанкционированного доступа.

ТСВ также должна иметь средства обработки журнала аудита, позволяющие осуществлять выборочный анализ:

- действий одного или нескольких пользователей;
- действий над одним или несколькими объектами или ресурсами;
- всех или подмножества исключительных ситуаций;
- действий, ассоциированных с заданными атрибутами политики безопасности субъектов и объектов.

Средства просмотра журнала аудита должны предусматривать возможность работы параллельно со штатным функционированием системы.

Уровень **AD-4**. Обнаружение попыток нарушения безопасности.

1. *Без изменений.*

2. *Дополнение.* ТСВ должна содержать средства мониторинга событий, возникновение которых может означать угрозу нарушения безопасности. Эти средства должны незамедлительно оповещать администратора системы и останавливать (прекращать) выполнение вызвавшего это событие процесса или всей системы.

3. *Без изменений.*

4. *Без изменений.*

5. *Без изменений.*

Уровень **AD-5**. Выявление попыток нарушения безопасности в режиме реального времени.

1. *Без изменений.*

2. *Без изменений.*

3. *Без изменений.*

4. *Без изменений.*

5. *Дополнение.* ТСВ должна обеспечивать возможность регистрации событий и выявления попыток нарушения безопасности в режиме реального времени и оповещать о них администратора. Эта возможность должна реализовываться специальным механизмом мониторинга событий, критичных с точки зрения политики безопасности.

5. *Политика управления доступом (произвольное и нормативное управление доступом)*

Требования к реализации политики произвольного управления доступом могут быть ранжированы в зависимости от базы применения политики (ко всем субъектам и объектам системы, к выбранным подмножествам субъектов и объектов, в зависимости от атрибутов безопасности субъектов и объектов) и предоставляемых средств управления доступом (возможность управлять распространением прав доступа, возможность организации контроля доступа к объектам пользователей только с их разреше-

ния). Кроме того, эти требования можно ранжировать в зависимости от уровня абстракции, на котором рассматриваются субъекты (пользователь, группа, роль) и объекты (область памяти, файл, запись в файле).

Для ранжирования требований нормативного управления доступом могут использоваться те же самые критерии, однако описание уровня рассмотрения субъектов и объектов в этом случае должно производиться более точно. Поскольку нормативное управление доступом основано на контроле информационных потоков, должны контролироваться соответствующие атрибуты субъектов (например, состояние процесса) и объектов (размер, режим доступа и т.д.).

Требования к управлению доступом ранжированы по четырем уровням. На уровне **АС-1** устанавливаются минимальные требования к реализации политики управления доступом и допускается осуществление управлением доступа только по отношению к некоторому подмножеству субъектов и объектов, а также ограниченные возможности управления атрибутами безопасности. На уровне **АС-2** требования к политике безопасности расширяются в сторону возможности одновременного применения нескольких политик управления доступом и средств управления экспортом и импортом объектов. На уровне **АС-3** управление доступом должно поддерживаться для всех субъектов и объектов. Если реализована политика нормативного управления доступом, она должна использовать все атрибуты безопасности объектов и субъектов. На данном уровне также требуется проводить назначение прав доступа к объектам на основании их типа. На следующем уровне **АС-4** управление доступом расширяется путем добавления атрибутов времени и местоположения. Появляется возможность задания прав пользователей в зависимости от их принадлежности к определенной группе или выполнения ими определенной роли. В дополнение к требованиям контроля создания и уничтожения объектов вводится контроль за ресурсами и наследованием атрибутов. Предполагается, что этот уровень будет использоваться в системах, где требуется точно определенное управление доступом.

Уровень **АС-1**. Минимальное управление доступом.

1. Задание множества атрибутов безопасности объектов и субъектов. ТСВ должна задавать и поддерживать атрибуты безопасности субъектов и объектов. Атрибуты субъекта должны включать индивидуальный и групповой идентификаторы пользователя, представленного этим субъектом. Атрибуты объекта должны включать набор возможных прав доступа к этому объекту (чтение, запись, выполнение).

2. Управление атрибутами безопасности объектов и субъектов. ТСВ должна определять правила назначения и изменения атрибутов безопасности субъектов и объектов и обеспечивать их безусловное выполнение. Эти правила должны быть основаны на следующих положениях:

- субъект может разрешать доступ к объекту для другого субъекта, только в том случае, если он сам обладает этим правом доступа;
- пользователи должны иметь возможность устанавливать режим совместного использования объектов и управлять им на основе индивидуальных и групповых атрибутов субъектов;
- пользователи должны обладать средствами для контроля за процессом распространения и передачи прав доступа и иметь возможность ограничивать его.

Если для разных подмножеств субъектов и объектов определены различные правила управления атрибутами безопасности, то их реализация должна быть согласованной и не противоречить политике безопасности.

3. Управление доступом субъектов к объектам. ТСВ должна определять правила назначения полномочий (авторизацию) с целью управления доступом субъектов к объектам и обеспечивать их соблюдение. Эти правила должны быть основаны на атрибутах субъектов и объектов и обеспечивать защиту объектов от несанкционированного доступа.

Правила назначения полномочий должны охватывать четко определенное подмножество субъектов и объектов, а также принадлежащих им атрибутов безопасности. Должна быть обеспечена возможность применения различных правил назначения полномочий для различных групп субъектов и объектов, в этом случае реализация этих правил должна быть согласованной и не противоречить политике безопасности.

4. Контроль за созданием и уничтожением объектов и субъектов. ТСВ должна контролировать создание и уничтожение субъектов и объектов, а также повторное использование объектов. Все полномочия на доступ к объекту должны быть отозваны перед его уничтожением и предоставлением занимаемых им ресурсов в распоряжение системы. Вся содержащаяся в нем информация, в том числе и зашифрованная, должна быть уничтожена.

5. Инкапсуляция объектов. Если в ТСВ поддерживается механизм инкапсуляции объектов, он должен обеспечивать:

- авторизацию доступа к инкапсулированным объектам;
- возможность создания пользователем инкапсулированных объектов и подсистем;
- средства доступа к инкапсулированным объектам.

Уровень **АС-2.** Базовые механизмы управления доступом.

1. *Дополнение.* Если одновременно поддерживается несколько политик управления доступом, атрибуты безопасности субъектов и объектов для каждой политики должны быть определены отдельно. Атрибуты субъектов и объектов должны точно отражать уровень их конфиденциальности и целостности.

2. *Дополнение.* Правила управления атрибутами безопасности субъектов и объектов должны регламентировать назначение атрибутов в ходе импорта и экспорта объектов. В том числе, управлять импортом в систему неклассифицированной информации, не имеющей атрибутов безопасности, и экспортом из системы информации, обладающей атрибутами безопасности.

3. *Дополнение.* Если одновременно поддерживается несколько политик управления доступом, правила назначения полномочий доступа должны быть определены отдельно для каждой политики. ТСВ должна обеспечивать корректность совместного применения политик безопасности, в том числе, совместное применение правил авторизации каждой политики.

4. *Без изменений.*

5. *Без изменений.*

Уровень **АС-3**. Расширенное управление доступом.

1. *Дополнение.* ТСВ должна незамедлительно сообщать пользователю о любых изменениях атрибутов ассоциированных с ним субъектов, повлекших за собой изменение уровня привилегированности пользователя. Пользователю должна быть предос-

тавлена возможность запросить у ТСВ текущие значения атрибутов безопасности ассоциированных с ним субъектов.

ТСВ должна поддерживать назначение атрибутов безопасности всем подключенным к системе физическим устройствам (например, максимальные и минимальные уровни конфиденциальности). Эти атрибуты должны использоваться ТСВ для отражения особенностей функционирования данных устройств, обусловленных их физическими параметрами.

2. *Без изменений.*

3. *Изменение.* Правила назначения полномочий доступа должны быть определены для всех субъектов и объектов, которые прямо или косвенно доступны субъектам.

Если применяется нормативное управление доступом, то правила назначения полномочий доступа должны учитывать все атрибуты субъектов и объектов.

4. *Без изменений.*

5. *Без изменений.*

Уровень **АС-4**. Точно определенная политика управления доступом.

1. *Дополнение.* В состав атрибутов безопасности субъектов должны входить показатели времени и местоположения, позволяющие дополнительно аутентифицировать ассоциированного с данным субъектом пользователя.

2. *Дополнение.* Правила управления атрибутами безопасности субъектов и объектов должны обеспечивать задание для каждого объекта списка индивидуальных субъектов и групп субъектов с указанием их прав доступа к данному объекту. Кроме того, правила управления атрибутами безопасности субъектов и объектов должны обеспечивать задание для каждого объекта списка индивидуальных субъектов и групп субъектов, не имеющих прав доступа к данному объекту.

Эти правила также должны обеспечивать возможность управления атрибутами в зависимости от времени и места – предоставление или отзыв прав доступа могут быть осуществлены в определенный момент времени и продлиться заданный период, а также зависеть от местоположения объектов и субъектов.

3. *Дополнение.* Правила назначения полномочий доступа должны включать возможность использования атрибутов времени и места осуществления доступа.

4. *Изменение.* ТСВ должна определять и поддерживать правила контроля за созданием и уничтожением субъектов и объектов, позволяющие указать для каждого субъекта и объекта:

- полномочия, требуемые для их создания и уничтожения:
- процедуру повторного использования объекта;
- ресурсы, требуемые для их создания и размещения;
- устанавливаемые по умолчанию значения атрибутов созданных субъектов и объектов и, если требуется, правила наследования атрибутов.

Правила создания и уничтожения субъектов и объектов должны определяться на основании их типов. Если для различных типов субъектов и объектов определены разные правила их создания и уничтожения, то должно быть показано, что их совокупность реализует политику безопасности, принятую в системе. Если одновременно используется несколько политик безопасности, правила создания и уничтожения субъектов и объектов должны быть определены для каждой политики.

5. *Без изменений.*

6. *Контроль скрытых каналов*

Для осуществления контроля за скрытыми каналами требуется присутствие в ТСВ программных, аппаратных и специальных средств, позволяющих обнаруживать скрытые каналы и ограничивать возможности их использования путем их полной ликвидации или минимизации пропускной способности. Ранжирование данной группы требований проводится на основе типов контролируемых скрытых каналов и предоставляемых возможностей контроля (аудит, минимизация пропускной способности, ликвидация).

Скрытые каналы в зависимости от способа кодирования информации подразделяются на два типа: с использованием памяти и с использованием времени. В первом случае для кодирования передаваемой информации используется область памяти (например, установление характерных признаков в имени и атрибутах файла или зарезервированные поля в заголовке сетевого пакета).

Во втором случае, информация кодируется определенной последовательностью и длительностью событий, происходящих в системе (например, с помощью модуляции интервалов обращения к устройствам, введения задержек между приемом и посылкой сетевых пакетов и т.д.).

Уровень **CCH-1** затрагивает только скрытые каналы, использующие память, и ограничивается контролем их использования. На уровне **CCH-2** добавляются требования минимизации пропускной способности и исключения возможностей использования скрытых каналов для штатного режима эксплуатации системы. Уровень **CCH-3** требует полного подавления всех типов скрытых каналов.

Уровень **CCH-1.** Контроль скрытых каналов, использующих память.

1. ТСВ и привилегированные приложения должны содержать функции контроля использования скрытых каналов, использующих память. Эти функции должны позволять идентифицировать источник и приемник скрытого обмена информацией и способ использования скрытого канала.

2. Функции ТСВ и привилегированных приложений, осуществляющие контроль скрытых каналов, должны быть определены для каждого скрытого канала и присутствовать в типовой конфигурации системы. Если для некоторых скрытых каналов функции контроля отсутствуют, должно быть проведено доказательство невозможности их использования для нарушения безопасности.

Уровень **CCH-2.** Контроль и ограничение пропускной способности скрытых каналов, использующих память.

1. *Дополнение.* Должно обеспечиваться ограничение пропускной способности или полное подавление скрытых каналов, использующих память. Пропускная способность каждого скрытого канала должна контролироваться администратором.

2. *Дополнение.* Функции ТСВ и привилегированных положений, обеспечивающие ограничение пропускной способности или полное подавление скрытых каналов, должны присутствовать в типовой конфигурации системы. Если для некоторых скрытых каналов функции ограничения пропускной способности или полного подавления отсутствуют, должно быть проведено

доказательство невозможности их использования для нарушения безопасности.

Уровень **CCH-3.** Контроль и ограничение пропускной способности скрытых каналов, использующих время.

1. *Без изменений.*

2. *Дополнение.* Функции контроля, ограничения пропускной способности и подавления скрытых каналов должны в полной мере распространяться и на скрытые каналы, использующие время.

7. *Контроль за распределением ресурсов*

Данные требования являются частью политики обеспечения работоспособности системы и позволяют контролировать использование ресурсов системы. Ранжирование проводится по отношению к множеству управляемых ресурсов (т.е. подмножества ресурсов с ограниченным распределением) и функциональным возможностям средств управления.

Уровень **AR-1** определяет базовые требования к контролю за предоставлением ресурсов в терминах ограниченного подмножества системных ресурсов, субъектов и объектов. Уровень **AR-2** расширяет область применения средств контроля за предоставлением ресурсов до множества всех системных ресурсов с одновременным введением контроля за попытками монопольного захвата ресурсов и их доступностью для всех субъектов. На уровне **AR-3** к этим требованиям добавляется управление распределением ресурсов на основании приоритета субъекта и введение фиксированных квантов, которыми распределяются ресурсы.

Уровень **AR-1.** Ограничение при распределении ресурсов.

ТСВ должна обеспечивать возможность ограничения множества субъектов и объектов, доступных пользователю одновременно. ТСВ должна контролировать распределение определенного подмножества системных ресурсов таким образом, чтобы ни один пользователь не мог нарушить работу другого пользователя путем захвата такого количества ресурсов системы, при котором другие пользователи не могут осуществлять доступ к объектам и субъектам. Для всех субъектов, объектов и ресурсов должны быть определены ограничения на время и количество использования, а для ресурсов – атрибуты, обозначающие их количество.

Уровень **AR-2.** Полный контроль за распределением ресурсов.

Дополнение. ТСВ должна контролировать распределение системных ресурсов таким образом, чтобы ни один пользователь не мог сделать любой системный ресурс недоступным для других пользователей или ограничить возможности ТСВ по обслуживанию других пользователей, путем захвата ресурсов или осуществления манипуляций с ТСВ.

Уровень **AR-3.** Распределение ресурсов на основании приоритетов.

Дополнение. ТСВ должна обеспечивать возможность распределения ресурсов на основании специально выделенных атрибутов, поставленных в соответствие каждому субъекту. ТСВ должна осуществлять распределение ресурсов в первую очередь субъектам, обладающим более высоким приоритетом. Все ресурсы должны выделяться блоками определенного размера (квантами).

8. *Политика управления безопасностью*

Ранжирование требований к средствам управления безопасностью основано на множестве управляемых параметров и уровне предоставляемых возможностей. Уровень **8М-1** содержит минимальные требования по управлению безопасностью. Уровень 5М-2 является базовым и предназначен для применения в большинстве систем. На уровне **8М-3** надежность механизма управления обеспечивается за счет разделения ролей администратора и оператора системы и применения более широкого набора средств управления безопасностью. На уровне **SM-4** требуется наличие доверенных средств управления безопасностью и введение контроля за администрированием системы.

Уровень **SM-1.** Минимальное управление безопасностью.

1. ТСВ должна содержать доверенные средства установки и настройки собственных конфигурационных параметров и инициализации критических внутренних структур данных перед заданием атрибутов безопасности пользователей и администраторов.

2. ТСВ должна поддерживать доверенные средства просмотра и редактирования параметров политики безопасности.

3. ТСВ должна включать доверенные средства просмотра, редактирования и удаления параметров регистрации пользователей и их бюджетов. Эти параметры должны включать уникальный идентификатор пользователя, его имя и служебное положение. Данные средства должны позволять администратору приостанавливать и возобновлять действие идентификаторов пользователей и их бюджетов.

4. ТСВ должна содержать доверенные средства контроля функционирования системы и состояния системных ресурсов. Эти средства должны обеспечивать подключение и отключение внешних устройств, съемных носителей информации, резервное копирование и восстановление объектов, эксплуатацию и тестирование программных и аппаратных компонентов ТСВ, запуск и остановку системы.

5. Средства управления безопасностью должны быть доступны только для администратора системы.

Уровень **SM-2**. Базовые механизмы управления безопасностью.

1. *Дополнение.* Средства управления безопасностью должны учитывать различие между режимами штатного функционирования и технического обслуживания системы и поддерживать управление безопасностью и в том и в другом режиме. Режим технического обслуживания системы должен позволять проводить восстановление после сбоев и запуск системы.

2. *Дополнение.* В состав параметров политики безопасности должны входить параметры идентификации, аутентификации, регистрации в системе и параметры управления доступом как для системы в целом, так и для каждого отдельного пользователя.

ТСВ должна позволять администратору определять политику идентификации и аутентификации для всех пользователей системы (период смены паролей, их длину и сложность). Средства управления параметрами политики безопасности должны позволять ограничивать:

- максимальный период отсутствия активности со стороны пользователя;
- максимальное время работы пользователя в системе;
- максимальное число последовательно осуществленных безуспешных попыток регистрации в системе.

Если в системе обеспечивается поддержка политики обеспечения работоспособности, ТСВ должна поддерживать механизм управления доступностью системных ресурсов посредством введения квот и ограничений на объем потребляемых ресурсов.

3. *Дополнение.* ТСВ должна содержать средства для однозначной идентификации каждого параметра политики безопасности. Кроме того, должна быть предусмотрена возможность получения списка атрибутов безопасности для каждого пользователя и списка пользователей, ассоциированных с каждым атрибутом безопасности.

Должна обеспечиваться возможность управления атрибутами политики безопасности субъектов, включая привилегии, атрибуты произвольного и нормативного управления доступом, а также централизованного контроля, назначения и снятия атрибутов политики безопасности.

4. *Без изменений.*

5. *Без изменений.*

Уровень **SM-3**. Управление безопасностью в соответствии с политикой безопасности.

1. *Дополнение.* Режим технического обслуживания системы должен включать средства инициализации параметров идентификации, аутентификации, регистрации в системе и назначения полномочий администратора системы.

2. *Дополнение.* В случае совместного использования нескольких методов аутентификации ТСВ должна предоставлять администратору возможность определять методы аутентификации пользователей в зависимости от соответствующих атрибутов политики безопасности.

Если ТСВ поддерживает одновременно несколько сеансов для одного пользователя, администратор должен иметь возможность ограничить число одновременных регистраций для каждого пользователя в зависимости от его атрибутов безопасности.

ТСВ должна позволять администратору ограничивать регистрацию для пользователя с определенным идентификатором или с определенного терминала после заданного количества неуспешных попыток регистрации с помощью этого идентификатора или терминала.

3. *Дополнение.* ТСВ должна автоматически приостанавливать полномочия пользователей в случае, если они не использовались в течение заданного администратором периода времени. ТСВ также должна обеспечивать автоматическое возобновление приостановленных полномочий по истечении указанного администратором времени.

4. *Дополнение.* ТСВ должна поддерживать разделение функций оператора и администратора. Функции оператора должны ограничиваться управлением внешними устройствами.

5. *Без изменений.*

Уровень **SM-4**. Расширенное управление безопасностью.

1. *Без изменений.*

2. Без изменений.

3. *Дополнение.* ТСВ должна содержать доверенные средства администрирования системы, осуществляющие контроль:

- конфигурации системы и регистрации пользователей;
- корректности инсталляции системы;
- отсутствия в ТСВ посторонних программ и данных.

ТСВ должна включать средства контроля безопасности начального состояния ТСВ после инициализации или восстановления.

ТСВ должна включать средства контроля соответствия между пользователями, субъектами, представляющими их в системе, и назначенными им атрибутами безопасности.

4. *Дополнение.* Средства контроля функционирования системы должны поддерживать разделение ролей администратора безопасности и аудитора, контролирующего администрирование. ТСВ должна выполнять заданные администратором действия только после их регистрации в журнале аудита. Не влияющие на безопасность системы действия администратора должны быть строго ограничены для обеспечения эффективного управления безопасностью.

5. *Без изменений.*

9. ***Мониторинг взаимодействий***

Ранжирование требований к мониторингу взаимодействий производится по отношению к области применения мониторинга и степени детализации взаимодействий. На уровне **RM-1** мони-

торинг взаимодействий ограничивается только заданными подмножествами субъектов и объектов, обращения к которым контролируются политикой управления доступом. На уровне **RM-2** мониторинг взаимодействий должен применяться для всех субъектов и объектов. Уровень **RM-3** увеличивает степень детализации с помощью мониторинга атрибутов безопасности и статуса объектов, субъектов и ресурсов. Уровень **RM-4**, предназначенный для использования в системах, где действуют привилегированные процессы, предусматривает поддержку модели мониторинга взаимодействий привилегированных процессов.

Уровень **RM-1.** Мониторинг взаимодействий для заданных подмножеств субъектов и объектов.

1. ТСВ должна осуществлять мониторинг всех взаимодействий, в которых участвуют субъекты, объекты и ресурсы, включенные в спецификацию ТСВ. Мониторинг должен обеспечивать контроль взаимодействий в соответствии с политикой безопасности.

2. Мониторинг взаимодействий должен осуществляться для заданного подмножества субъектов, объектов и ресурсов, находящихся под контролем политики безопасности системы, а также для обращений к их атрибутам безопасности (правам доступа, уровням конфиденциальности, ролям, квотам и т.д.).

3. Мониторинг взаимодействий привилегированных субъектов должен осуществляться в соответствии с атрибутами безопасности этих субъектов.

Уровень **RM-2.** Мониторинг взаимодействий для всех субъектов и объектов.

1. *Без изменений.*

2. *Дополнение.* Мониторинг взаимодействий должен осуществляться для всех объектов, субъектов и ресурсов, и их атрибутов безопасности.

3. *Без изменений.*

Уровень **RM-3.** Мониторинг взаимодействий и контроль атрибутов безопасности.

1. *Без изменений.*

2. *Дополнение.* Требования мониторинга взаимодействий обращений к атрибутам субъектов, объектов и ресурсов распро-

страняются на полное множество атрибутов (состояние, размер, режим использования).

3. *Без изменений.*

Уровень **RM-4.** Мониторинг взаимодействий привилегированных субъектов.

1. *Без изменений.*

2. *Без изменений.*

3. *Дополнение.* Мониторинг взаимодействий привилегированных субъектов должен осуществляться на основе модели безопасности и мониторинга, определенной для этих субъектов.

10. *Логическая защита ТСВ*

Ранжирование требований логической защиты ТСВ проводится на основе их возможностей по обеспечению безопасности ТСВ. Уровень **LP-1** содержит основные требования к изоляции ТСВ. На уровне **LP-2** эти требования расширяются за счет введения средств контроля целостности структур данных ТСВ и исключения влияния на состояние ТСВ со стороны непривилегированных пользователей. Эти требования призваны сделать невозможным применение злоумышленником средств проникновения в ТСВ. На уровне **LP-3** вводится требование синхронности контроля целостности ТСВ.

Уровень **LP-1.** Базовые средства изоляции ТСВ.

ТСВ должна функционировать внутри собственного домена, изолированного от остальных компонентов системы. Изоляция домена ТСВ должна обеспечивать защиту от внешних воздействий, модификации программ или данных ТСВ.

1. Изоляция компонентов ТСВ должна включать:

- изоляцию адресного пространства ТСВ от адресного пространства непривилегированных субъектов таким образом, чтобы они не могли получить доступ по чтению и записи к программам и данным ТСВ;

- взаимодействие между доменом ТСВ и остальными компонентами системы должно осуществляться таким образом, чтобы неконтролируемый обмен информацией с ТСВ был невозможен;

- параметры, передаваемые в ТСВ, должны контролироваться на допустимость их значений или принадлежность к адресному пространству ТСВ.

2. Для обеспечение надежности изоляции ТСВ права доступа к объектам, переданным в ТСВ в качестве параметров, должны проверяться на соответствие требуемым, а обращения к объектам ТСВ со стороны средств, обеспечивающих изоляцию, контролироваться монитором пересылок.

Уровень **LP-2.** Изоляция и контроль целостности ТСВ.

1. *Без изменений.*

2. *Дополнение.* Средства защиты ТСВ также должны осуществлять контроль целостности структур данных ТСВ и предотвращать влияние на них со стороны непривилегированных пользователей.

3. Контроль целостности структур данных ТСВ с помощью вычисления функции-инварианта, определенной на множестве переменных, объектов и функций, должен осуществляться до и после любого обращения к ТСВ.

4. Для предотвращения влияния на ТСВ со стороны непривилегированных пользователей необходимо обеспечить, чтобы любое обращение пользователя к ТСВ не приводило к нарушениям в обработке запросов остальных пользователей.

Уровень **LP-3.** Изоляция и синхронный контроль целостности ТСВ.

1. *Без изменений.*

2. Без изменений.

3. *Без изменений.*

4. *Дополнение.* Защита ТСВ должна обеспечивать синхронность функций контроля целостности.

5. Синхронность функций контроля целостности означает, что действия, основанные на результатах проверки целостности, осуществляются непосредственно после завершения процесса проверки, и между этими событиями состояние ТСВ измениться не может.

11. *Физическая защита ТСВ*

Ранжирование требований физической защиты ТСВ производится на основе обеспечиваемого уровня защиты, то есть воз-

можности предвидеть, обнаруживать и предотвращать атаки на систему на физическом уровне.

На уровне **PP-1** от средств обеспечения физической защиты требуется применение административных мер и контроля среды функционирования. На уровне **PP-2** выдвигается требование к устройствам распознавать попытки физического вмешательства в их работу. Уровень **PP-3** требует наличия средств противодействия атакам на конфиденциальность и целостность системы, а также изменениям в среде функционирования.

Уровень **PP-1.** Административные меры и контроль среды функционирования.

1. Должны быть определены административные меры и параметры физического контроля среды функционирования, необходимые для обеспечения защиты ТСВ.

2. Должны иметься и надлежащим образом применяться средства и устройства, необходимые для осуществления физического контроля за компонентами ТСВ.

Уровень **PP-2.** Обнаружение атак на физическом уровне.

1. *Без изменений.*

2. Дополнение. Средства и устройства, осуществляющие физический контроль, должны обеспечивать однозначное обнаружение физического воздействия на ТСВ. Эти устройства должны быть надежны и устойчивы к непосредственному физическому воздействию.

Уровень **PP-3.** Противодействие атакам на физическом уровне и неблагоприятным изменениям в среде функционирования.

1. *Без изменений.*

2. Дополнение. Должны иметься средства противодействия атакам на физическом уровне, их характеристики должны соответствовать требованиям политики безопасности. Для обеспечения конфиденциальности эти устройства должны противостоять попыткам кражи и исследования компонентов ТСВ с помощью физического воздействия, подслушивания, перехвата и анализа излучений. Для обеспечения целостности эти устройства должны противодействовать несанкционированному изменению состава аппаратного обеспечения, нарушению его функционирования, а также воздействиям на хранящуюся в системе информацию ме-

ханическими или электромагнитными методами. Для обеспечения работоспособности системы эти устройства должны противодействовать возникновению ситуаций, затрудняющих обслуживание пользователей (вибрации, вода, огонь и другие формы физических воздействий).

12. ***Самоконтроль ТСВ***

Требования самоконтроля ТСВ ранжируются на основе перечня контролируемых элементов (аппаратное, программное и специальное обеспечение) и возможностей механизмов проверки (периодичность, длительность, глубина проверки).

На уровне **SC-1** представлены минимальные требования к самоконтролю ТСВ, предполагается, что их выполнения будет достаточно для большинства коммерческих приложений. Уровень **SC-2** содержит расширенные требования, включающие в себя тестирование при включении питания, загрузке системы, а также управляемые оператором средства тестирования, применяемые для периодического контроля функционирования элементов ТСВ. На уровне **SC-3** появляются требования к тестированию программных компонентов ТСВ. На уровне **SC-4** требуется, чтобы тестирование осуществлялось регулярно на протяжении всего периода функционирования системы.

Уровень **SC-1**. Минимальный самоконтроль.

ТСВ должна включать аппаратные и/или программные средства периодического контроля целостности и корректности функционирования собственных аппаратных и специальных компонентов.

Уровень **SC-2**. Базовые механизмы самоконтроля.

Дополнение. В состав средств контроля должны входить: тестирование при включении питания, тестирование в ходе загрузки системы и средства тестирования, управляемые оператором.

Тесты при включении питания должны осуществлять проверку всех аппаратных и специальных компонентов ТСВ, включая оперативную память, шины, соединения, разъемы, управляющие контроллеры, процессор, адаптеры дисковых накопителей и других устройств, коммуникационные порты, консоль и клавиатуру. Эти тесты также должны осуществлять проверку

всех компонентов, используемых для выполнения тестов при загрузке системы, и тестов, управляемых оператором.

Тесты при загрузке системы должны включать в себя проверку компонентов центрального процессора (арифметические и логические устройства, математический сопроцессор, устройство декодирования инструкций, контроллер прерываний, кэш, буфер трансляции адресов, внутренние и внешние шины), а также системной шины, контроллеров оперативной памяти и устройств, используемых в ходе тестов, управляемых оператором, и при удаленном тестировании системы.

Выполняемые оператором тесты должны обеспечивать однократное или многократное тестирование компонентов ТСВ и системы в целом, регистрацию результатов тестирования, и, в случае выявления неисправности, выполнять специальные процедуры локализации неисправностей и уведомлять оператора.

Уровень **SC-3**. Тестирование программных средств.

Дополнение. Должны иметься управляемые и конфигурируемые программные или специальные средства тестирования целостности и корректности программных компонентов ТСВ – программ, данных и носителей информации. Эти средства должны включать проверку контрольных сумм и другие механизмы контроля.

Уровень **SC-4**. Регулярное тестирование программных средств.

Дополнение. Тесты контроля целостности и корректности функционирования программных компонентов ТСВ должны выполняться при каждом изменении содержания или структуры этих компонентов, возникающих при сбоях и отказах, произошедших из-за действий непривилегированных субъектов.

13. *Инициализация и восстановление ТСВ*

Требования инициализации и восстановления безопасного состояния ТСВ ранжируются по отношению к уровню предоставляемых возможностей: ручное восстановление (уровни **TR-1** и **TR-2),** автоматическое (уровень **TR-3),** обнаружение объектов пользователей (уровень **TR-4**), минимизация потерь объектов (уровень **TR-5**).

Уровень **TR-1**. Минимальные требования восстановления и инициализации.

1. ТСВ должна содержать механизмы, обеспечивающие гарантированное восстановление безопасного состояния ТСВ после сбоев без нарушения функций защиты.

Уровень **TR-2**. Базовые средства восстановления и инициализации ТСВ.

1. *Без изменений.*

2. В случае невозможности автоматического восстановления и реинициализации безопасного состояния, ТСВ должна переходить в особое состояние, в котором доступ может осуществляться только с помощью специальных административных процедур, с помощью которых можно осуществить восстановление ТСВ вручную, без нарушений функций защиты.

Уровень **TR-3**. Автоматическое восстановление и инициализация ТСВ.

1. *Без изменений.*

2. *Изменение.* ТСВ должна включать средства автоматического восстановления безопасного состояния ТСВ после ошибок и сбоев. Эти средства должны по возможности исключать потерю системных и пользовательских объектов. Должны быть определены требования, или правила политики безопасности, позволяющие подтвердить безопасность ТСВ после восстановления.

Уровень **TR-4**. Обнаружение потерь объектов.

1. *Без изменений.*

2. *Дополнение.* ТСВ должна включать специальную контрольную функцию восстановления, способную обнаруживать повреждение или разрушение объектов в результате сбоев и предупреждать об этом пользователей.

Уровень **TR-5**. Минимизация потерь объектов.

1. *Без изменений.*

2. *Дополнение.* Все вызываемые извне функции и операции ТСВ должны быть атомарными, то есть, либо завершаться полным выполнением указанных действий, либо, при возникновении сбоев, сохранять исходное состояние используемых объектов, субъектов и ресурсов. Применение атомарных функций позволяет минимизировать искажения и потери объектов при сбоях системы.

14. ***Ограничение привилегий при работе с ТСВ***

Требования ограничения привилегий при работе с ТСВ ранжируются на основе детализации описания привилегий, ассоциированных с отдельными функциями или группами функций ТСВ (уровень **РО-1**), с отдельными компонентами (модулями) ТСВ и административными ролями (**РО-2**), с отдельными операциями (**РО-3**) и динамически изменяющихся в ходе выполнения операций (**РО-4**).

Уровень **РО-1**. Назначение привилегий для выполнения функций ТСВ.

1. Должны быть определены привилегии, необходимые для выполнения отдельных функций ТСВ или их групп. Также должны быть определены привилегированные функции и объекты ТСВ, такие как файлы регистрации пользователей, файлы паролей, файлы, содержащие уровни безопасности пользователей и объектов, списки ролей пользователей, файлы журнала аудита.

2. Должен быть назначен минимальный уровень привилегий, необходимый и достаточный для осуществления доступа к установленным привилегированным функциям и объектам ТСВ.

Уровень **РО-2.** Назначение привилегий доступа к компонентам ТСВ.

1. *Дополнение.* Должна обеспечиваться возможность назначения необходимых привилегий действиям, выполняемым в ТСВ привилегированными пользователями (администраторами).

2. *Изменение.* Всем функциям и компонентам (модулям) ТСВ должен быть поставлен в соответствие минимальный уровень привилегий, необходимый и достаточный для их доступа к ним.

3. Должна быть обеспечена поддержка реализации привилегий модулей ТСВ с помощью низкоуровневых процедур и механизмов.

Уровень **РО-3.** Назначение привилегий для выполнения отдельных операций.

1. *Без изменений.*

2. *Дополнение.* Должны быть определены привилегии, необходимые для выполнения отдельных операций ТСВ. Для каждой операции должен быть назначен минимальный уровень привилегий, необходимый и достаточный для ее выполнения.

3. *Изменение.* Должна быть обеспечена поддержка реализации привилегий для отдельных операций ТСВ с помощью низкоуровневых процедур и механизмов.

Уровень **РО-4.** Динамическое назначение привилегий для выполнения отдельных операций.

1. *Без изменений.*

2. *Дополнение.* Установленные привилегии должны использоваться всеми функциональными компонентами ТСВ для контроля и ограничения распространения ошибок в работе механизмов защиты и предоставления полномочий, которые могут повлечь за собой нарушение политики безопасности. Должны быть определены функции ТСВ, позволяющие при необходимости динамически повышать привилегии отдельных операций ТСВ (но не выше заданной границы) и автоматически их понижать по завершению выполнения соответствующих операций. Эти меры должны ограничивать использование высокопривилегированных операций ТСВ, потенциально предоставляющих пользователю возможности использования этих привилегий для нарушения политики безопасности.

3. *Без изменений.*

15. *Простота использования ТСВ*

Ранжирование требований данной группы отражает имеющиеся возможности управления конфигурацией ТСВ. На уровне **EU-1** сформулированы общие требования, отражающие необходимость наличия развитых средств управления безопасностью системы, вместо использования обычных редакторов для модификации параметров безопасности или содержимого файлов регистрации пользователей. Требования к функциональности средств администрирования расширяются на уровне **EU-2** посредством введения возможности установки атрибутов безопасности по умолчанию для некоторых субъектов и объектов и наличия средств, позволяющих приложениям обеспечить собственную защиту и защиту своих объектов от несанкционированного использования. На уровнях **EU-3 и EU-4** требования усиливаются и расширяются за счет увеличения множеств субъектов и объектов, на которые они распространяются для стандартной и полной конфигурации системы.

Уровень **EU-1**. Простота управления безопасностью.

1. ТСВ должна обеспечивать поддержку функций администрирования. Должна быть предусмотрена возможность задания значений параметров безопасности по умолчанию для средств администрирования.

Уровень **EU-2**. Простота разработки приложений.

1. *Дополнение.* ТСВ должна обеспечивать автоматическую установку атрибутов безопасности по умолчанию для определенных субъектов и объектов и возможность модификации этих атрибутов.

2. ТСВ должна предусматривать четко определенный программный интерфейс взаимодействия со всеми принятыми в системе политиками безопасности для поддержки приложений, которые могут обеспечить поддержку этих политик безопасности на прикладном уровне. ТСВ должна предоставлять пользователю возможность понижения полномочий используемых приложений.

Уровень **EU-3.** Простота использования стандартной конфигурации системы.

1. *Изменение.* ТСВ должна обеспечивать автоматическую установку атрибутов безопасности по умолчанию для определенных субъектов, объектов и служб, присутствующих в стандартной конфигурации системы, а также возможность модификации этих атрибутов.

2. *Без изменений.*

Уровень **EU-4.** Простота использования полной конфигурации системы.

1. *Изменение.* ТСВ должна обеспечивать автоматическую установку атрибутов безопасности по умолчанию для всех субъектов, объектов и служб системы, а также возможность модификации этих атрибутов.

2. *Без изменений.*

ПРИЛОЖЕНИЕ 2

«Оранжевая книга» США

Наиболее значимыми стандартами информационной безопасности являются: «Критерии безопасности компьютерных систем министерства обороны США», «Европейские критерии безопасности информационных технологий», руководящие документы Гостехкомиссии России, «Канадские критерии безопасности компьютерных систем» и др.

С 1983 по 1988 год Министерство обороны США и Национальный комитет компьютерной безопасности разработали систему стандартов в области компьютерной безопасности, которая включает более десяти документов. Этот список возглавляют «Критерии оценки безопасности компьютерных систем», которые по цвету обложки чаще называют «Оранжевой книгой». В 1995 году Национальный центр компьютерной безопасности США опубликовал «Пояснения к критериям безопасности компьютерных систем», объединившие все имеющиеся на тот момент дополнения и разъяснения к «Оранжевой книге».

В «Оранжевой книге» надежная система определяется как «система, использующая достаточные аппаратные и программные средства, чтобы обеспечить одновременную обработку информации разной степени секретности группой пользователей без нарушения прав доступа».

Надежность систем оценивается по двум основным критериям:

Политика безопасности – набор законов, правил и норм поведения, определяющих, как организация обрабатывает, защищает и распространяет информацию. В частности, правила определяют, в каких случаях пользователь имеет право оперировать с определенными наборами данных. Чем надежнее система, тем строже и многообразнее должна быть политика безопасности. В зависимости от сформулированной политики можно выбирать конкретные механизмы, обеспечивающие безопасность системы. Политика безопасности – это активный компонент защиты, включающий в себя анализ возможных угроз и выбор мер противодействия.

Гарантированность – мера доверия, которая может быть оказана архитектуре и реализации системы. Гарантированность можно определить тестированием системы в целом и ее компонентов. Гарантированность показывает, насколько корректны механизмы, отвечающие за проведение в жизнь политики безопасности. Гарантированность можно считать пассивным компонентом защиты, надзирающим за самими защитниками.

Важным средством обеспечения безопасности является механизм подотчетности (протоколирования). Надежная система должна фиксировать все события, касающиеся безопасности. Ведение протоколов должно дополняться аудитом, то есть анализом регистрационной информации.

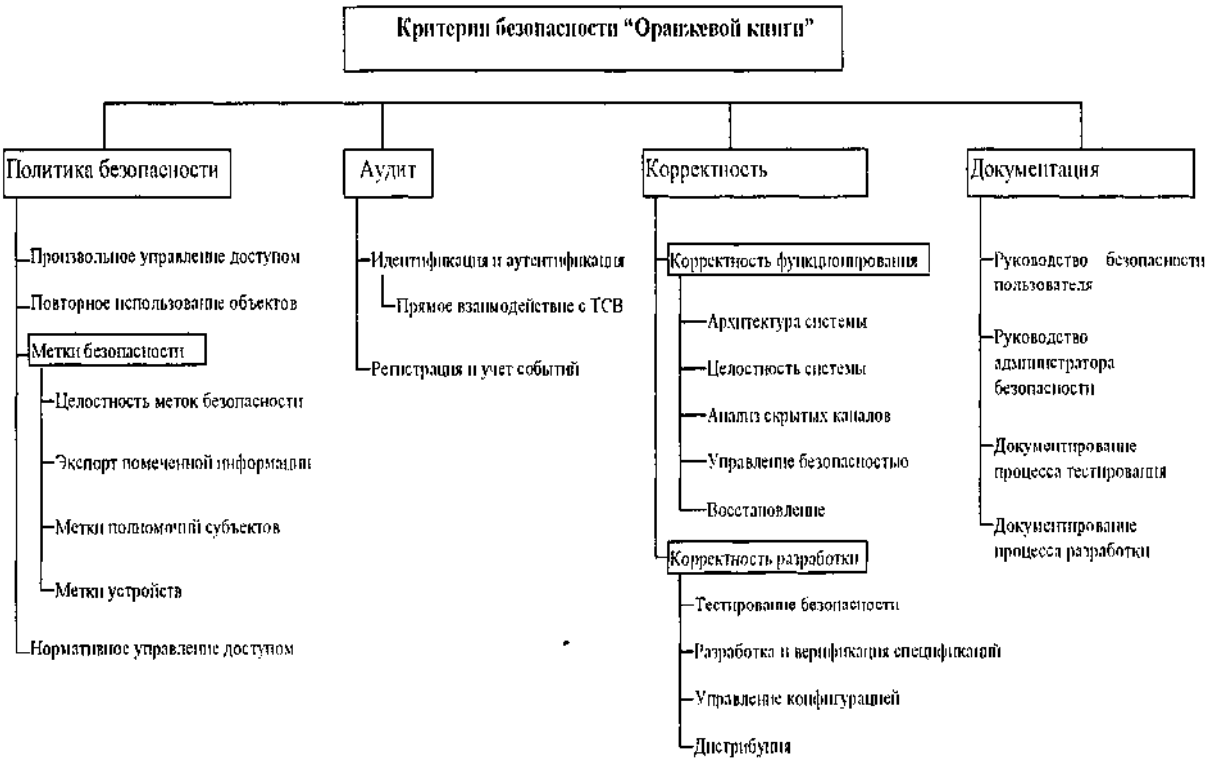


Рис. 2.1 – Структура «Оранжевой книги»

При оценке степени гарантированности, с которой систему можно считать надежной, центральной является концепция надежной вычислительной базы. Вычислительная база – это совокупность защитных механизмов компьютерной системы (включая аппаратное и программное обеспечение), отвечающих за проведение в жизнь политики безопасности. Надежность вычислительной базы определяется исключительно ее реализацией и кор-

ректностью исходных данных, которые вводит административный персонал (например, это могут быть данные о степени благонадежности пользователей).

Основное назначение надежной вычислительной базы – выполнять функции монитора обращений, то есть контролировать допустимость выполнения субъектами определенных операций над объектами. Каждое обращение пользователя к программам или данным проверяется на предмет согласованности со списком действий, допустимых для пользователя.

От монитора обращений требуется выполнение трех свойств:

Изолированность. Монитор должен быть защищен от отслеживания своей работы.

Полнота. Монитор должен вызываться при каждом обращении, не должно быть способов его обхода.

Верифицируемость. Монитор должен быть компактным, чтобы его можно было проанализировать и протестировать, будучи уверенным в полноте тестирования.

Основные элементы политики безопасности. Согласно «Оранжевой книге», политика безопасности должна включать в себя по крайней мере следующие элементы:

- произвольное управление доступом;
- безопасность повторного использования объектов;
- метки безопасности;
- принудительное управление доступом.

Рассмотрим перечисленные элементы подробнее.

Произвольное управление доступом. Произвольное управление доступом – это метод ограничения доступа к объектам, основанный на учете личности субъекта или группы, в которую субъект входит.

Произвольность управления состоит в том, что некоторое лицо (обычно владелец объекта) может по своему усмотрению давать другим субъектам или отбирать у них права доступа к объекту. Текущее состояние прав доступа при произвольном управлении описывается матрицей, в строках которой перечислены субъекты, а в столбцах – объекты. В клетках, расположенных на пересечении строк и столбцов, записываются способы досту-

па, допустимые для субъекта по отношению к объекту, например: чтение, запись, выполнение, возможность передачи прав другим субъектам и т.п.

Очевидно, прямолинейное представление подобной матрицы невозможно (поскольку она очень велика), да и не нужно (поскольку она разрежена, то есть большинство клеток в ней пусты). В операционных системах более компактное представление матрицы доступа основывается или на структурировании совокупности субъектов (владелец/группа/прочие в ОС UNIX), или на механизме списков управления доступом, то есть на представлении матрицы по столбцам, когда для каждого объекта перечисляются субъекты вместе с их правами доступа. За счет использования метасимволов можно компактно описывать группы субъектов, удерживая тем самым размеры списков управления доступом в разумных рамках.

Большинство операционных систем и систем управления базами данных реализуют именно произвольное управление доступом. Главное его достоинство – гибкость, главные недостатки – рассредоточенность управления и сложность централизованного контроля, а также оторванность прав доступа от данных, что позволяет копировать секретную информацию в общедоступные файлы.

Безопасность повторного использования объектов. Безопасность повторного использования объектов – важное на практике дополнение средств управления доступом, предохраняющее от случайного или преднамеренного извлечения секретной информации из «мусора». Безопасность повторного использования должна гарантироваться для областей оперативной памяти, в частности для буферов с образами экрана, расшифрованными паролями и т.п., для дисковых блоков и магнитных носителей в целом.

Важно обратить внимание на следующий момент. Поскольку информация о субъектах также представляет собой объект, необходимо позаботиться о безопасности «повторного использования субъектов». Когда пользователь покидает организацию, следует не только лишить его возможности входа в систему, но и запретить доступ ко всем объектам. В противном случае новый сотрудник может получить ранее использовавшийся идентифика-

тор, а с ним и все права своего предшественника. Современные интеллектуальные периферийные устройства усложняют обеспечение безопасности повторного использования объектов. Действительно, принтер может буферизовать несколько страниц документа, которые останутся в памяти даже после окончания печати. Необходимо предпринять специальные меры, чтобы «вытолкнуть» их оттуда.

Впрочем, иногда организации защищаются от повторного использования слишком ревностно – путем уничтожения магнитных носителей. На практике заведомо достаточно троекратной записи случайных последовательностей бит.

Метки безопасности

Для реализации принудительного управления доступом с субъектами и объектами используются метки безопасности. Метка субъекта описывает его благонадежность, метка объекта – степень закрытости содержащейся в нем информации. Согласно «Оранжевой книге», метки безопасности состоят из двух частей: уровня секретности и списка категорий. Уровни секретности, поддерживаемые системой, образуют упорядоченное множество, которое может выглядеть, например, так:

- совершенно секретно;
- секретно;
- конфиденциально;
- несекретно.

Категории образуют неупорядоченный набор. Их назначение – описать предметную область, к которой относятся данные. В военной области каждая категория может соответствовать, например, определенному виду вооружений. Механизм категорий позволяет разделить информацию «по отсекам», что способствует лучшей защищенности. Субъект не может получить доступ к «чужим» категориям, даже если его уровень благонадежности – «совершенно секретно». Специалист по танкам не узнает тактико-технические данные самолетов.

Главная проблема, которую необходимо решать в связи с метками, – это обеспечение их целостности. Во-первых, не долж-

но быть непомеченных субъектов и объектов, иначе в меточной безопасности появятся легко используемые бреши. Во-вторых, при любых операциях с данными метки должны оставаться правильными. В особенности это относится к экспорту и импорту данных. Например, печатный документ должен открываться заголовком, содержащим текстовое и/или графическое представление метки безопасности. Аналогично, при передаче файла по каналу связи должна передаваться и ассоциированная с ним метка, причем в таком виде, чтобы удаленная система могла ее протрактовать, несмотря на возможные различия в уровнях секретности и наборе категорий.

Метки безопасности субъектов более подвижны, чем метки объектов. Субъект может в течение сеанса работы с системой изменять свою метку, естественно, не выходя за предопределенные для него рамки. Иными словами, он может сознательно занижать свой уровень благонадежности, чтобы уменьшить вероятность непреднамеренной ошибки. Вообще, принцип минимизации привилегий – весьма разумное средство защиты.

Принудительное управление доступом. Принудительное управление доступом основано на сопоставлении меток безопасности субъекта и объекта. Субъект может читать информацию из объекта, если уровень секретности субъекта не ниже, чем у объекта, а все категории, перечисленные в метке безопасности объекта, присутствуют в метке субъекта. В таком случае говорят, что метка субъекта доминирует над меткой объекта. Смысл сформулированного правила понятен – читать можно только то, что положено.

Субъект может записывать информацию в объект, если метка безопасности объекта доминирует над меткой субъекта. В частности, «конфиденциальный» субъект может писать в секретные файлы, но не может – в несекретные (разумеется, должны также выполняться ограничения на набор категорий). На первый взгляд подобное ограничение может показаться странным, однако оно вполне разумно. Ни при каких операциях уровень секретности информации не должен понижаться, хотя обратный процесс вполне возможен. Посторонний человек может случайно узнать секретные сведения и сообщить их куда следует, однако лицо,

допущенное к работе с секретными документами, не имеет права раскрывать их содержание простому смертному.

Описанный способ управления доступом называется принудительным, поскольку он не зависит от воли субъектов (даже системных администраторов). После того как зафиксированы метки безопасности субъектов и объектов, оказываются зафиксированными и права доступа. В терминах принудительного управления нельзя выразить предложение «разрешить доступ к объекту X еще и для пользователя Y».

Конечно, можно изменить метку безопасности пользователя Y, но тогда он, скорее всего, получит доступ ко многим дополнительным объектам, а не только к X.

Принудительное управление доступом реализовано во многих вариантах операционных систем и СУБД, отличающихся повышенными мерами безопасности. В частности, такие варианты существуют для SunOS и СУБД Ingres. Независимо от практического использования принципы принудительного управления являются удобным методологическим базисом для начальной классификации информации и распределения прав доступа. Удобнее мыслить в терминах уровней секретности и категорий, чем заполнять неструктурированную матрицу доступа.

Классы безопасности. «Критерии оценки безопасности компьютерных систем» Министерства обороны США открыли путь к ранжированию информационных систем по степени надежности. В «Оранжевой книге» определяются четыре уровня надежности (безопасности) – D, C, B и A.

Уровень D предназначен для систем, признанных неудовлетворительными. В настоящее время он пуст, и ситуация едва ли когда-нибудь изменится. По мере перехода от уровня C к A к надежности систем предъявляются все более жесткие требования. Уровни C и B подразделяются на классы (C1, C2, B1, B2, B3) с постепенным возрастанием надежности. Таким образом, всего имеется шесть классов безопасности – C1, C2, B1, B2, B3, A1. Чтобы система в результате процедуры сертификации могла быть отнесена к некоторому классу, ее политика безопасности и гарантированность должны удовлетворять приводимым ниже требованиям. Поскольку при переходе к каждому следующему классу требования только добавляются, будем говорить лишь о том но-

вом, что присуще данному классу, группируя требования в согласии с предшествующим изложением.

Итак, ниже следуют критерии оценки надежных компьютерных систем.

Требования к политике безопасности. Требования к политике безопасности, проводимой системой, подразделяются в соответствии с основными направлениями политики, предусматриваемыми «Оранжевой книгой».

Произвольное управление доступом:

Класс С1 – вычислительная база должна управлять доступом именованных пользователей к именованным объектам. Механизм управления (права для владельца/группы/прочих, списки управления доступом) должен позволять специфицировать разделение файлов между индивидами и/или группами.

Класс С2 – в дополнение к С1, права доступа должны гранулироваться с точностью до пользователя. Механизм управления должен ограничивать распространение прав доступа – только авторизованный пользователь, например владелец объекта, может предоставлять права доступа другим пользователям. Все объекты должны подвергаться контролю доступа.

Класс В3 – в дополнение к С2, обязательно должны использоваться списки управления доступом с указанием разрешенных режимов. Должна быть возможность явного указания пользователей или их групп, доступ которых к объекту запрещен.

(Примечание. Поскольку классы В1 и В2 не упоминаются, требования к ним в плане добровольного управления доступом те же, что и для С2.

Аналогично, требования к классу А1 те же, что и для В3.)

Повторное использование объектов:

Класс С2 – при выделении хранимого объекта из пула ресурсов вычислительной базы необходимо ликвидировать все следы предыдущих использований.

Метки безопасности:

Класс В1 – вычислительная база должна управлять метками безопасности, связанными с каждым субъектом и хранимым объектом. Метки являются основой функционирования механизма принудительного управления доступом.

При импорте непомеченной информации соответствующий уровень секретности должен запрашиваться у авторизованного пользователя и все такие действия следует протоколировать.

Класс B2 – в дополнение к B1, помечаться должны все ресурсы системы, например ПЗУ, прямо или косвенно доступные субъектам.

Целостность меток безопасности:

Класс B1 – метки должны адекватно отражать уровни секретности субъектов и объектов. При экспорте информации метки должны преобразовываться в точное и однозначно трактуемое внешнее представление, сопровождающее данные. Каждое устройство ввода/вывода (в том числе коммуникационный канал) должно трактоваться как одноуровневое или многоуровневое. Все изменения трактовки и ассоциированных уровней секретности должны протоколироваться.

Класс B2 – в дополнение к B1, вычислительная база должна немедленно извещать терминального пользователя об изменении его метки безопасности. Пользователь может запросить информацию о своей метке. База должна поддерживать присваивание всем подключенным физическим устройствам минимального и максимального уровня секретности. Эти уровни должны использоваться при проведении в жизнь ограничений, налагаемых физической конфигурацией системы, например расположением устройств.

Принудительное управление доступом:

Класс B1 – вычислительная база должна обеспечить проведение в жизнь принудительного управления доступом всех субъектов ко всем хранимым объектам.

Субъектам и объектам должны быть присвоены метки безопасности, являющиеся комбинацией упорядоченных уровней секретности, а также категорий. Метки являются основой принудительного управления доступом. Надежная вычислительная база должна поддерживать по крайней мере два уровня секретности.

Вычислительная база должна контролировать идентификационную и аутентификационную информацию. При создании новых субъектов, например процессов, их метки безопасности не должны доминировать над меткой породившего их пользователя.

Класс B2 – в дополнение к B1, все ресурсы системы (в том числе ПЗУ, устройства ввода/вывода) должны иметь метки безопасности и служить объектами принудительного управления доступом.

Требования к подотчетности

Идентификация и аутентификация:

Класс C1 – пользователи должны идентифицировать себя, прежде чем выполнять какие-либо иные действия, контролируемые вычислительной базой. Для аутентификации должен использоваться какой-либо защитный механизм, например пароли. Аутентификационная информация должна быть защищена от несанкционированного доступа.

Класс C2 – в дополнение к C1, каждый пользователь системы должен уникальным образом идентифицироваться. Каждое регистрируемое действие должно связываться с конкретным пользователем.

Класс B1 – в дополнение к C2, вычислительная база должна поддерживать метки безопасности пользователей.

Предоставление надежного пути:

Класс B2 – вычислительная база должна поддерживать надежный коммуникационный путь к себе для пользователя, выполняющего операции начальной идентификации и аутентификации. Инициатива в общении по этому пути должна исходить исключительно от пользователя.

Класс B3 – в дополнение к B2, коммуникационный путь может формироваться по запросу, исходящему как от пользователя, так и от самой базы. Надежный путь может использоваться для начальной идентификации и аутентификации, для изменения текущей метки безопасности пользователя и т.п. Общение по надежному пути должно быть логически отделено и изолировано от других информационных потоков.

Аудит:

Класс C2 – вычислительная база должна создавать, поддерживать и защищать журнал регистрационной информации, относящейся к доступу к объектам, контролируемым базой. Должна быть возможность регистрации следующих событий:

использование механизма идентификации и аутентификации;

внесение объектов в адресное пространство пользователя, например открытие файла, запуск программы;

удаление объектов;

действия системных операторов, системных администраторов, администраторов безопасности;

другие события, затрагивающие информационную безопасность.

Каждая регистрационная запись должна включать следующие поля:

дата и время события;

идентификатор пользователя;

тип события;

результат действия (успех или неудача).

Для событий идентификации/аутентификации регистрируется также идентификатор устройства, например терминала. Для действий с объектами регистрируются имена объектов. Системный администратор может выбирать набор регистрируемых событий для каждого пользователя.

Класс B1 – в дополнение к C2, должны регистрироваться операции выдачи на печать и ассоциированные внешние представления меток безопасности.

При операциях с объектами, помимо имен, регистрируются их метки безопасности. Набор регистрируемых событий может различаться в зависимости от уровня секретности объектов.

Класс B2 – в дополнение к B1, должна быть возможность регистрировать события, связанные с организацией тайных каналов с памятью.

Класс B3 – в дополнение к B2, должна быть возможность регистрации появления или накопления событий, несущих угрозу политике безопасности системы. Администратор безопасности должен немедленно извещаться о попытках нарушения политики безопасности, а система, в случае продолжения попыток, должна пресекать их наименее болезненным способом.

Требования к гарантированности

Архитектура системы:

Класс C1 – вычислительная база должна поддерживать область для собственного выполнения, защищенную от внешних воздействий, в частности от изменения команд и/или данных, и от попыток слежения за ходом работы. Ресурсы, контролируемые базой, могут составлять определенное подмножество всех субъектов и объектов системы.

Класс C2 – в дополнение к C1, вычислительная база должна изолировать защищаемые ресурсы в той мере, как это диктуется требованиями контроля доступа и подотчетности.

Класс B1 – в дополнение к C2, вычислительная база должна обеспечивать взаимную изоляцию процессов путем разделения их адресных пространств.

Класс B2 – в дополнение к B1, вычислительная база должна быть внутренне структурирована на хорошо определенные, относительно независимые модули.

Вычислительная база должна эффективно использовать имеющееся оборудование для отделения элементов, критически важных с точки зрения защиты, от прочих компонентов системы. Модули базы должны проектироваться с учетом принципа минимизации привилегий. Для защиты логически раздельных хранимых объектов должны использоваться аппаратные средства, например сегментация. Должен быть полностью определен пользовательский интерфейс с вычислительной базой.

Класс B3 – в дополнение к B2, вычислительная база должна быть спроектирована и структурирована таким образом, чтобы использовать полный и концептуально простой защитный механизм. Этот механизм должен играть центральную роль во внутренней структуризации вычислительной базы и всей системы. База должна активно использовать разделение данных по уровням. Значительные инженерные усилия должны быть направлены на уменьшение сложности вычислительной базы и на вынесение из нее модулей, не являющихся критически важными с точки зрения защиты.

Целостность системы:

Класс С1 – должны быть в наличии аппаратные и/или программные средства, позволяющие периодически проверять корректность функционирования аппаратных и микропрограммных компонентов вычислительной базы.

Анализ тайных каналов передачи информации:

Класс В2 – системный архитектор должен тщательно проанализировать возможности по организации тайных каналов с памятью и оценить максимальную пропускную способность каждого выявленного канала.

Класс В3 – в дополнение к В2, аналогичная процедура должна быть проделана для временных каналов.

Класс А1 – в дополнение к В3, для анализа должны использоваться формальные методы.

Надежное администрирование:

Класс В2 – система должна поддерживать разделение функций оператора и администратора.

Класс В3 – в дополнение к В2, должна быть специфицирована роль администратора безопасности. Получить права администратора безопасности можно только после выполнения явных, протоколируемых действий. Не относящиеся к защите действия администратора безопасности должны быть по возможности ограничены.

Надежное восстановление:

Класс В3 – должны существовать процедуры и/или механизмы, позволяющие произвести восстановление после сбоя или иного нарушения работы без ослабления защиты.

Тестирование:

Класс С1 – защитные механизмы должны быть протестированы на предмет соответствия их поведения системной документации. Тестирование должно подтвердить, что у неавторизованного пользователя нет очевидных способов обойти или разрушить средства защиты вычислительной базы.

Класс С2 – в дополнение к С1, тестирование должно подтвердить отсутствие очевидных недостатков в механизмах изоляции ресурсов и защиты регистрационной информации.

Класс В1 – в дополнение к С2, группа специалистов, полностью понимающих конкретную реализацию вычислительной ба-

зы, должна подвергнуть описание архитектуры, исходные и объектные коды тщательному анализу и тестированию. Цель должна состоять в выявлении всех дефектов архитектуры и реализации, позволяющих субъекту без должной авторизации читать, изменять, удалять информацию или приводить базу в состояние, когда она перестает обслуживать запросы других субъектов. Все выявленные недостатки должны быть исправлены или нейтрализованы, после чего база подвергается повторному тестированию, чтобы убедиться в отсутствии прежних или приобретении новых недостатков.

Класс B2 – в дополнение к B1, должна быть продемонстрирована относительная устойчивость вычислительной базы к попыткам проникновения.

Класс B3 – в дополнение к B2, должна быть продемонстрирована устойчивость вычислительной базы к попыткам проникновения.

Класс A1 – в дополнение к B3, тестирование должно продемонстрировать, что реализация вычислительной базы соответствует формальным спецификациям верхнего уровня. Основу тестирования средств защиты от проникновения в систему должно составлять наличие спецификаций на исходные тексты.

Верификация спецификаций архитектуры:

Класс B1 – должна существовать неформальная или формальная модель политики безопасности, поддерживаемой вычислительной базой. Модель должна соответствовать основным посылкам политики безопасности на протяжении всего жизненного цикла системы.

Класс B2 – в дополнение к B1, модель политики безопасности должна быть формальной. Для вычислительной базы должны существовать описательные спецификации верхнего уровня, точно и полно определяющие ее интерфейс.

Класс B3 – в дополнение к B2, должны быть приведены убедительные аргументы соответствия между спецификациями и моделью.

Класс A1 – в дополнение к B3, помимо описательных должны быть представлены формальные спецификации верхнего уровня, относящиеся к аппаратным и/или микропрограммным элементам, составляющим интерфейс вычислительной базы.

Комбинация формальных и неформальных методов должна подтвердить соответствие между спецификациями и моделью. Должны использоваться современные методы формальной спецификации и верификации систем, доступные Национальному центру компьютерной безопасности США.

Конфигурационное управление:

Класс B2 – в процессе разработки и сопровождения вычислительной базы должна использоваться система конфигурационного управления, обеспечивающая контроль за изменениями в описательных спецификациях верхнего уровня, иных архитектурных данных, реализационной документации, исходных текстах, работающей версии объектного кода, тестовых данных и документации. Конфигурационное управление должно обеспечивать соответствие друг другу всех аспектов текущей версии вычислительной базы. Должны предоставляться средства генерации новых версий базы по исходным текстам и средства для сравнения версий, чтобы убедиться в том, что произведены только запланированные изменения.

Класс A1 – в дополнение к B2, механизм конфигурационного управления должен распространяться на весь жизненный цикл и все компоненты системы, имеющие отношение к обеспечению безопасности, включая спецификации и документацию. Для защиты эталонной копии материалов, использующихся для генерации надежной вычислительной базы, должна использоваться комбинация физических, административных и технических мер.

Надежное распространение:

Класс A1 – должна поддерживаться целостность соответствия между эталонными данными, описывающими текущую версию вычислительной базы, и эталонной копией текстов этой версии. Должны существовать процедуры, подтверждающие соответствие между поставляемыми клиентам аппаратными и программными компонентами и эталонной копией.

Требования к документации

Руководство пользователя по средствам безопасности:

Класс C1 – отдельный фрагмент документации (глава, том) должен описывать защитные механизмы, предоставляемые вы-

числительной базой, и их взаимодействие между собой, содержать рекомендации по их использованию.

Руководство администратора по средствам безопасности:

Класс С1 – руководство должно содержать сведения о функциях и привилегиях, которыми управляет системный администратор посредством механизмов безопасности.

Класс С2 – в дополнение к С1, должны описываться процедуры обработки регистрационной информации и управления файлами с такой информацией, а также структура записей для каждого типа регистрируемых событий.

Класс В1 – в дополнение к С2, руководство должно описывать функции оператора и администратора, затрагивающие безопасность, в том числе действия по изменению характеристик пользователей. Должны быть представлены рекомендации по согласованному и эффективному использованию средств безопасности, их взаимодействию друг с другом, по безопасной генерации новых версий вычислительной базы.

Класс В2 – в дополнение к В1, должны быть указаны модули вычислительной базы, содержащие механизмы проверки обращений. Должна быть описана процедура безопасной генерации новой версии базы после внесения изменений в исходные тексты.

Класс В3 – в дополнение к В2, должна быть описана процедура, обеспечивающая безопасность начального запуска системы и возобновления ее работы после сбоя.

Тестовая документация:

Класс С1 – разработчик системы должен представить экспертному совету документ, содержащий план тестов, процедуры прогона тестов и результаты тестов.

Класс В2 – в дополнение к С1, тесты должны подтверждать действенность мер по уменьшению пропускной способности тайных каналов передачи информации.

Класс А1 – в дополнение к В2, должно быть описано соответствие между формальными спецификациями верхнего уровня и исходными текстами.

Описание архитектуры:

Класс С1 – должны быть описаны подход к безопасности, используемый производителем, и применение этого подхода при

реализации вычислительной базы. Если база состоит из нескольких модулей, должен быть описан интерфейс между ними.

Класс B1 – в дополнение к C1, должно быть представлено неформальное или формальное описание модели политики безопасности, проводимой в жизнь вычислительной базой. Необходимо наличие аргументов в пользу достаточности избранной модели для реализации политики безопасности. Должны быть описаны защитные механизмы базы и их место в модели.

Класс B2 – в дополнение к B1, модель политики безопасности должна быть формальной и доказательной. Должно быть показано, что описательные спецификации верхнего уровня точно отражают интерфейс вычислительной базы. Должно быть показано, как база реализует концепцию монитора обращений, почему она устойчива к попыткам отслеживания ее работы, почему ее нельзя обойти и почему она реализована корректно. Должна быть описана структура базы, чтобы облегчить ее тестирование и проверку соблюдения принципа минимизации привилегий. Документация должна содержать результаты анализа тайных каналов передачи информации и описание мер протоколирования, помогающих выявлять каналы с памятью.

Класс B3 – в дополнение к B2, должно быть неформально продемонстрировано соответствие между описательными спецификациями верхнего уровня и реализацией вычислительной базы.

Класс A1 – в дополнение к B3, должно быть неформально продемонстрировано соответствие между формальными спецификациями верхнего уровня и реализацией вычислительной базы.

Таковы, согласно «Оранжевой книге», требования к классам безопасности информационных систем.

ПРИЛОЖЕНИЕ 3

Подсистема регистрации программно-аппаратного комплекса средств защиты информации от несанкционированного доступа «АККОРД 1.95-00»

Программа работы с журналами регистрации «LogView»

1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

Программа LOGVIEW.EXE предназначена для работы с журналами регистрации, которые создаются в процессе функционирования подсистемы разграничения доступа ПАК СЗИ «Аккорд-1.95-00». Для каждого сеанса работы пользователя создается отдельный файл журнала. Имя файла генерируется с помощью системной даты, времени и некоторой случайной компоненты, чтобы исключить совпадение имен файлов журнала.

2. ОБЩИЕ СВЕДЕНИЯ О РАБОТЕ

При работе программно-аппаратного комплекса средств защиты от несанкционированного доступа «Аккорд-1.95-00» события регистрируются в файлах журнала в упакованном формате. Если комплекс защиты используется в сетевой версии, то в журнале фиксируется имя станции.

Запустить программу LOGVIEW.EXE из каталога ACCORD. При запуске программы на экран выводится окно выбора журнала (рис. 3.1) для просмотра.

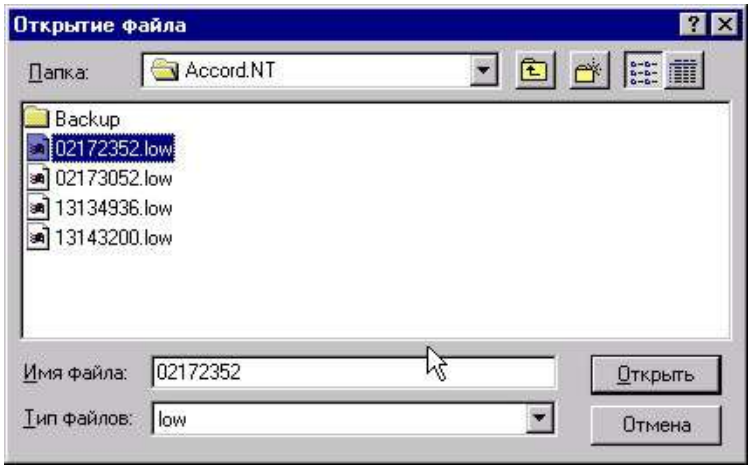


Рис. 3.1 – Окно выбора файла журнала

Выбрав мышью нужный файл, необходимо нажать кнопку «Открыть». На экран выводится окно просмотра журнала (рис. 3.2).

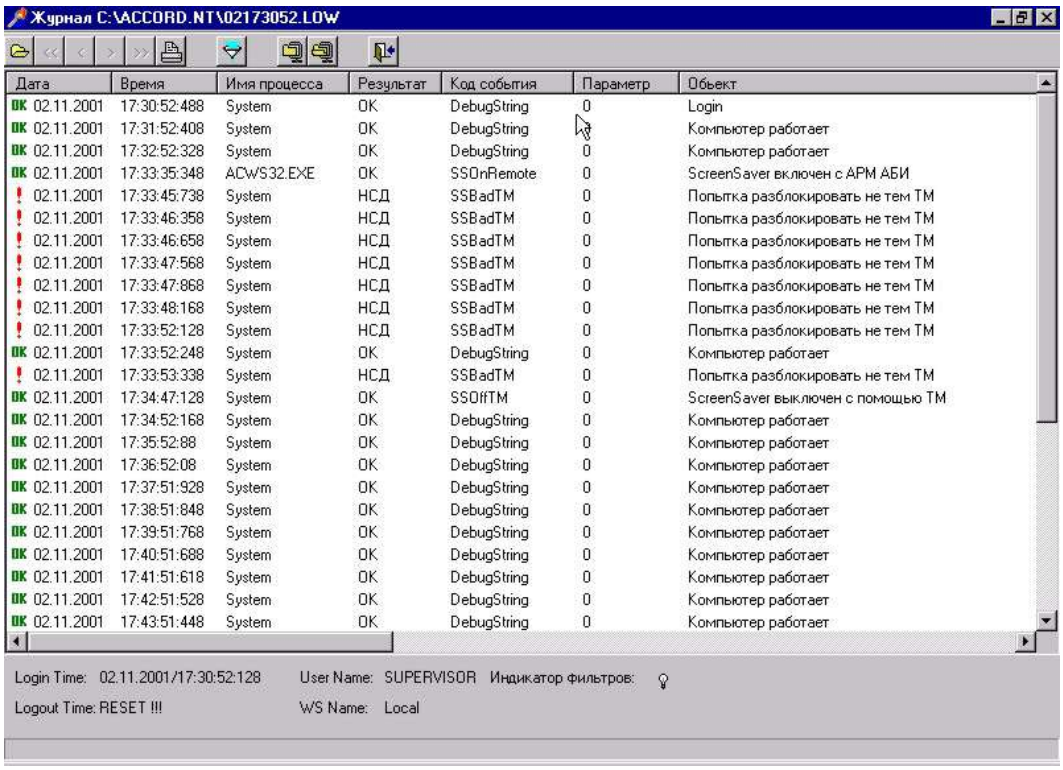


Рис. 3.2 – Главное окно программы LOGVIEW

По умолчанию выводятся следующие параметры регистрации:

- дата;
- время с точностью до тысячных долей секунды;
- имя процесса, который выполнил операцию;
- результат операции:
  - НСД – попытка несанкционированного доступа;
  - ОК – нормальное выполнение операции;
  - Ошибка – системная ошибка при выполнении операции;
- код события;
- параметр;
- объект.

Просмотр журнала можно выполнять в расширенном режиме, в этом случае выводятся все поля базы данных регистрации. Такой режим может потребоваться только для очень подробного анализа событий и в большинстве случаев избыточен.

С помощью «мыши» можно изменять ширину колонок. В нижней панели окна выводится имя пользователя и рабочей

станции, а также дата и время начала и окончания сеанса данного пользователя. Если сеанс был завершен не стандартными средствами ОС, а выключением питания компьютера, то в поле Logout Time выводится слово «RESET!!!».

В верхней части окна расположены функциональные кнопки. При установке на клавишу курсора мыши выводится подсказка. Для работы с журналом доступны следующие команды:

- загрузить файл – выбор файла журнала для просмотра;
- на первую страницу – быстрый переход в начало файла;
- на страницу вперед – переход на следующую страницу;
- на страницу назад – переход на предыдущую страницу;
- на последнюю страницу – быстрый переход в конец файла;
- печать страницы – вывод на печать текущей страницы;
- установить/снять все фильтры;
- добавить в архив – добавить в архив файл, или группу файлов журнала;
- извлечь из архива – извлечь файлы журнала из архива;
- выход из программы;
- включить/выключить детализацию.

3. АРХИВАЦИЯ/РАЗАРХИВАЦИЯ ЖУРНАЛОВ

При выборе кнопки «Добавить в архив» выводится окно выбора файла архива (рис. 3.3). Если файл архива уже существует, то его можно выбрать мышью, если в строке «Имя файла» ввести наименование нового архива, он будет создан.

Рис. 3.3 – Выбор файла архива

Далее в открывшемся окне необходимо выбрать файл, или группу файлов для архивации.

При выборе кнопки «Извлечь из архива» выводится окно выбора файла архива (рис. 3.4). Файл архива можно выбрать мышью или в строке «Имя файла» ввести имя архива. В открывшемся окне можно выбрать каталог, в который будут помещены разархивированные файлы журнала. При этой операции происходит извлечение всех файлов из выбранного архива.

Рис. 3.4 – Окно выбора файла архива для извлечения журнала

4. ПРОСМОТР ЖУРНАЛА РЕГИСТРАЦИИ СОБЫТИЙ

В этом режиме для удобства просмотра и анализа журнала можно устанавливать фильтры для отдельных полей базы данных.

- **Фильтрация по имени процесса**

Для вывода на экран окна установки фильтра (рис. 3.5.) необходимо установить курсор на заголовке колонки «Имя процесса» и нажать левую кнопку мыши.

Рис. 3.5 – Установка фильтра по имени процесса

Можно ввести как полное имя процесса, так и часть имени. После нажатия на кнопку «Ок» происходит поиск в журнале, и на экран выводятся только те записи, которые удовлетворяют заданному критерию фильтрации. Поиск производится без учета регистра введенных символов.

- **Фильтрация по результату операции**

Для фильтрации по результату операции необходимо установить курсор на заголовке колонки «Результат операции» и нажать левую кнопку мыши. На экран выводится окно установки фильтра (рис. 3.6).

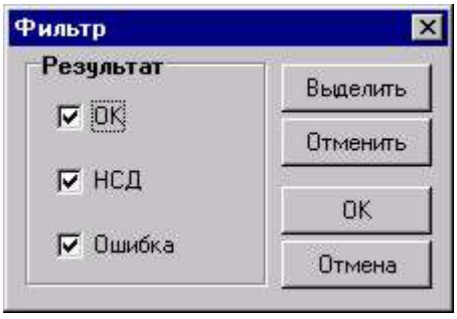


Рис. 3.6 – Установка фильтра по результату операции

Нажатием на левую клавишу «мыши» можно установить/сбросить отметку возле каждой операции. После нажатия кнопки «Ок» на экран выводятся только те события, результат которых совпадает с операциями, отмеченными для фильтрации.

- **Фильтрация по коду событий**

Установить курсор на заголовке колонки «Код события» и нажать левую кнопку мыши. На экран выводится окно установки фильтра (рис. 3.7).

Все события, регистрируемые СЗИ «Аккорд», сгруппированы в пять классов, для каждого из которых можно установить или снять отметку фильтрации. При нажатии на кнопку «Выбор» выводится полный список событий данной группы. Конкретно для каждого события в списке также можно установить отметку выбора для фильтрации.

Рис. 3.7 – Установка фильтра по коду события

Рассмотрим подробнее регистрируемые события.

- **Сообщения СЗИ:**

Рис. 3.8 – Установка фильтра для событий класса «Сообщения СЗИ»

В этом классе событий фиксируется только одно событие – DebugString – это сообщения, возникающие при работе СЗИ «Аккорд». Хотя событие одно, но его содержание может быть различным, и текст этих сообщений отображается в поле «Объект».

- **Screen-Sever**

В этой группе собраны события, которые относятся к обработке операций блокировки и разблокировки экрана и клавиатуры (рис. 3.9).

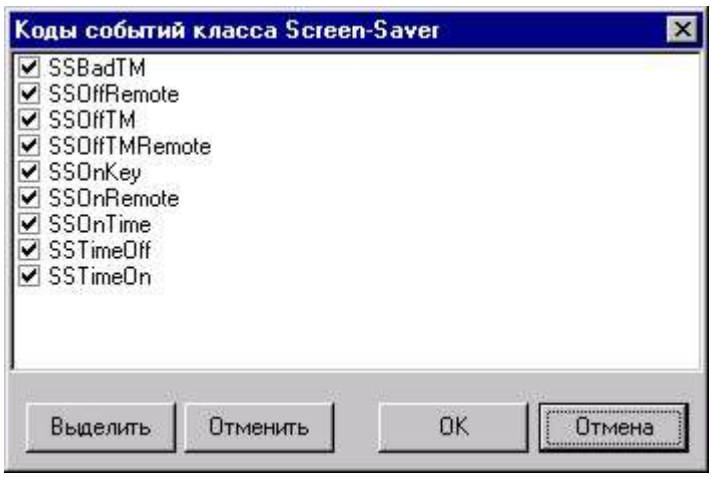


Рис. 3.9 – Установка фильтра для событий класса «Screen-Saver»

- **Проверка файлов**

В этой группе собраны события, которые относятся к операциям контроля целостности файлов и процессов (рис. 3.10).

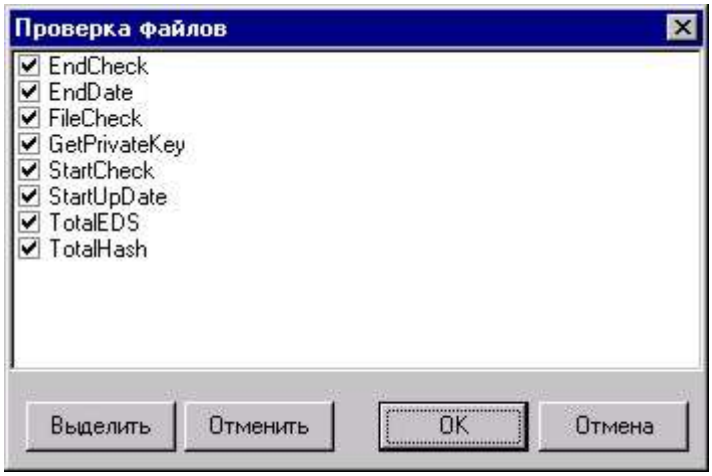


Рис. 3.10 – Установка фильтра для событий класса «Проверка файлов»

- **PM**

Это группа событий, которые относятся к контролю файловых операций (рис. 3.11).

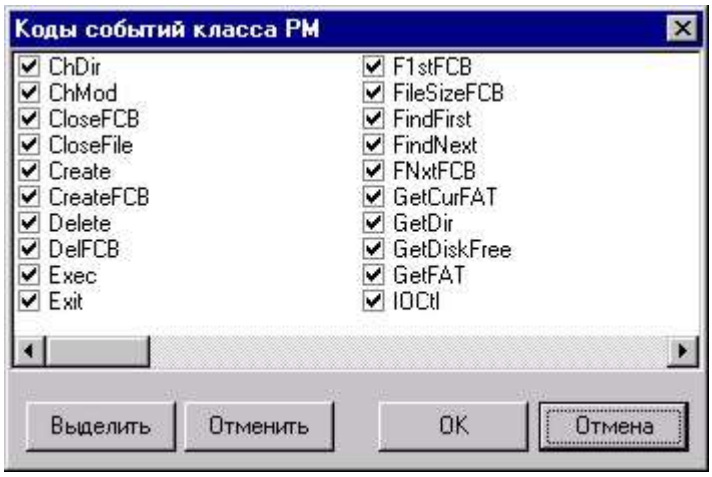


Рис. 3.11 – Установка фильтра для событий класса «PM»

➢ **Registry**

События данной группы регистрируются только в том случае, когда в настройках СЗИ «Аккорд» установлен флаг «Контролировать доступ к реестру». Список событий приводится на рис. 3.12.

Следует отметить, что для операций в журнале событий можно получить полное название операции. Для этого достаточно установить курсор на нужный Вам код события и нажать левую кнопку «мыши». В нижней строке окна появится полное название события.

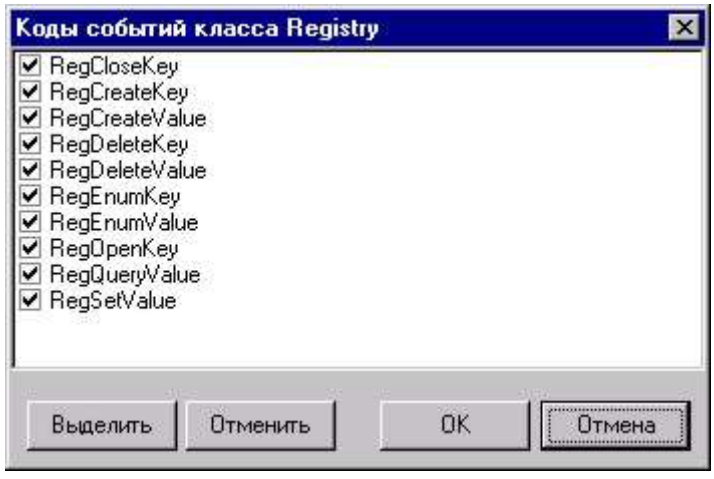


Рис. 3.12 – Установка фильтра для событий класса «Registry»

- **Фильтрация по наименованию объекта**

Установить курсор на заголовке колонки «Объект» и нажать левую кнопку мыши. На экран выводится окно установки фильтра (рис. 3.13).

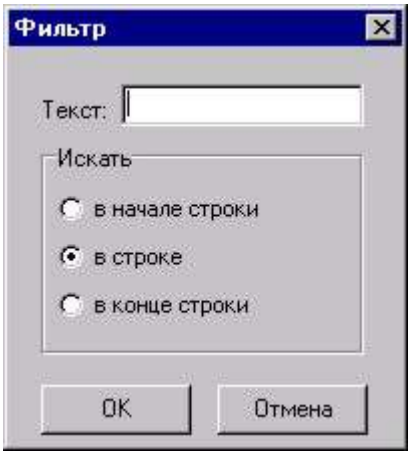


Рис. 3.13 – Установка фильтра по наименованию объекта

5. ВЫВОД НА ПЕЧАТЬ

Для вывода на печатающее устройство необходимо левой клавишей «мыши» щелкнуть на кнопке с пиктограммой принтера в верхней строке окна. На печать выводится текущее отображение журнала (с установленными фильтрами). Такой режим печати предусмотрен потому, что объем журнала, особенно с высоким уровнем детализации, может составлять сотни килобайт.

6. ПРЕДВАРИТЕЛЬНАЯ СОРТИРОВКА ЖУРНАЛОВ

Для более эффективной организации работы с журналами регистрации событий следует воспользоваться программой LOGBASE.EXE. Эта программа выполняет быстрый просмотр всех журналов, расположенных в текущем каталоге, и выводит список этих журналов, отсортированный по именам пользователей и дате/времени (рис. 3.14). Можно отметить мышью необходимые файлы и нажать кнопку «Просмотр». Программа LogView загрузится и откроет для просмотра выбранные файлы.

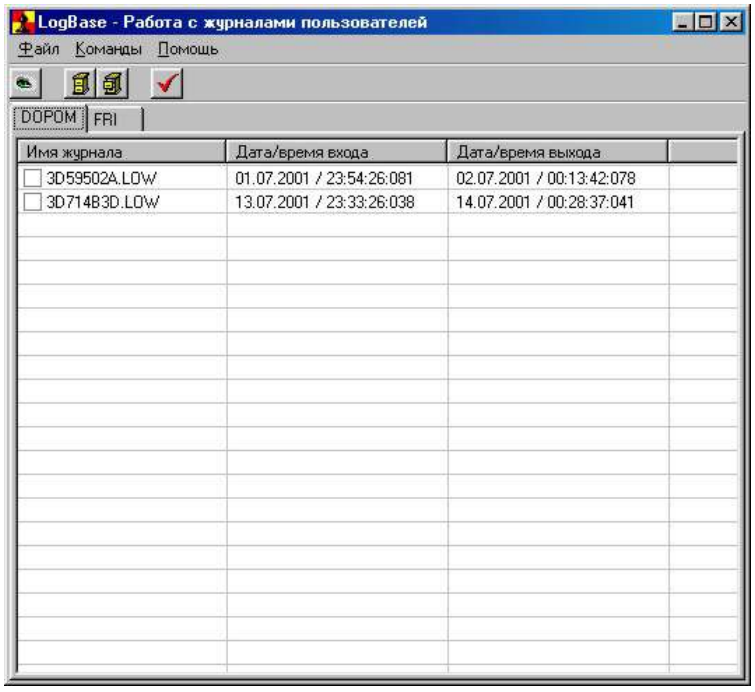


Рис. 3.14 – Предварительная сортировка журналов регистрации

ПРИЛОЖЕНИЕ 4

ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ В WINDOWS NT

Статистика показывает, что в большинстве случаев несанкционированного проникновения в систему можно избежать, если системный администратор уделяет должное внимание средствам защиты. Эффективность обеспечения безопасности компьютерных систем всегда зависит от качества настройки программно-аппаратных средств. Операционная система Windows NT имеет богатый набор средств защиты. Однако установленные по умолчанию значения параметров защиты не всегда удовлетворяют предъявляемым требованиям. Рассмотрим основные средства и методы обеспечения безопасности, входящие в состав Windows NT 4.0 и 5.0.

1. ***Физическая защита***

К физическим средствам защиты относится:

- обеспечение безопасности помещений, где размещены серверы сети;
- ограничение посторонним лицам физического доступа к серверам, концентраторам, коммутаторам, сетевым кабелям и другому оборудованию;
- использование средств защиты от сбоев электросети.

2. ***Администрирование учетных записей***

В функции Менеджера учетных записей входит поддержка механизма идентификации и проверки подлинности пользователей при входе в систему. Все необходимые настройки хранятся в базе данных Менеджера учетных записей. К ним относится:

- учетные записи пользователей;
- учетные записи групп;
- учетные записи компьютеров домена;
- учетные записи доменов.

База данных Менеджера учетных записей представляет собой куст системного реестра, находящегося в ветви HKEY_LOCAL_MACHINE, и называется SAM (рис. 4.1). Как и все остальные кусты, он хранится в отдельном файле в каталоге

%Systemroot%\System32\Config, который также носит название SAM. В этом каталоге обычно находятся минимум два файла SAM: один без расширения – сама база учетных записей; второй имеет расширение .log – журнал транзакций базы.

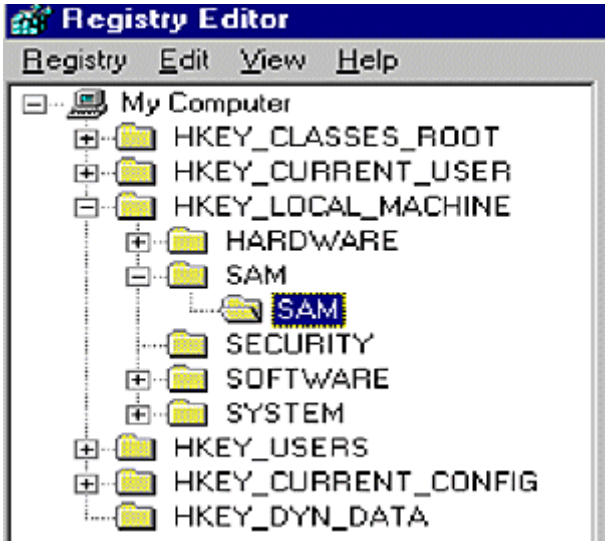


Рис. 4.1

Наиболее интересным является раздел учетных записей пользователей: в них хранится информация об именах и паролях. Следует заметить, что пароли не хранятся в текстовом виде. Они защищены процедурой хеширования. Это не значит, что, не зная пароля в текстовом виде, злоумышленник не проникнет в систему. При сетевом подключении не обязательно знать текст пароля, достаточно хешированного пароля. Поэтому достаточно получить копию базы данных SAM и извлечь из нее хешированный пароль.

При установке системы Windows NT доступ к файлу %Systemroot%\System32\Config\sam для обычных программ заблокирован. Однако, используя утилиту Ntbackup, любой пользователь с правом Backup files and directories может скопировать его. Кроме того, злоумышленник может попытаться переписать его копию (Sam.sav) из каталога %Systemroot%\System32\Config или архивную копию (Sam.) из каталога %Systemroot%\Repair.

Поэтому для защиты информации, хранящейся в базе данных SAM, необходимо следующее:

- исключить загрузку серверов в DOS-режиме (все разделы установить под NTFS, отключить загрузку с флоппи- и компакт-

дисков, желательно установить на BIOS пароль (хотя эта мера уже давно устарела, поскольку некоторые версии BIOS имеют «дырки» для запуска компьютера без пароля, все-таки злоумышленник потеряет на этом время для входа в систему);

- ограничить количество пользователей с правами Backup Operators и Server Operators;
- после установки или обновления удалить файл Sam.sav;
- отменить кэширование информации о безопасности на компьютерах домена (имена и пароли последних десяти пользователей, регистрировавшихся ранее на данном компьютере, сохраняются в его локальном реестре).

Один из популярных методов проникновения в систему – подбор пароля. Для борьбы с этим обычно устанавливают блокировку учетной записи пользователя (Account Lockout) после определенного числа неудачных попыток входа, используя для этого утилиту User Manager в диалоговом окне Account Policy, доступном через меню Polcies/Accounts (рис. 4.2).

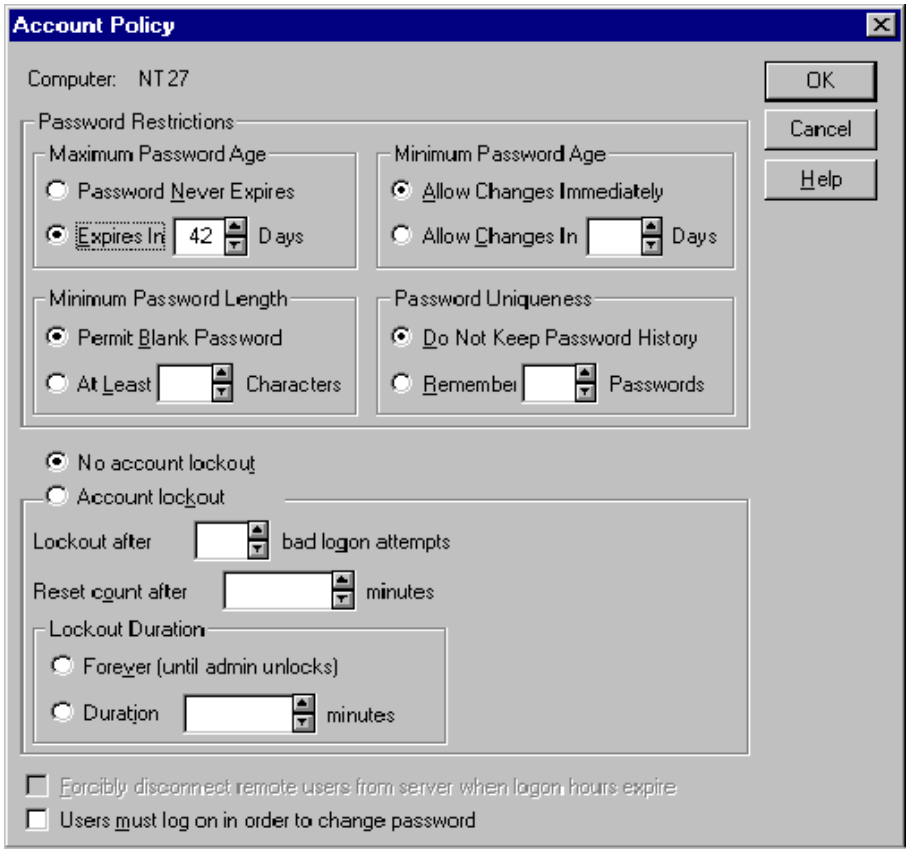


Рис. 4.2

Исключением является учетная запись администратора. И если он имеет право на вход через сеть, это открывает лазейку для спокойного угадывания пароля. Для защиты рекомендуется переименовать пользователя Administrator, установить блокировку учетных записей, запретить администратору вход в систему через сеть, запретить передачу SMB пакетов через TCP/IP (порты 137,138,139), установить протоколирование неудачных входов; необходимо ввести фильтрацию вводимых пользователем паролей, установить Service Pack 2 или 3 (используется динамическая библиотека Passfilt.dll).

Для включения данной фильтрации необходимо в реестре в разделе HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa добавить

Параметр	Notification Packages
Тип	REG_MULTI_SZ
Значение	PASSFILT

Если этот параметр уже существует и содержит величину FPNWCLNT (File Personal NetWare Client), то необходимо дописать новую строку под FPNWCLNT. Если окажется, что наборов фильтра мало, то создайте свою библиотеку, используя статью Q151082 в Microsoft KnowledgeBase, где приведен пример написания модуля фильтра.

3. *Защита файлов и каталогов (папок) в Windows NT 4.0*

Операционная система Windows NT 4.0 поддерживает файловые системы FAT (File Allocation Table) и NTFS (New Technology File System). Первая поддерживается такими известными операционными системами, как MS-DOS, Windows 3.X, Windows 95/98 и OS/2, вторая – только Windows NT. У FAT и NTFS различные характеристики производительности, разный спектр предоставляемых возможностей и т.д. Основное отличие файловой системы NTFS от других (FAT, VFAT (Virtual File Allocation Table), HPFS) состоит в том, что только она одна удовлетворяет стандарту безопасности C2, в частности, NTFS обеспечивает защиту файлов и каталогов при локальном доступе.

Защиту ресурсов с использованием FAT можно организовать с помощью прав доступа: *Чтение, Запись, Полный.*

Таким образом, можно рекомендовать создавать дисковые разделы NTFS вместо FAT. Если все же необходимо использовать раздел FAT, то его надо сделать отдельным разделом для приложений MS-DOS и не размещать в нем системные файлы Windows NT.

Поскольку файлы и каталоги в Windows NT являются объектами, контроль безопасности осуществляется на объектном уровне. Дескриптор безопасности любого объекта в разделе NTFS содержит два списка контроля доступа (ACL) – дискреционный (discretionary ACL (DACL)) и системный (system ACL (SACL)).

В операционной системе Windows NT управление доступом к файлам и каталогам NTFS возлагается не на администратора, а на владельца ресурса и контролируется системой безопасности с помощью маски доступа (access mask), содержащейся в записях списка контроля доступа ACL.

Маска доступа включает стандартные (Synchronize, Write_Owner, Write_Dac, Read_Control, Delete), специфические (Read (Write) _Data, Append_Data, Read(Write)_Attributes, Read(Write)_ExtendedAttributes, Execute) и родовые (Generic_ Read(Write), Generic_Execute) права доступа. Все эти права входят в дискреционный список контроля доступа (DACL). Вдобавок маска доступа содержит бит, который соответствует праву Access_System_Security. Это право контролирует доступ к системному списку контроля доступа (SACL).

В списке DACL определяется, каким пользователям и группам разрешен или запрещен доступ к данному ресурсу. Именно этим списком может управлять владелец объекта.

Список SACL задает определенный владельцем тип доступа, что заставляет систему генерировать записи проверки в системном протоколе событий. Только системный администратор управляет этим списком.

На самом же деле для администрирования используются не отдельные права доступа, а разрешения (permissions) NTFS. Разрешения подразделяются на:

➢ индивидуальные – набор прав, позволяющий предоставлять пользователю доступ того или иного типа (табл. 1);

➢ стандартные – наборы индивидуальных разрешений для выполнения над файлами или каталогами действий определенного уровня (табл. 2);

➢ специальные – комбинация индивидуальных разрешений, не совпадающих ни с одним стандартным набором (табл. 3).

По умолчанию при инсталляции Windows NT и файловой системы NTFS устанавливаются довольно «свободные» разрешения, позволяющие обычным пользователям получать доступ к ряду системных файлов и каталогам.

Существует несколько файлов операционной системы, расположенных в корневой директории системного раздела, которые также необходимо защитить, назначив следующие разрешения (табл. 4).

Необходимо иметь в виду, что такие разрешения затруднят пользователям установку программного обеспечения. Также будет невозможна запись в.ini файлы в системном каталоге.

Таблица 1

Разрешение	**Права доступа**	**Операция над**	
		файлами	папками
Read	Read_Control Read_Data Read_Atributes Read_EA Synchronize	Операции чтения файла, просмотр атрибутов, прав доступа, а также имени владельца	Операции отображения содержимого папки, просмотра атрибутов, прав доступа, а также имени ее владельца
Write	Read_Control Write_Data Append_Data Write_Atributes Write_EA Synchronize	Операции изменения файла и его атрибутов, просмотра прав доступа и имени владельца	Операции создания подпапок и файлов, изменения атрибутов файлов, просмотра прав доступа и имени владельца
Execute	Read_Control Read_Atributes Synchronize Execute	Операции запуска программы, просмотр атрибутов, прав	Операции просмотра атрибутов и прав доступа, а также имени владельца и

Продолжение табл. 1

Разрешение	**Права доступа**	**Операция над**	
		файлами	папками
		доступа, а также имени владельца	изменения подпапок
Delete	Delete	Операции удаления файла	Операции удаления папок
Change Permission	Write_Dac	Операции изменения прав доступа	Операции изменения прав доступа
Take Ownership	Write_Owner	Операции изменения владельца файла	Операции изменения владельца папки

Таблица 2

Разрешение	**Индивидуальные разрешения**	**Операции**
No Access	Нет	Запрещение доступа к файлу. Пользователь, для которого оно установлено, не может получить доступ к файлу даже в том случае, если он входит в группу пользователей, имеющих права доступа к данному документу
Read	Read, Execute	Предоставление пользователю права на просмотр файлов и запуск приложений, хранящихся в папке
Change	Read, Write, Execute, Delete	Разрешение (дополнительно к правам, предоставляемым правом Read) на создание и удаление файлов и папок, модификацию содержимого файлов
Full Control	Все	Разрешение (дополнительно к правам, предоставляемым правом Change) на изменение прав доступа и вступление во владение файлами и папками

Количество пользователей с правами администратора рекомендуется свести к минимуму. Учетную запись Guest лучше вообще удалить, хотя она при установке (по умолчанию) и так отключена, а вместо этой учетной записи создать для каждого пользователя свою временную учетную запись с соответствующими разрешениями и правами.

Таблица 3

Разрешение	**Разрешения к**		**Операции**
	Папкам	файлам	
No Access	Нет	Нет	Запрещение доступа к папке и содержащимся в ней файлам
List	Read, Execute	Не устанавливает	Разрешение на просмотр имен файлов и содержимого папок, а также их структуры
Read	Read, Execute	Read, Execute	Предоставление пользователю права на просмотр файлов и запуск приложений, хранящихся в папке
Add	Write, Execute	Не устанавливает	Разрешения (дополнительно к правам, предоставляемым правом Read) создавать папки и файлы. Не позволяет отображать структуру папок
Add & Read	Read, Write, Execute	Read, Execute	Предоставление прав, указанных в правах Add и Read
Change	Read, Write, Execute, Delete	Read, Write, Execute, Delete	Разрешение (дополнительно к правам, предоставляемым правами Add и Read) создавать и удалять файлы и папки, модифицировать содержимое файлов
Full Control	Все	Все	Разрешение (дополнительно к правам, предоставляемым правом Change) на изменение прав доступа и вступление во владение файлами и папками

Таблица 4

Объект защиты	Учетная запись	Разрешение
\Boot.ini, \Ntdetect.com, \Ntldr	**Administrators**	Full Control
	SYSTEM	Full Control
\Autoexec.bat, \Config.sys	**Administrators**	Full Control
	SYSTEM	Full Control
	Любому пользователю	Read
\TEMP directory	**Administrators**	Full Control
	SYSTEM	Full Control
	CREATOR OWNER	Full Control
	Users	Special Directory Access – Read, Write and Execute, Special File Access – None

4. ***Защита реестра***

Системный реестр (registry) Windows NT – это база данных, содержащая информацию о конфигурации и значениях параметров всех компонентов системы (устройствах, операционной системе и приложениях). Основные кусты реестра находятся в ветви HKEY_LOCAL_MACHINE и называются SAM, SECURITY, SOFTWARE и SYSTEM. Куст SAM – это база данных Менеджера учетных записей. SECURITY хранит информацию, используемую локальным Менеджером безопасности (LSA). В кусте SOFTWARE находятся параметры и настройки программного обеспечения, а в SYSTEM содержатся данные о конфигурации, необходимые для загрузки операционной системы (драйверы, устройства и службы).

Доступ пользователей к полям реестра следует разграничить. Это можно осуществить с помощью утилиты Regedt32 (рис. 4.3).

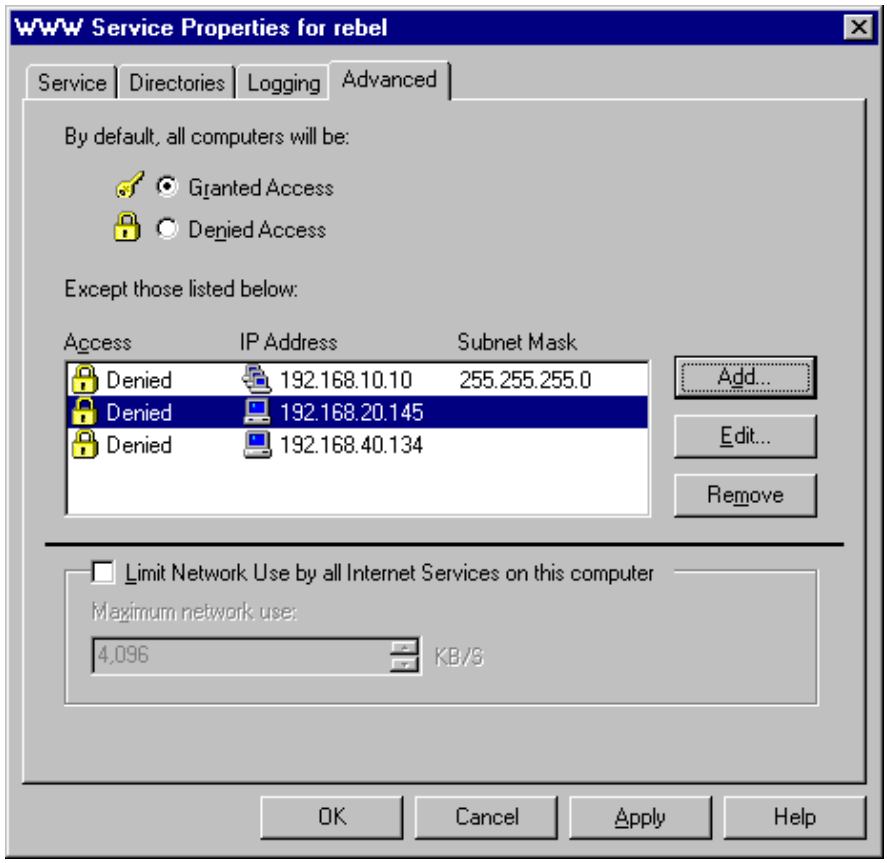


Рис. 4.3

Установленные в системе по умолчанию разрешения на доступ к разделам реестра нельзя модифицировать рядовым пользователям. Поскольку некоторые разделы реестра доступны членам группы Everyone, после установки Windows NT необходимо изменить разрешения в разделе (табл. 5).

Таблица 5

Раздел	Объект защиты
HKEY_LOCAL_MACHINE	\Software
	\Software\Microsoft\RPC (и подразделы)
	\Software\Microsoft\Windows NT\ CurrentVersion
	\Software\Microsoft\Windows NT\ CurrentVersion\Profile List
	\Software\Microsoft\Windows NT\ CurrentVersion\AeDebug

Продолжение табл. 5

Раздел	Объект защиты
	\Software\Microsoft\Windows NT\ CurrentVersion\Compatibility
	\Software\Microsoft\Windows NT\ CurrentVersion\Drivers
	\Software\Microsoft\Windows NT\ CurrentVersion\Embedding
	\Software\Microsoft\Windows NT\ CurrentVersion\Fonts
	\Software\Microsoft\Windows NT\ CurrentVersion\FontSubstitutes
	\Software\Microsoft\Windows NT\ CurrentVersion\Font Drivers
	\Software\Microsoft\Windows NT\ CurrentVersion\Font Mapper
	\Software\Microsoft\Windows NT\ CurrentVersion\Font Cache
	\Software\Microsoft\Windows NT\ CurrentVersion\GRE_Initialize
	\Software\Microsoft\Windows NT\ CurrentVersion\MCI
	\Software\Microsoft\Windows NT\ CurrentVersion\MCI Extensions
	\Software\Microsoft\Windows NT\ CurrentVersion\Port (и подразделы)
	\Software\Microsoft\Windows NT\ CurrentVersion\Type1 Installer
	\Software\Microsoft\Windows NT\ CurrentVersion\WOW (и подразделы)
	\Software\Microsoft\Windows NT\ CurrentVersion\Windows3.1MigrationStatus (и подразделы)
	\System\CurrentControlSet\Services\LanmanServer\Shares
	\System\CurrentControlSet\Services\UPS
	\Software\Microsoft\Windows\CurrentVersion\Run
	\Software\Microsoft\Windows\CurrentVersion\RunOnce

Окончание табл. 5

Раздел	Объект защиты
	\Software\Microsoft\Windows\CurrentVersion\Uninstall
HKEY_CLASSES_ROOT	\HKEY_CLASSES_ROOT (и подразделы)
HKEY_USERS	\.DEFAULT

Для доступа к разделу HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\PerfLib можно вообще удалить группу Everyone, а вместо нее добавить группу INTERACTIVE с правом Read.

Для ограничения удаленного доступа к системному реестру используется запись в разделе HKEY_LOCAL_MACHINE\System\CurrentcontrolSet\Control\SecurePipeServers\winreg. По умолчанию право удаленного доступа к реестру имеют члены группы Administrators. В Workstation этот раздел отсутствует, и его необходимо создать. Право удаленного доступа к реестру получают только пользователи и группы, указанные в списке прав доступа к указанному разделу. К некоторым разделам реестра необходимо предоставить доступ по сети другим пользователям или группам; для этого эти разделы можно указать в параметрах Machine и Users подраздела HKEY_LOCAL_MACHINE\System\CurrentControlSet\Contro\SecurePipeServers\winreg\AllowedPaths.

5. *Безопасность сервера SMB*

Доступ к файлам и принтерам по сети в операционной системе Windows NT обеспечивает сервер SMB (Server Message Block), называемый просто сервером или LAN Manager сервером. SMB осуществляет проверку подлинности клиента, пытающегося получить доступ к информации по сети. Существует два режима работы системы контроля: проверка на уровне ресурса (Share Level) и проверка на уровне пользователя (User Level). Windows NT не поддерживает доступ на уровне ресурса.

При проверке на уровне пользователя сервер выполняет идентификацию пользователя на основе базы учетных записей. Протокол SMB обеспечивает защиту в начальный момент сеанса, затем все данные пользователя передаются по сети в открытом

виде. Если вы хотите обеспечить конфиденциальность информации, необходимо использовать программные или аппаратные средства шифрования транспортного канала (например PPTP, входящего в Windows NT).

Сеансы протокола SMB можно подделать или перехватить. Шлюз может перехватить сеанс SMB и получить такой же доступ к файловой системе, как и легальный пользователь, инициирующий сеанс. Но шлюзы редко используются в локальных сетях.

Возможность передачи по сети пароля пользователя в открытом виде делает систему уязвимой. После установки Service Pack 3 в операционной системе автоматически отключает возможность передачи пароля в открытом виде, но существуют SMB-серверы, не принимающие шифрованный пароль (например, Lan Manager для UNIX). Чтобы включить передачу «открытого» пароля, необходимо установить в реестре в разделе HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanManServer\Parametrs

Параметр	EnablePlainTextPassword
Тип	REG_DWORD
Значение	1

Следует отметить, что корпорация Microsoft модифицировала протокол SMB, который назван SMB Signing. При этом клиент и сервер проверяют подлинность каждого сообщения, поступающего по протоколу SMB. Для этого в каждое сообщение SMB помещается электронная подпись, удостоверяющая знание пароля пользователя клиентом или сервером, пославшим это сообщение. Таким образом, электронная подпись удостоверяет, что команда SMB, во-первых, создана стороной, владеющей паролем пользователя; во-вторых, создана в рамках именно этого сеанса; и, в-третьих, сообщение, передаваемое между сервером и клиентом, – подлинник.

Для включения проверки электронных подписей в сообщения SMB необходимо установить Service Pack 3 и произвести установку параметров в реестре сервера и клиента, для сервера – в разделе HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanManServer\Parametrs

Параметр	EnableSecuritySignature
Тип	REG_DWORD
Значение	1

Если значение равно 0 (по умолчанию), то поддержка SMB Signing на сервере выключена. В отличие от сервера у клиента значение EnableSecuritySignature по умолчанию уже равно 1.

При инициализации сервера образуются папки административного назначения (Administrative shares), которые обеспечивают доступ к корневому каталогу тома. Доступ к этим ресурсам по умолчанию разрешен только членам групп Administrators, Backup Operators, Server Operators и Power Users. Если вы хотите отменить доступ к ним, то необходимо в реестре в разделе HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanManServer\Parametrs

Параметр	AutoShareServer
Тип	REG_DWORD
Установить значение	0

или, используя утилиту System Policy Editor, снять флажки с параметров Create Hidden Drive Shares в разделе Windows NT Network\Sharing (рис. 4.4).

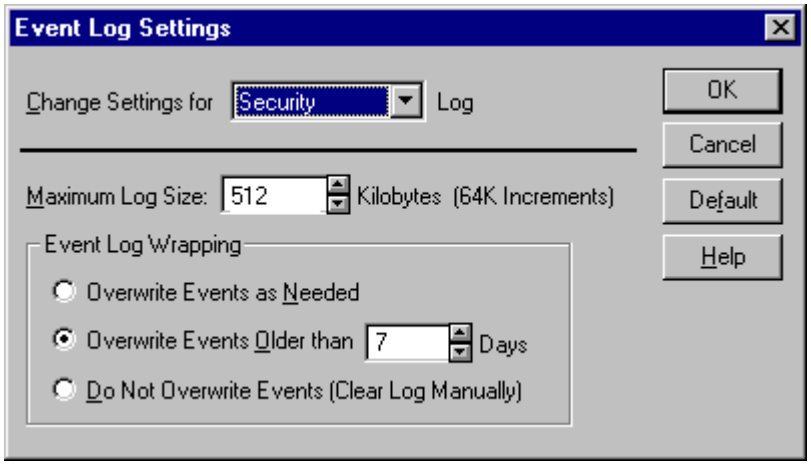


Рис. 4.4

Необходимо ограничить права анонимного пользователя. Инсталляция Service Pack 3 закрывает доступ к реестру системы для анонимного пользователя.

6. *Безопасность сервера IIS*

Microsoft Internet Information Server (IIS) был создан для унификации работы всех служб Internet. Он представляет собой высокоинтегрированный пакет серверных служб поддержки HTTP, FTP и Gopher.

Защита IIS основана на средствах обеспечения безопасности Windows NT. В их число входят:

- *учетные записи пользователей*. Для предотвращения несанкционированного доступа к узлу IIS следует контролировать учетные записи пользователей. К основным методам защиты также относятся: применение формуляра «Гость из Internet», регистрация по имени и паролю пользователя (по схеме аутентификации Windows NT) и выбор сложных для угадывания паролей;
- *установка NTFS*;
- *права доступа*. Основным механизмом доступа через сервер IIS является анонимный доступ. Из механизмов проверки подлинности лишь Windows NT Challenge-Response, используемый сервером HTTP, можно считать относительно защищенным. Поэтому не применяйте для аутентификации базовую схему, так как имя пользователя и пароль при этом передаются по сети открытым способом;
- *уменьшение числа протоколов и отключение службы Server*. Уменьшив число протоколов, которыми пользуются сетевые адаптеры, вы заметно усилите защиту. Чтобы пользователи не смогли просматривать разделяемые ресурсы IIS, отключите службу Server. Отключение этой службы затруднит злоумышленникам поиск слабых мест в вашей системе;
- *защита информации в FTP*. FTP всегда использует защиту на уровне пользователя. Это значит, что для доступа к серверу FTP пользователь должен пройти процедуру регистрации. Сервис FTP сервера IIS для идентификации пользователей, желающих получить доступ, может использовать базу данных пользовательских бюджетов Windows NT Server. Однако при этой процедуре FTP передает всю информацию только открытым текстом, что создает опасность перехвата пользовательских имен и паролей.

Проблема раскрытия паролей устраняется при таких конфигурациях сервера FTP, когда он разрешает анонимный доступ. При анонимном входе пользователь должен ввести в качестве

пользовательского имени **anonymous** и свой почтовый (e-mail) адрес – в качестве пароля. Анонимные пользователи получают доступ к тем же файлам, доступ к которым разрешен бюджету *lVSR_computemame.*

Кроме того, к сервису FTP сервера IIS Windows NT можно разрешить исключительно анонимный доступ. Такой вариант хорош тем, что при нем отсутствует возможность рассекречивания паролей в общей сети. Анонимный доступ к FTP разрешен по умолчанию;

- *контроль доступа по IP-адресу*. Существует дополнительная возможность контроля доступа к серверу IIS – разрешение или запрещение доступа с конкретных IP-адресов (рис. 4.5). Например, можно запретить доступ к своему серверу с определенного IP-адреса; точно так же можно сделать сервер недоступным для целых сетей. С другой стороны, можно разрешить доступ к серверу только определенным узлам;

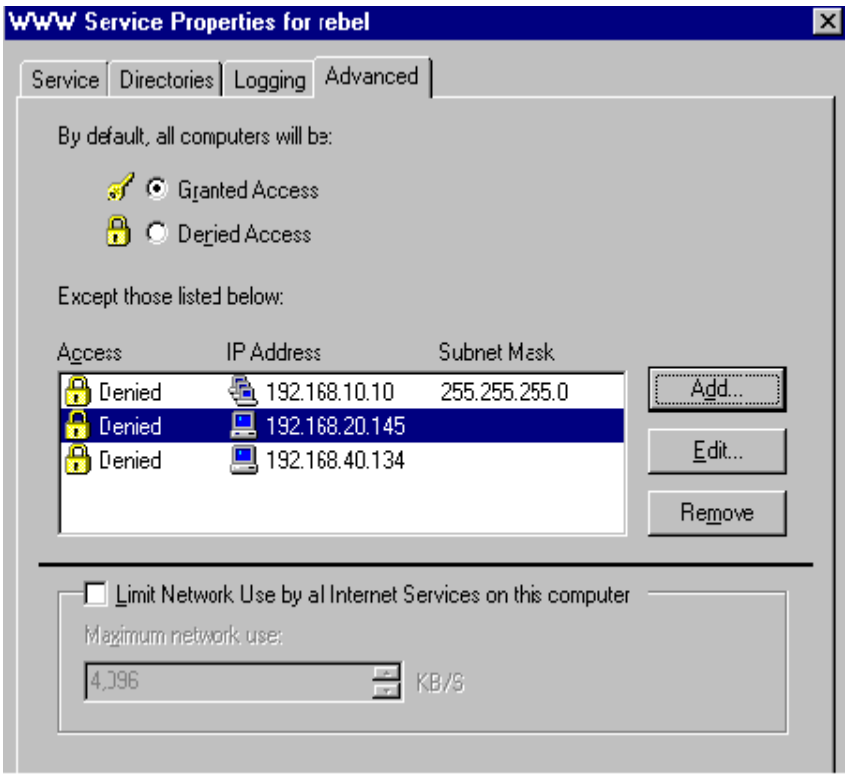


Рис. 4.5

- *схемы шифрования*. Чтобы обеспечить безопасность пакетов во время их пересылки по сети, приходится применять различные схемы шифрования. Необходимость в такой защите вы-

звана тем, что при пересылке пакетов по сети не исключен перехват кадров. Большинство схем шифрования работает внутри прикладного и транспортного уровня модели OSI. Некоторые схемы могут работать и на более низких уровнях. Используются такие протоколы, как: SSL, PCT, SET, PPTP, PGP.

7. *Аудит*

Аудит – одно из средств защиты сети Windows NT. С его помощью можно отслеживать действия пользователей и ряд системных событий в сети. Фиксируются следующие параметры, касающиеся действий, совершаемых пользователями:

- выполненное действие;
- имя пользователя, выполнившего действие;
- дата и время выполнения.

Аудит, реализованный на одном контроллере домена, распространяется на все контроллеры домена. Настройка аудита позволяет выбрать типы событий, подлежащих регистрации, и определить, какие именно параметры будут регистрироваться.

В сетях с минимальными требованиями к безопасности подвергайте аудиту:

- успешное использование ресурсов, только в том случае, если эта информация вам необходима для планирования;
- успешное использование важной и конфиденциальной информации.

В сетях со средними требованиями к безопасности подвергайте аудиту:

- успешное использование важных ресурсов;
- удачные и неудачные попытки изменения стратегии безопасности и административной политики;
- успешное использование важной и конфиденциальной информации.

В сетях с высокими требованиями к безопасности подвергайте аудиту:

- удачные и неудачные попытки регистрации пользователей;
- удачное и неудачное использование любых ресурсов;
- удачные и неудачные попытки изменения стратегии безопасности и административной политики.

Аудит приводит к дополнительной нагрузке на систему, поэтому необходимо регистрировать лишь события, действительно представляющие интерес.

Windows NT записывает события в три журнала:

- *Системный журнал* (system log) содержит сообщения об ошибках, предупреждения и другую информацию, исходящую от операционной системы и компонентов сторонних производителей. Список событий, регистрируемых в этом журнале, предопределен операционной системой и компонентами сторонних производителей и не может быть изменен пользователем. Журнал находится в файле Sysevent.evt.
- *Журнал безопасности* (Security Log) содержит информацию об успешных и неудачных попытках выполнения действий, регистрируемых средствами аудита. События, регистрируемые в этом журнале, определяются заданной вами стратегией аудита. Журнал находится в файле Secevent.evt.
- *Журнал приложений* (Application Log) содержит сообщения об ошибках, предупреждения и другую информацию, выдаваемую различными приложениями. Список событий, регистрируемых в этом журнале, определяется разработчиками приложений. Журнал находится в файле Appevent.evt.

Все журналы размещены в папке %Systemroot%\System32\Config.

При выборе событий для проведения аудита следует учитывать возможность переполнения журнала. Для настройки журнала используйте диалоговое окно Event Log Settings (рис. 4.6).

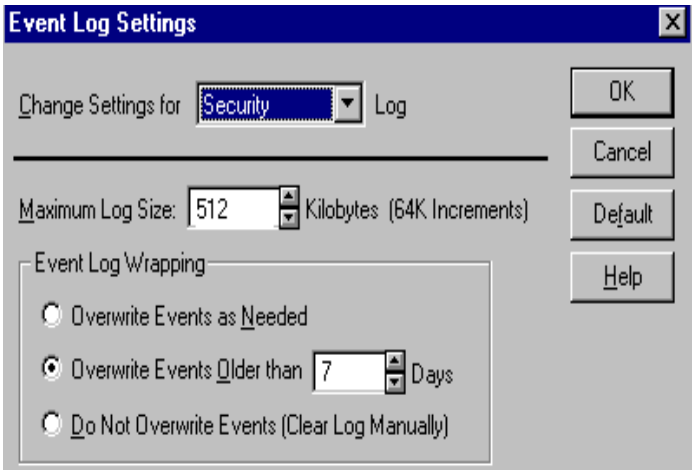


Рис. 4.6

С помощью этого окна можно управлять:

- размером архивируемых журналов (размер по умолчанию – 512 Кбайт, можно изменить размер от 64 до 4 194 240 Кбайт);
- методикой замещения устаревших записей журнала;
- Overwrite Events as Need – в случае заполнения журнала при записи новых событий операционная система удаляет самые старые события;
- Overwrite Events Older then X Days – в случае заполнения журнала при записи новых событий удаляются сами события, но только если они старше X дней, иначе новые события будут проигнорированы;
- Do not Overwrite Events – в случае заполнения журнала новые события не фиксируются. Очистка журнала производится вручную.

Первый этап планирования стратегии аудита – выбор подлежащих аудиту событий в диалоговом окне Audit Policy утилиты User Manager for Domains (User Manager) (рис. 4.7).

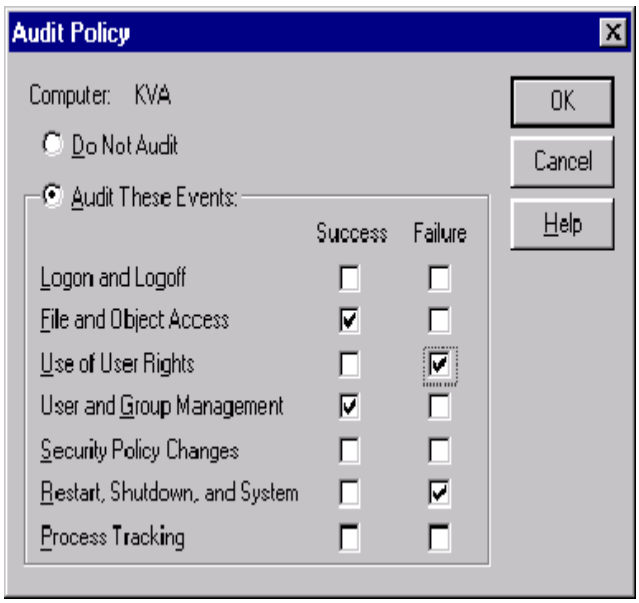


Рис. 4.7

Приведем типы событий, которые могут регистрироваться:

- Logon and Logoff – регистрация пользователя в системе или выход из нее, а также установка и разрыв сетевого соединения;

- File and Object Access – доступ к папкам, файлам и принтерам, подлежащим аудиту;
- Use of User Rights – использование привилегий пользователей (кроме прав, связанных с входом и выходом из системы);
- User and Group Management – создание, изменение и удаление учетных записей пользователей и групп, а также изменения в ограничениях учетной записи;
- Security Policy Changes – изменения в привилегиях пользователей, стратегии аудита и политике доверительных отношений;
- Restart, Shutdown and System – перезапуск или выключение компьютера пользователем; возникновение ситуации, влияющей на безопасность системы;
- Process Tracking – события, которые вызывают запуск и завершение программ.

8. *Службы безопасности Windows NT 5.0*

Система безопасности Windows NT 5.0 позволяет реализовать все новые подходы к проверке подлинности пользователя и защиты данных. В ее состав входит:

- полное интегрирование с активным каталогом Windows NT 5.0 для обеспечения масштабируемого управления учетными записями в больших доменах с гибким контролем доступа и распределением административных полномочий;
- протокол проверки подлинности Kerberos версии 5 – стандарт безопасности для Internet, реализуемый как основной протокол проверки подлинности входа в сеть;
- проверка подлинности с применением сертификатов, основанных на открытых ключах;
- безопасные сетевые каналы, базирующиеся на стандарте SSL;
- файловая система с шифрованием.

Для единообразного обращения к различным протоколам разработан новый интерфейс прикладного программирования Win32 – *интерфейс поставщиков поддержки безопасности* (Security Support Provider Interface, SSPI). SSPI позволяет изолировать проверку подлинности пользователя, которая может осуществляться по разным протоколам, – от применяющих ее служб

и приложений. Интерфейс SSPI представляет собой несколько наборов доступных прикладным программам процедур, выполняющих:

- *управление мандатами* (Credential Management) – работу с информацией о клиенте (пароль, билет и т.д.);
- *управление контекстом* (Context Management) – создание контекста безопасности клиента;
- *поддержку передачи сообщений* (Message Support) – проверку целостности переданной информации (работает в рамках контекста безопасности клиента);
- *управление пакетами* (Package Management) – выбор протокола безопасности.
- Протокол проверки подлинности Kerberos определяет взаимодействие между клиентами и службой проверки подлинности *Центром распределения ключей* (Key Distribution Center, KDC).

В Windows NT 5.0 появится новое средство защиты информации – *файловая система с шифрованием* (Encrypted File System, EFS), позволяющая хранить файлы и папки в зашифрованном виде.