Лекция 8. Модели безопасности основных операционных систем

Цель лекции: курса.

План лекции:
Введение.
1. Механизмы защиты операционных систем
2. Анализ защищенности современных операционных систем
3. Система безопасности операционной системы Windows
4. Защита в операционной системе Unix
5. Защита в операционной системе Novell NetWare
Заключение
Контрольные вопросы

Ключевые слова: [выбрать самостоятельно].

Содержание лекции:

Введение

++++++++++++++++++++++++++

1. Механизмы защиты операционных систем

Операционная система есть специально организованная совокупность программ, которая управляет ресурсами системы (ЭВМ, вычислительной системы, других компонентов ИВС) с целью наиболее эффективного их использования и обеспечивает интерфейс пользователя с ресурсами.

Операционные системы, подобно аппаратуре ЭВМ, на пути своего развития прошли несколько поколений.

ОС первого поколения были направлены на ускорение и упрощение перехода с одной задачи пользователя на другую задачу (другого пользователя), что поставило проблему обеспечения безопасности данных, принадлежащих разным задачам.

Второе поколение ОС характеризовалось наращиванием программных средств обеспечения операций ввода-вывода и стандартизацией обработки прерываний. Надежное обеспечение безопасности данных в целом осталось нерешенной проблемой.

К концу 60-х гг. XX в. начал осуществляться переход к мультипроцессорной организации средств ВТ, поэтому проблемы распределения ресурсов и их защиты стали более острыми и трудноразрешимыми. Решение этих проблем привело к соответствующей организации ОС и широкому применению аппаратных средств защиты (защита памяти, аппаратный контроль, диагностика и т.п.).

Основной тенденцией развития вычислительной техники была и остается идея максимальной доступности ее для пользователей, что входит в противоречие с требованием обеспечения безопасности данных.

Под механизмами защиты ОС будем понимать все средства и механизмы защиты данных, функционирующие в составе ОС. Операционные системы, в составе которых функционируют средства и механизмы защиты данных, часто называют защищенными системами.

Под безопасностью ОС будем понимать такое состояние ОС, при котором невозможно случайное или преднамеренное нарушение функционирования ОС, а также нарушение безопасности находящихся под управлением ОС ресурсов системы. Укажем следующие особенности ОС, которые позволяют выделить вопросы обеспечения безопасности ОС в особую категорию:

- *управление всеми ресурсами системы;*
- *наличие встроенных механизмов, которые прямо или косвенно влияют на безопасность программ и данных, работающих в среде ОС;*
- *обеспечение интерфейса пользователя с ресурсами системы;*
- *размеры и сложность ОС.*

Большинство ОС обладают дефектами с точки зрения обеспечения безопасности данных в системе, что обусловлено выполнением задачи обеспечения максимальной доступности системы для пользователя.

Рассмотрим типовые функциональные дефекты ОС, которые могут привести к созданию каналов утечки данных.

1. *Идентификация. Каждому ресурсу в системе должно быть присвоено уникальное имя - идентификатор. Во многих системах пользователи не имеют возможности удостовериться в том, что используемые ими ресурсы действительно принадлежат системе.*
2. *Пароли. Большинство пользователей выбирают простейшие пароли, которые легко подобрать или угадать.*
3. *Список паролей. Хранение списка паролей в незашифрованном виде дает возможность его компрометации с последующим НСД к данным.*
4. *Пороговые значения. Для предотвращения попыток несанкционированного входа в систему с помощью подбора пароля необходимо ограничить число таких попыток, что в некоторых ОС не предусмотрено.*
5. *Подразумеваемое доверие. Во многих случаях программы ОС считают, что другие программы работают правильно.*
6. *Общая память. При использовании общей памяти не всегда после выполнения программ очищаются участки оперативной памяти (ОП).*
7. *Разрыв связи. В случае разрыва связи ОС должна немедленно закончить сеанс работы с пользователем или повторно установить подлинность субъекта.*
8. *Передача параметров по ссылке, а не по значению (при передаче параметров по ссылке возможно сохранение параметров в ОП после проверки их корректности, нарушитель может изменить эти данные до их использования).*
9. *Система может содержать много элементов (например, программ), имеющих различные привилегии.*

Основной проблемой обеспечения безопасности ОС является проблема создания механизмов контроля доступа к ресурсам системы. Процедура контроля доступа заключается в проверке соответствия запроса субъекта предоставленным ему правам доступа к ресурсам. Кроме того, ОС содержит вспомогательные средства защиты, такие как средства мониторинга, профилактического контроля и аудита. В совокупности механизмы контроля доступа и вспомогательные средства защиты образуют механизмы управления доступом.

Средства профилактического контроля необходимы для отстранения пользователя от непосредственного выполнения критичных с точки зрения безопасности данных операций и передачи этих операций под контроль ОС. Для обеспечения безопасности данных работа с ресурсами системы осуществляется с помощью специальных программ ОС, доступ к которым ограничен.

Средства мониторинга осуществляют постоянное ведение регистрационного журнала, в который заносятся записи о всех событиях в системе. В ОС могут использоваться средства сигнализации о НСД, которые используются при обнаружении нарушения безопасности данных или попыток нарушения.

Контроль доступа к данным. При создании механизмов контроля доступа необходимо, прежде всего, определить множества субъектов и объектов доступа. Субъектами могут быть, например, пользователи, задания, процессы и процедуры. Объектами - файлы, программы, семафоры, директории, терминалы, каналы связи,

устройства, блоки ОП и т.д. Субъекты могут одновременно рассматриваться и как объекты, поэтому у субъекта могут быть права на доступ к другому субъекту. В конкретном процессе в данный момент времени субъекты являются активными элементами, а объекты - пассивными.

Для осуществления доступа к объекту субъект должен обладать соответствующими полномочиями. Полномочие есть некий символ, обладание которым дает субъекту определенные права доступа по отношению к объекту, область защиты определяет права доступа некоторого субъекта ко множеству защищаемых объектов и представляет собой совокупность всех полномочий данного субъекта.

При функционировании системы необходимо иметь возможность создавать новые субъекты и объекты. При создании объекта одновременно создается и полномочие субъектов по использованию этого объекта. Субъект, создавший такое полномочие, может воспользоваться им для осуществления доступа к объекту или же может создать несколько копий полномочия для передачи их другим субъектам.

С традиционной точки зрения средства управления доступом позволяют специфицировать и контролировать действия, которые субъекты (пользователи и процессы) могут выполнять над объектами (информацией и другими компьютерными ресурсами). В данном разделе речь пойдет о логическом управлении доступом, которое, в отличие от физического, реализуется программными средствами. Логическое управление доступом - это основной механизм многопользовательских систем, призванный обеспечить конфиденциальность и целостность объектов и, до некоторой степени, их доступность (путем запрещения обслуживания неавторизованных пользователей).

Рассмотрим формальную постановку задачи в традиционной трактовке. Имеется совокупность субъектов и набор объектов. Задача логического управления доступом состоит в том, чтобы для каждой пары "субъект-объект" определить множество допустимых операций и контролировать выполнение установленного порядка.

Отношение "субъекты-объекты" можно представить в виде матрицы доступа, в строках которой перечислены субъекты, в столбцах - объекты, а в клетках, расположенных на пересечении строк и столбцов, записаны дополнительные условия (например, время и место действия) и разрешенные виды доступа. Фрагмент матрицы может выглядеть, например, как показано в табл. 1.

Таблица 1. Фрагмент матрицы доступа

	Файл	Программа	Линия связи	Реляционная таблица
Пользователь 1	o r w	c системной консоли	e	rw с 8:00 до 18:00
Пользователь 2			a	

Обозначение: "o" - разрешение на передачу прав доступа другим пользователям, "r" - чтение, "w" - запись, "e" - выполнение, "a" - добавление информации.

Тема логического управления доступом - одна из сложнейших в области информационной безопасности. Дело в том, что само понятие объекта (а тем более видов доступа) меняется от сервиса к сервису. Для операционной системы к объектам относятся файлы, устройства и процессы. Применительно к файлам и устройствам обычно рассматриваются права на чтение, запись, выполнение (для программных файлов), иногда на удаление и добавление. Отдельным правом может быть возможность передачи полномочий доступа другим субъектам (так называемое право владения). Процессы можно создавать и уничтожать. Современные операционные системы могут поддерживать и другие объекты.

Для систем управления реляционными базами данных объект - это база данных, таблица, представление, хранимая процедура. К таблицам применимы операции поиска, добавления, модификации и удаления данных, у других объектов. В результате при задании матрицы доступа нужно принимать во внимание не только принцип

распределения привилегий для каждого сервиса, но и существующие связи между сервисами (приходится заботиться о согласованности разных частей матрицы). Аналогичная трудность возникает при экспорте/импорте данных, когда информация о правах доступа, как правило, теряется (поскольку на новом сервисе она не имеет смысла). Следовательно, обмен данными между различными сервисами представляет особую опасность с точки зрения управления доступом, а при проектировании и реализации разнородной кон

фигурации необходимо позаботиться о согласованном распределении прав доступа субъектов к объектам и о минимизации числа способов экспорта/импорта данных.
Матрицу доступа, ввиду ее разреженности (большинство клеток - пустые), неразумно хранить в виде двухмерного массива. Обычно ее хранят по столбцам, т.е. для каждого объекта поддерживается список "допущенных" субъектов вместе с их правами. Элементами списков могут быть имена групп и шаблоны субъектов, что служит большим подспорьем администратору. Некоторые проблемы возникают только при удалении субъекта, когда приходится удалять его имя из всех списков доступа; впрочем, эта операция производится нечасто.
Списки доступа - исключительно гибкое средство. С их помощью легко выполнить требование о гранулярности прав с точностью до пользователя. Посредством списков несложно добавить права или явным образом запретить доступ (например, чтобы наказать нескольких членов группы пользователей). Безусловно, списки являются лучшим средством произвольного управления доступом.
Подавляющее большинство операционных систем и систем управления базами данных реализуют именно произвольное управление доступом. Основное достоинство произвольного управления - гибкость. К сожалению, у "произвольного" подхода есть ряд недостатков. Рассредоточенность управления доступом ведет к тому, что доверенными должны быть многие пользователи, а не только системные операторы или администраторы. Из-за рассеянности или некомпетентности сотрудника, владеющего секретной информацией, эту информацию могут узнать и все остальные пользователи. Следовательно, произвольность управления должна быть дополнена жестким контролем за реализацией избранной политики безопасности.
Второй недостаток, который представляется основным, состоит в том, что права доступа существуют отдельно от данных. Ничто не мешает пользователю, имеющему доступ к секретной информации, записать ее в доступный всем файл или заменить полезную утилиту ее "троянским" аналогом. Подобная "разделенность" прав и данных существенно осложняет проведение несколькими системами согласованной политики безопасности и, главное, делает практически невозможным эффективный контроль согласованности.
Возвращаясь к вопросу представления матрицы доступа, укажем, что для этого можно использовать также функциональный способ, когда матрицу не хранят в явном виде, а каждый раз вычисляют содержимое соответствующих клеток. Например, при принудительном управлении доступом применяется сравнение меток безопасности субъекта и объекта.
Удобной надстройкой над средствами логического управления доступом является ограничивающий интерфейс, когда пользователя лишают самой возможности попытаться совершить несанкционированные действия, включив в число видимых ему объектов только те, к которым он имеет доступ. Подобный подход обычно реализуют в рамках системы меню (пользователю показывают лишь допустимые варианты выбора) или посредством ограничивающих оболочек, таких как restricted shell в ОС Unix.

Рис.

Рис. 9.1. Схема модели Харрисона, Руззо и Ульмана

При принятии решения о предоставлении доступа обычно анализируется следующая информация:

1) идентификатор субъекта (идентификатор пользователя, сетевой адрес компьютера и т.п.). Подобные идентификаторы являются основой произвольного (или дискреционного) управления доступом;

2) атрибуты субъекта (метка безопасности, группа пользователя и т.п.). Метки безопасности - основа мандатного управления доступом.

Непосредственное управление правами доступа осуществляется на основе одной из моделей доступа:

- матричной модели доступа (модель Харрисона-Руззо-Ульмана);
- многоуровневой модели доступа (модель Белла- Лападулы).

Разработка и практическая реализация различных защищенных ОС привела Харрисона, Руззо и Ульмана к построению формальной модели защищенных систем. Схема модели Харрисона, Руззо и Ульмана (HRU-модели) приведена на рис. 9.1.

9.2. АНАЛИЗ ЗАЩИЩЕННОСТИ СОВРЕМЕННЫХ ОПЕРАЦИОННЫХ СИСТЕМ

9.2.1. АНАЛИЗ ВЫПОЛНЕНИЯ СОВРЕМЕННЫМИ ОС ФОРМАЛИЗОВАННЫХ ТРЕБОВАНИЙ К ЗАЩИТЕ ИНФОРМАЦИИ ОТ НСД

Анализировать выполнение современными универсальными ОС требований, задаваемых для класса защищенности АС 1В, не имеет смысла в принципе. Для большинства ОС либо полностью не реализуется основной для данных приложений мандатный механизм управления доступом к ресурсам, либо не выполняется его важнейшее требование "Должно осуществляться управление потоками информации с помощью меток конфиденциальности. При этом уровень конфиденциальности накопителя должен быть не ниже уровня конфиденциальности записываемой на него информации". В связи с этим далее будем говорить лишь о возможном соответствии средств защиты современных ОС классу АС 1Г (защита конфиденциальной информации).

В качестве альтернативных реализаций ОС рассмотрим семейства Unix и Windows (естественно, Windows NT/2000, так как о встроенных механизмах защиты ОС Windows 9x/Me говорить вообще не приходится).

Сначала остановимся на принципиальном, даже, можно сказать, концептуальном противоречии между реализованными в ОС механизмами защиты и принятыми формализованными требованиями. Концептуальном в том смысле, что это противоречие характеризует не какой-либо один механизм защиты, а общий подход к построению системы защиты.

Противоречие состоит в принципиальном различии подходов (соответственно требований) к построению схемы администрирования механизмов защиты и, как следствие, это коренным образом сказывается на формировании общих принципов задания и реализации политики безопасности в организации, распределения ответственности за защиту информации, а также на определении того, кого относить к потенциальным злоумышленникам (от кого защищать информацию).

Для иллюстрации из совокупности формализованных требований к системе защиты конфиденциальной информации рассмотрим следующие два требования:

1) право изменять правила разграничения доступа (ПРД) должно предоставляться выделенным субъектам (администрации, службе безопасности и т.д.);

2) должны быть предусмотрены средства управления, ограничивающие распространения прав на доступ.

Данные требования жестко регламентируют схему (или модель) администрирования механизмов защиты. Это должна быть централизованная схема, единственным элементом которой выступает выделенный субъект, в частности, администратор (администратор

безопасности). При этом конечный пользователь исключен в принципе из схемы администрирования механизмов защиты.

При реализации концепции построения системы защиты, регламентируемой рассматриваемыми требованиями, пользователь не наделяется элементом доверия, так как он может считаться потенциальным злоумышленником, что и имеет место на практике.

Теперь в общих чертах рассмотрим концепцию, реализуемую в современных универсальных ОС. Здесь "владельцем" файлового объекта, т.е. лицом, получающим право на задание атрибутов (или ПРД) доступа к файловому объекту, является лицо, создающее файловый объект. Так как файловые объекты создают конечные пользователи, то именно они и назначают ПРД к создаваемым им файловым объектам. Другими словами, в ОС реализуется распределенная схема назначения ПРД, где элементами схемы администрирования являются собственно конечные пользователи.

В данной схеме пользователь должен наделяться практически таким же доверием, как и администратор безопасности, при этом нести наряду с ним ответственность за обеспечение компьютерной безопасности. Отметим, что данная концепция реализуется и большинством современных приложений, в частности СУБД, где пользователь может распространять свои права на доступ к защищаемым ресурсам. Кроме того, не имея в полном объеме механизмов защиты компьютерной информации от конечного пользователя, в рамках данной концепции невозможно рассматривать пользователя в качестве потенциального злоумышленника. А как мы увидим далее, именно с несанкционированными действиями пользователя на защищаемом компьютере (причем как сознательными, так и нет) связана большая часть угроз компьютерной безопасности.

Отметим, что централизованная и распределенная схемы администрирования - это две диаметрально противоположные точки зрения на защиту, требующие совершенно различных подходов к построению моделей и механизмов защиты. При этом сколько-нибудь гарантированную защиту информации можно реализовать только при принятии концепции полностью централизованной схемы администрирования, что подтверждается известными угрозами ОС.

Возможности моделей, методов и средств защиты будем рассматривать применительно к реализации именно концепции централизованного администрирования. Одним из элементов данной концепции является рассмотрение пользователя в качестве потенциального злоумышленника, способного осуществить НСД к защищаемой информации.

9.2.2. ОСНОВНЫЕ ВСТРОЕННЫЕ МЕХАНИЗМЫ ЗАЩИТЫ ОС И ИХ НЕДОСТАТКИ

Кратко остановимся на основных механизмах защиты, встроенных в современные универсальные ОС. Сделаем это применительно к возможности реализации ими принятой нами для рассмотрения концепции защиты конфиденциальной информации.

9.2.2.1. ОСНОВНЫЕ ЗАЩИТНЫЕ МЕХАНИЗМЫ ОС СЕМЕЙСТВА UNIX

Защита ОС семейства Unix в общем случае базируется на трех основных механизмах:

1) идентификации и аутентификация пользователя при входе в систему;
2) разграничении прав доступа к файловой системе, в основе которого лежит реализация дискреционной модели доступа;
3) аудит, т. е. регистрация событий.

При этом отметим, что для различных клонов ОС семейства Unix возможности механизмов защиты могут незначительно различаться, однако будем рассматривать ОС Unix в общем случае, без учета некоторых незначительных особенностей отдельных ОС этого семейства.

Построение файловой системы и разграничение доступа к файловым объектам имеет особенности, присущие данному семейству ОС. Рассмотрим кратко эти особенности. Все дисковые накопители (тома) объединяются в единую виртуальную файловую систему путем операции монтирования тома. При этом содержимое тома проецируется на

выбранный каталог файловой системы. Элементами файловой системы являются также все устройства, подключаемые к защищаемому компьютеру (монтируемые к файловой системе). Поэтому разграничение доступа к ним осуществляется через файловою систему.
Каждый файловый объект имеет индексный дескриптор, в котором среди прочего хранится информация о разграничении доступа к данному файловому объекту. Права доступа делятся на три категории: доступ для владельца, доступ для группы и доступ для остальных пользователей. В каждой категории определяются права на чтение, запись и исполнение (в случае каталога - просмотр).
Пользователь имеет уникальные символьный идентификатор (имя) и числовой идентификатор (UID). Символьный идентификатор предъявляется пользователем при входе в систему, числовой используется операционной системой для определения прав пользователя в системе (доступ к файлам и т.д.).
Принципиальные недостатки защитных механизмов ОС семейства Unix. Рассмотрим в общем случае недостатки реализации системы защиты ОС семейства Unix в части невыполнения требований к защите конфиденциальной информации, напрямую связанные с возможностью НСД к информации.
Для начала отметим, что в ОС семейства Unix, вследствие реализуемой ею концепции администрирования (не централизованная), невозможно обеспечить замкнутость (или целостность) программной среды. Это связано с невозможностью установки атрибута "исполнение" на каталог (для каталога данный атрибут ограничивает возможность "обзора" содержимого каталога). Поэтому при разграничении администратором доступа пользователей к каталогам, пользователь, как "владелец" создаваемого им файла, может занести в свой каталог исполняемый файл и, как его "владелец", установить на файл атрибут "исполнение", после чего запустить записанную им программу. Эта проблема непосредственно связана с реализуемой в ОС концепцией защиты информации.
Не в полном объеме реализуется дискреционная модель доступа, в частности, не могут разграничиваться права доступа для пользователя "root" (UID = 0), т.е. данный субъект доступа исключается из схемы управления доступом к ресурсам. Соответственно все запускаемые им процессы имеют неограниченный доступ к защищаемым ресурсам. С этим недостатком системы защиты связано множество атак, в частности:

- несанкционированное получение прав root;
- запуск с правами root собственного исполняемого файла (локально либо удаленно внедренного), при этом несанкционированная программа получает полный доступ к защищаемым ресурсам и т.д.

Кроме того, в ОС семейства Unix невозможно встроенными средствами гарантированно удалять остаточную информацию. Для этого в системе абсолютно отсутствуют соответствующие механизмы.
Необходимо также отметить, что большинство ОС данного семейства не обладают возможностью контроля целостности файловой системы, т.е. не содержат соответствующих встроенных средств. В лучшем случае дополнительными утилитами может быть реализован контроль конфигурационных файлов ОС по расписанию в то время, как важнейшей возможностью данного механизма можно считать контроль целостности программ (приложений) перед их запуском, контроль файлов данных пользователя и т.д.
Что касается регистрации (аудита), то в ОС семейства Unix не обеспечивается регистрация выдачи документов на "твердую копию", а также некоторые другие требования к регистрации событий.
Если же трактовать требования к управлению доступом в общем случае, то при защите компьютера в составе ЛВС необходимо управление доступом к узлам сети. Однако встроенными средствами зашиты некоторых ОС семейства Unix управление доступом к узлам не реализуется.

Из приведенного анализа видно, что многие механизмы, необходимые с точки зрения выполнения формализованных требований, большинством ОС семейства Unix не реализуется в принципе, либо реализуется лишь частично.

9.2.2.2. ОСНОВНЫЕ ЗАЩИТНЫЕ МЕХАНИЗМЫ ОС СЕМЕЙСТВА WINDOWS (NT/2000/XP)

Теперь кратко остановимся на основных механизмах защиты, реализованных в ОС семейства Windows, и проведем анализ защищенности ОС семейства Windows (NT/2000). Отметим, что здесь ряд объектов доступа (в частности, устройства, реестр ОС и т.д.) не являются объектами файловой системы. Поэтому возникает вопрос, как следует трактовать требование "Система защиты должна контролировать доступ наименованных субъектов (пользователей) к наименованным объектам (файлам, программам, томам и т.д.)". Не ясно, являются ли объектами доступа, к которым, следуя формальным требованиям, необходимо разграничивать доступ пользователей, например, реестр ОС и т.д.

В отличие от семейства ОС Unix, где все задачи разграничительной политики доступа к ресурсам решаются средствами управления доступом к объектам файловой системы, доступ в данных ОС разграничивается собственным механизмом для каждого ресурса. Другими словами, при рассмотрении механизмов защиты ОС Windows встает задача определения и задания требований к полноте разграничений (это определяется тем, что считать объектом доступа).

Также, как и для семейства ОС Unix, здесь основными механизмами защиты являются:

1) идентификация и аутентификация пользователя при входе в систему;

2) разграничение прав доступа к ресурсам, в основе которого лежит реализация дискреционной модели доступа (отдельно к объектам файловой системы, к устройствам, к реестру ОС, к принтерам и др.);

3) аудит, т. е. регистрация событий.

Здесь явно выделяются (в лучшую сторону) возможности разграничений прав доступа к файловым объектам (для NTFS) - существенно расширены атрибуты доступа, устанавливаемые на различные иерархические объекты файловой системы (логические диски, каталоги, файлы). В частности, атрибут "исполнение" может устанавливаться и на каталог, тогда он наследуется соответствующими файлами.

При этом существенно ограничены возможности управления доступом к другим защищаемым ресурсам, в частности, к устройствам ввода. Например, здесь отсутствует атрибут "исполнение", т.е. невозможно запретить запуск несанкционированной программы с устройств ввода.

Принципиальные недостатки защитных механизмов ОС семейства Windows (NT/2000/XP). Прежде всего рассмотрим принципиальные недостатки защиты ОС семейства Windows, напрямую связанные с возможностью НСД к информации. При этом в отличие от ОС семейства Unix в ОС Windows невозможна в общем случае реализация централизованной схемы администрирования механизмов защиты или соответствующих формализованных требований. Вспомним, что в ОС Unix это распространялось лишь на запуск процессов. Связано это с тем, что в ОС Windows принята иная концепция реализации разграничительной политики доступа к ресурсам (для NTFS).

В рамках этой концепции разграничения для файла приоритетнее, чем для каталога, а в общем случае - разграничения для включаемого файлового объекта приоритетнее, чем для включающего. Это приводит к тому, что пользователь, создавая файл и являясь его "владельцем", может назначить любые атрибуты доступа к такому файлу (т.е. разрешить к нему доступ любому иному пользователю). Обратиться к этому файлу может пользователь (которому назначил права доступа "владелец") вне зависимости от установленных администратором атрибутов доступа на каталог, в котором пользователь

создает файл. Данная проблема непосредственно связана с реализуемой в ОС Windows концепцией защиты информации.

Далее, в ОС семейства Windows (NT/2000/XP) не в полном объеме реализуется дискреционная модель доступа, в частности, не могут разграничиваться права доступа для пользователя "Система". В ОС присутствуют не только пользовательские, но и системные процессы, которые запускаются непосредственно системой. При этом доступ системных процессов не может быть разграничен. Соответственно, все запускаемые системные процессы имеют неограниченный доступ к защищаемым ресурсам. С этим недостатком системы защиты связано множество атак, в частности, несанкционированный запуск собственного процесса с правами системного. Кстати, это возможно и вследствие некорректной реализации механизма обеспечения замкнутости программной среды.

В ОС семейства Windows (NT/2000/XP) невозможно в общем случае обеспечить замкнутость (или целостность) программной среды. Это связано совершено с иными проблемами, чем в ОС семейства Unix, в которых невозможно установить атрибут "исполнение" на каталог. Для выяснения сложности данного вопроса рассмотрим два способа, которыми в общем случае можно реализовать данный механизм, причем оба способа несостоятельны. Итак, механизм замкнутости программной среды в общем случае может быть обеспечен:

- заданием списка разрешенных к запуску процессов с предоставлением возможности пользователям запускать процессы только из этого списка. При этом процессы задаются полнопутевыми именами, причем средствами разграничения доступа обеспечивается невозможность их модернизации пользователем. Данный подход просто не реализуется встроенными в ОС механизмами;
- разрешением запуска пользователями программ только из заданных каталогов при невозможности модернизации этих каталогов. Одним из условий корректной реализации данного подхода является запрет пользователям запуска программ иначе, чем из соответствующих каталогов. Некорректность реализации ОС Windows данного подхода связана с невозможностью установки атрибута "исполнение" на устройства ввода (дисковод или CD-ROM). В связи с этим при разграничении доступа пользователь может запустить несанкционированную программу с дискеты, либо с диска CD-ROM (очень распространенная атака на ОС данного семейства).

Здесь же стоит отметить, что с точки зрения обеспечения замкнутости программной среды [т.е. реализации механизма, обеспечивающего возможность пользователям запускать только санкционированные процессы (программы)] действия пользователя по запуску процесса могут быть как явными, так и скрытыми.

Явные действия предполагают запуск процессов (исполняемых файлов), которые однозначно идентифицируются своим именем. Скрытые действия позволяют осуществлять встроенные в приложения интерпретаторы команд. Примером таковых могут служить офисные приложения. При этом скрытыми действиями пользователя будет запуск макроса.

В данном случае идентификации подлежит лишь собственно приложение, например, процесс winword.exe. При этом он может помимо своих регламентированных действий выполнять те скрытые действия, которые задаются макросом (соответственно, те, которые допускаются интерпретатором), хранящимся в открываемом документе. То же относится и к любой виртуальной машине, содержащей встроенный интерпретатор команд. При этом отметим, что при использовании приложений, имеющих встроенные интерпретаторы команд (в том числе офисных приложений), не в полном объеме обеспечивается выполнение требования по идентификации программ.

Возвращаясь к обсуждению недостатков, отметим, что в ОС семейства Windows (NT/2000/XP) невозможно встроенными средствами гарантированно удалять остаточную информацию. В системе просто отсутствуют соответствующие механизмы.

Кроме того, ОС семейства Windows (NT/2000/XP) не обладают в полном объеме возможностью контроля целостности файловой системы. Встроенные механизмы системы позволяют контролировать только собственные системные файлы, не обеспечивая контроль целостности файлов пользователя. Кроме того, они не решают важнейшую задачу данных механизмов - контроль целостности программ (приложений) перед их запуском, контроль файлов данных пользователя и др.
Что касается регистрации (аудита), то в ОС семейства Windows (NT/2000/XP) не обеспечивается регистрация выдачи документов на "твердую копию", а также некоторые другие требования к регистрации событий.

Опять же, если трактовать требования к управлению доступом в общем случае, то при защите компьютера в составе ЛВС необходимо управление доступом к узлам сети (распределенный пакетный фильтр). В ОС семейства Windows (NT/2000/XP) механизм управления доступа к узлам в полном объеме не реализуется.
Что касается разделяемых сетевых ресурсов, то фильтрации подвергается только входящий доступ к разделяемому ресурсу, а запрос доступа на компьютере, с которого он осуществляется, фильтрации не подлежит. Это принципиально, так как не могут подлежать фильтрации приложения, которыми пользователь осуществляет доступ к разделяемым ресурсам. Благодаря этому, очень распространенными являются атаки на протокол NETBIOS.
Кроме того, в полном объеме управлять доступом к разделяемым ресурсам возможно только при установленной на всех компьютерах ЛВС файловой системы NTFS. В противном случае невозможно запретить запуск несанкционированной программы с удаленного компьютера, т.е. обеспечить замкнутость программной среды в этой части.
Из приведенного анализа можно видеть, что многие механизмы, необходимые с точки зрения выполнения формализованных требований, ОС семейства Windows не реализуют в принципе, либо реализуют лишь частично.
С учетом сказанного можем сделать важный вывод относительно того, что большинством современных универсальных ОС не выполняются в полном объеме требования к защите АС по классу 1Г. Это значит, что, учитывая требования нормативных документов, они не могут без использования добавочных средств защиты применяться для защиты даже конфиденциальной информации. При этом следует отметить, что основные проблемы защиты здесь вызваны не невыполнимостью ОС требований к отдельным механизмам защиты, а принципиальными причинами, обусловленными реализуемой в ОС концепцией защиты. Концепция эта основана на реализации распределенной схемы администрирования механизмов защиты, что само по себе является невыполнением формализованных требований к основным механизмам защиты.

9.2.3. АНАЛИЗ СУЩЕСТВУЮЩЕЙ СТАТИСТИКИ УГРОЗ ДЛЯ СОВРЕМЕННЫХ УНИВЕРСАЛЬНЫХ ОС.

СЕМЕЙСТВА ОС И ОБЩАЯ СТАТИСТИКА УГРОЗ

На сегодняшний день существует достаточно большая статистика угроз ОС, направленных на преодоление встроенных в ОС механизмов защиты, позволяющих изменить настройки механизмов безопасности, обойти разграничения доступа и т.д. Таким образом, статистика фактов НСД к информации показывает, что большинство распространенных систем (универсального назначения) довольно уязвимы с точки зрения безопасности. И это несмотря на отчетливую тенденцию к повышению уровня защищенности этих систем.
Здесь необходимо отметить, что на практике современные информационные системы, предназначенные для обработки конфиденциальной информации, строятся уже с учетом дополнительных мер безопасности, что также косвенно подтверждает изначальную уязвимость современных ОС.

Рассмотрим операционные системы, фигурирующие в публикуемых списках системных и прикладных ошибок, позволяющих получить несанкционированный доступ к системе, понизить степень ее защищенности или добиться отказа в обслуживании (системного сбоя):

MS Windows 9X BSD AIX BSD Novell Netware
MS Windows NT Solaris SCO HPUX IOS (Cisco)
MS Windows 2000 Sun OS Linux IRIX Digital Unix

Общее количество известных успешных атак для различных ОС, представлено в табл. 9.1, а их процентное соотношение - на диаграмме рис. 9.1.

Вследствие того, что большинство атак для операционных систем, построенных на базе Unix (BSD или AT&T), достаточно похожи, целесообразно объединить их в одну группу. Тоже самое можно сказать и об ОС семейства Windows. Таким образом, далее будем рассматривать только семейства ОС: Unix, MS Windows, Novell NetWare.

9.1. Количество известных успешных атак для различных ОС

Тип ОС	Количество атак	Тип ОС	Количество атак
MS Windows NT/2000	130	Linux	167
MS Windows 9X/ME	120	IRIX	84
BSD	64	HPUX	65
BSDI	10	AIX	42
Solaris	125	SCO	40
Sun OS	40	Novell NetWare	10
Digital Unix	25	IOS (Cisco)	7

180
160
140
120
100
80
60
40
20
0
& * <ь # ^ .
*/•/ °
*
(У3
<е
Л'У
>«v°

Рис.

Рис. 9.1. Статистика соотношения угроз для различных ОС

Общее количество известных успешных атак для различных групп ОС представлено в табл. 9.2, а их процентное соотношение - на диаграмме рис. 9.2.

9.2. Общее количество успешных атак для различных групп ОС

Тип ОС	Количество атак
MS Windows	230
Unix	660
Novell Netware	10

Novell Netware MS Windows 1%

Рис.

Рис. 9.2. Статистика соотношения угроз для семейств ОС

Относительно ОС Novell следует заметить, что данная ОС изначально создавалась как защищенная (не универсального назначения) ОС, основной функцией которой был защищенный файловый сервис. Это, с одной стороны, должно было обеспечить ее более высокий уровень защищенности, с другой стороны, налагало определенные ограничения по использованию. Однако, начиная с пятой версии, данная ОС начала приобретать свойства универсальности (с точки зрения применяемых протоколов и приложений), что в какой-то мере сказалось и на уровне ее защищенности.

9.2.4. ОБЗОР И СТАТИСТИКА МЕТОДОВ, ЛЕЖАЩИХ В ОСНОВЕ АТАК НА СОВРЕМЕННЫЕ ОС. КЛАССИФИКАЦИЯ МЕТОДОВ И ИХ СРАВНИТЕЛЬНАЯ СТАТИСТИКА

Анализируя рассматриваемые атаки, все методы, позволяющие несанкционированно вмешаться в работу системы, можно разделить на следующие группы:

1) позволяющие несанкционированно запустить исполняемый код;
2) позволяющие осуществить несанкционированные операции чтения/записи файловых или других объектов;
3) позволяющие обойти установленные разграничения прав доступа;
4) приводящие к отказу (Denial of Service) в обслуживании (системный сбой);
5) использующие встроенные недокументированные возможности (ошибки и закладки);
6) использующие недостатки системы хранения или выбора (недостаточная длина) данных об аутентификации (пароли) и позволяющие путем реверсирования, подбора или полного перебора всех вариантов получить эти данные;
7) троянские программы;
8) прочие.

Диаграмма, представляющая собой соотношение групп атак (для представленной выше их классификации) для ОС семейства Windows, представлена на рис. 9.3, для ОС семейства Unix - на рис. 9.4.

Из приведенного анализа можно сделать следующий важный вывод: угрозы, описанные в большинстве групп, напря-
мую используют различные недостатки ОС и системных приложений и позволяют при полностью сконфигурированных и
работающих встроенных в ОС механизмах защиты осуществлять НСД, что подтверждает необходимость усиления встроен-
ных механизмов защиты.

35%
30%
25%
20%
15%
10%
5%
0%

Рис.

Рис. 9.3. Соотношение групп атак для ОС семейства Windows
В- $ £
к ю го о
к Р.
9%

27%
6%
5%
5%
4%
45% 40% 35% 30% 25% - 20% - 15% - 10% - 5% 0%
о. го
5 1
2 о. te
ьа
39%
24%
8%
4%
3%

Рис.

Рис. 9.4. Соотношение групп атак для ОС семейства UNIX

Анализируя представленную статистику угроз, можно сделать вывод, что большая их часть связана именно с недостатками средств защиты ОС, отмеченными выше, т.е. недостатками, связанными с невыполнением (полным, либо частичным) формализованных требований к защите, среди которых, в первую очередь, могут быть выделены:

1) некорректная реализация механизма управления доступом, прежде всего, при разграничении доступа к защищаемым объектам системных процессов и пользователей, имеющих права администратора;
2) отсутствие обеспечения замкнутости (целостности) программной среды.

Как видно, большинство атак осуществлялось либо с использованием некоторых прикладных программ, либо с применением встроенных в виртуальные машины средств программирования, т.е. возможность большинства атак напрямую связана с возможностью запуска злоумышленником соответствующей программы. При этом запуск может быть осуществлен как явно, так и скрыто, в рамках возможностей встроенных в приложения интерпретаторов команд.

Проведенный анализ известных угроз современным универсальным ОС полностью подтверждает, что большая их часть обусловлена именно реализуемым в ОС концептуальным подходом, состоящим в реализации схемы распределенного администрирования механизмов защиты. В рамках этой схемы пользователь рассматривается как доверенное лицо, являющееся элементом схемы администрирования и имеющее возможность назначать/изменять ПРД. При этом он не воспринимается как потенциальный злоумышленник, который может сознательно или несознательно осуществить НСД к информации, следовательно назначение механизмов добавочной защиты ОС состоит в реализации централизованной схемы администрирования

механизмов защиты, в рамках которой будет осуществляться противодействие НСД пользователя к информации.

9.3. СИСТЕМА БЕЗОПАСНОСТИ ОПЕРАЦИОННОЙ СИСТЕМЫ WINDOWS NT

Операционная система Windows NT всегда обладала прекрасными и широко применимыми на практике возможностями защиты. Однократная регистрация в домене Windows NT предоставляет пользователям доступ к ресурсам всей корпоративной сети.

Полноценный набор инструментов Windows NT Server облегчает администраторам управление системой защиты и ее поддержку. Например, администратор может контролировать круг пользователей, имеющих права доступа к сетевым ресурсам: файлам, каталогам, серверам, принтерам и приложениям. Учетными записями пользователей и правами для каждого ресурса можно управлять централизованно.

С помощью простых графических инструментов администратор задает принадлежность к группам, допустимое время работы, срок действия и другие параметры учетной записи. Администратор получает возможность аудита всех событий, связанных с защитой доступа пользователей к файлам, каталогам, принтерам и иным ресурсам. Система также способна блокировать учетную запись пользователя, если число неудачных попыток регистрации превышает заранее определенное. Адми

нистраторы вправе устанавливать срок действия паролей, принуждать пользователей к периодической смене паролей и выбору паролей, затрудняющих несанкционированный доступ.

С точки зрения пользователя система защиты Windows NT Server полноценна и несложна в обращении. Простая процедура регистрации обеспечивает доступ к соответствующим ресурсам. Для пользователя невидимы такие процессы, как шифрование пароля на системном уровне. Пользователь сам определяет права доступа к тем ресурсам, которыми владеет. Например, чтобы разрешить совместное использование своего документа, он указывает, кто и как может с ним работать. Разумеется, доступ к ресурсам предприятия контролируется только администраторами с соответствующими полномочиями.

Более глубокий уровень безопасности - то, как Windows NT Server защищает данные, находящиеся в физической памяти компьютера. Доступ к ним предоставляется только имеющим на это право программам. Если данные больше не содержатся на диске, система предотвращает несанкционированный доступ к той области диска, где они содержались. При такой системе защиты никакая программа не "подсмотрит" в виртуальной памяти машины информацию, с которой оперирует в данный момент другое приложение.

Удаленный доступ через открытые сети и связь предприятий через Интернет стимулируют постоянное и быстрое развитие технологий безопасности. В качестве примера можно выделить сертификаты открытых ключей и динамические пароли. Архитектура безопасности Windows NT однозначно оценивается как превосходящая и эти, и многие будущие технологии. Перечислим функции безопасности Windows NT:

1. Информация о доменных правилах безопасности и учетная информация хранятся в каталоге Active Directory (служба каталогов Active Directory обеспечивает тиражирование и доступность учетной информации на многих контроллерах домена, а также позволяет удаленное администрирование).
2. В Active Directory поддерживается иерархичное пространство имен пользователей, групп и учетных записей машин (учетные записи могут быть сгруппированы по организационным единицам).
3. Административные права на создание и управление группами учетных записей пользователей могут быть делегированы на уровень организационных единиц (возможно установление дифференцированных прав доступа к отдельным свойствам пользовательских объектов).
4. Тиражирование Active Directory позволяет изменять учетную информацию на любом контроллере домена, а не только на первичном (копии Active Directory,

хранящиеся на других контроллерах домена, обновляются и синхронизируются автоматически).

5. Доменная модель изменена и использует Active Directory для поддержки многоуровневого дерева доменов (управление доверительными отношениями между доменами упрощено в пределах всего дерева доменов).

6. В систему безопасности включены новые механизмы аутентификации, такие как Kerberos v5 и TLS (Transport Layer Security), базирующиеся на стандартах безопасности Интернета.

7. Протоколы защищенных каналов (SSL 3.0/TLS) обеспечивают поддержку надежной аутентификации клиента (осуществляется сопоставление мандатов пользователей в форме сертификатов открытых ключей с существующими учетными записями Windows NT).

8. Дополнительно к регистрации посредством ввода пароля может поддерживаться аутентификация с использованием смарт-карт.

В состав Windows NT входит Microsoft Certificate Server, позволяющий выдавать сотрудникам и партнерам сертификаты X.509 версии 3. Системные администраторы могут указывать, сертификаты каких уполномоченных являются доверяемыми в системе и, таким образом, контролировать аутентификацию доступа к ресурсам.

Внешние пользователи, не имеющие учетных записей Windows NT, могут быть аутентифицированы с помощью сертификатов открытых ключей и соотнесены с существующей учетной записью. Права доступа, назначенные для этой учетной записи, определяют права внешних пользователей на доступ к ресурсам.

В распоряжении пользователей находятся простые средства управления парами закрытых (открытых) ключей и сертификатами, используемые для доступа к ресурсам системы.

Технология шифрования встроена в операционную систему и позволяет использовать цифровые подписи для идентификации потоков.

9.3.1. СЕРВЕР АУТЕНТИФИКАЦИИ KERBEROS

Kerberos - это программный продукт, разработанный в середине 1980-х гг. в Массачусетском технологическом институте и претерпевший с тех пор ряд принципиальных изменений. Клиентские компоненты Kerberos присутствуют в большинстве современных операционных систем.

Keiberos предназначен для решения следующей задачи. Имеется открытая (незащищенная) сеть, в узлах которой сосредоточены субъекты - пользователи, а также клиентские и серверные программные системы. Каждый субъект обладает секретным ключом. Чтобы субъект С мог доказать свою подлинность субъекту S (без этого S не станет обслуживать С), он должен не только назвать себя, но и продемонстрировать знание секретного ключа. С не может просто послать S свой секретный ключ, во-первых, потому, что сеть открыта (доступна для пассивного и активного прослушивания), а, во-вторых, потому, что S не знает (и не должен знать) секретный ключ С. Требуется менее прямолинейный способ демонстрации знания секретного ключа.

Система Kerberos представляет собой доверенную третью сторону (т.е. сторону, которой доверяют все), владеющую секретными ключами обслуживаемых субъектов и помогающую им в попарной проверке подлинности.

Чтобы с помощью Kerberos получить доступ к S (обычно это сервер), С (как правило - клиент) посылает Kerberos запрос, содержащий сведения о нем (клиенте) и о запрашиваемой услуге. В ответ Kerberos возвращает так называемый билет, зашифрованный секретным ключом сервера, и копию части информации из билета, зашифрованную секретным ключом клиента. Клиент должен расшифровать вторую порцию данных и переслать ее вместе с билетом серверу. Сервер, расшифровав билет, может сравнить его содержимое с дополнительной информацией, присланной клиентом. Совпадение свидетельствует

о том, что клиент смог расшифровать предназначенные ему данные (ведь содержимое билета никому, кроме сервера и Kerberos, недоступно), т.е. продемонстрировал знание секретного ключа. Значит, клиент - именно тот, за кого себя выдает. Подчеркнем, что секретные ключи в процессе проверки подлинности не передавались по сети (даже в зашифрованном виде) - они только использовались для шифрования. Как организован первоначальный обмен ключами между Kerberos и субъектами и как субъекты хранят свои секретные ключи - вопрос отдельный.

Проиллюстрируем описанную процедуру рис. 9.5, на котором обозначено: c и s - сведения (например, имя), соответственно, о клиенте и сервере; d1 и d2 - дополнительная (по отношению к билету) информация; Tc.s - билет для клиента C на обслуживание у сервера S; Kc и Ks - секретные ключи клиента и сервера; {info}K - информация info, зашифрованная ключом K.

Клиент C

Клиент C Kerbors: c. s....

{Клиент направляет Kerberos сведения

□ себе и о запрашиваемом сервере}

Kerboros Клиент C: {d1 }Kc, {Tc_s}Ks {Kerboros возвращает билет, зашифрованный ключом сервера, и дополнительную информацию, зашифрованную ключом клиента}

Клиент C Сервер S: d2. {Tc.s}Ks {Клиент направляет на сервер билет и дополнительную информацию}

Сервер S

Рис.

Рис. 9.5. Проверка сервером S подлинности клиента C

Протокол аутентификации Kerberos определяет взаимодействие между клиентом и сетевым сервисом аутентификации, известным как KDC (Key Distribution Center). В Windows NT KDC используется как сервис аутентификации на всех контроллерах домена. Домен Windows NT эквивалентен области Kerberos, но к ней обращаются как к домену. Реализация протокола Kerberos в Windows NT основана на определении Kerberos в RFC1510, клиент Kerberos реализован в виде ПФБ (поставщика функций безопасности) Windows NT, основанном на SSPI. Начальная аутентификация Kerberos интегрирована с процедурой WinLogon. Сервер Kerberos (KDC) интегрирован с существующими службами безопасности Windows NT, исполняемыми на контроллере домена. Для хранения информации о пользователях и группах он использует службу каталогов Active Directory.

Протокол Kerberos усиливает существующие функции безопасности Windows NT и добавляет новые:

1) повышенная скорость аутентификации при установлении начального соединения (сервер приложений не обращается к контроллеру домена для аутентификации клиента);

2) делегирование аутентификации в многоярусных архитектурах клиент-сервер (при подключении клиента к серверу, последний имперсонирует (олицетворяет) клиента в этой системе, но если серверу для завершения транзакции нужно выполнить сетевое подключение к другому серверу, протокол Kerberos позволяет делегировать аутентификацию первого сервера и подключиться ко второму от имени клиента);

3) транзитивные доверительные отношения для междудоменной аутентификации (т.е пользователь может быть аутентифицирован в любом месте дерева доменов) упрощают управление доменами в больших сетях с несколькими доменами.

Интеграция Kerberos. Протокол Kerberos полностью интегрирован с системой безопасности и контроля доступа Windows NT. Начальная регистрация в Windows NT обеспечивается процедурой WinLogon, использующей ПФБ Kerberos для получения начального билета TGT. Другие компоненты системы, например, Redirector, применяют

интерфейс SSPI к ПФБ Kerberos для получения сеансового билета для удаленного доступа к файлам сервера SMB.
Взаимодействие Kerberos. Протокол Kerberos версии 5 реализован в различных системах и используется для единообразия аутентификации в распределенной сети.
Под взаимодействием Kerberos подразумевается общий протокол, позволяющий учетным записям аутентифицированных пользователей, хранящимся в одной базе, осуществлять доступ ко всем сервисам в гетерогенной среде. Взаимодействие Kerberos основывается на следующих характеристиках:
1) общий протокол аутентификации пользователя или сервиса по основному имени при сетевом подключении;
2) возможность определения доверительных отношений между областями Kerberos и создания ссылочных запросов билетов между областями;
3) поддержка определенных в RFC 1510 требований к взаимодействию, относящихся к алгоритмам шифрования и контрольных сумм, взаимной аутентификации и другим возможностям билетов;
4) поддержка форматов маркера безопасности Kerberos версии 5 для установления контекста и обмена сообщениями.
Поддержка Kerberos открытых ключей. В Windows NT также реализованы расширения протокола Kerberos, поддерживающие дополнительно к аутентификации с совместно используемым секретным ключом аутентификацию, основанную на парах открытого (закрытого) ключа. Поддержка открытых ключей позволяет клиентам запрашивать начальный ключ TGT с помощью закрытого ключа, в то время как KDC проверяет запрос с помощью открытого ключа, полученного из сертификата X.509 (хранится в пользовательском объекте в каталоге Active Directory), Сертификат пользователя может быть выдан

как сторонним уполномоченным сертификации (Certification Authority), так и Microsoft Certificate Server, входящим в Windows NT. После начальной аутентификации закрытым ключом используются стандартные протоколы Kerberos для получения сеансовых билетов на доступ к сетевым службам.
Модель безопасности Windows NT обеспечивает однородный и унифицированный механизм контроля за доступом к ресурсам домена на основе членства в группах. Компоненты безопасности Windows NT доверяют хранимой в каталоге информации о защите. Например, сервис аутентификации Windows NT хранит зашифрованные пароли пользователей в безопасной части каталога объектов пользователя. По умолчанию операционная система "считает", что правила безопасности защищены и не могут быть изменены кем-либо несанкционированно. Общая политика безопасности домена также хранится в каталоге Active Directory.
Делегирование административных полномочий - гибкий инструмент ограничения административной деятельности рамками части домена. Этот метод позволяет предоставить отдельным сотрудникам возможность управления пользователями или группами в заданных пределах и, в то же время, не дает им прав на управление учетными записями, относящимися к другим подразделениям.
Права на определение новых пользователей или создание групп пользователей делегируются на уровне OU или контейнера, в котором создана учетная запись.
Существует три способа делегирования административных полномочий:
1) на изменение свойств определенного контейнера, например, LocalDomainPolicies самого домена;
2) на создание и удаление дочерних объектов определенного типа (пользователи, группы, принтеры и пр.) внутри OU;
3) на обновление определенных свойств некоторых дочерних объектов внутри OU (например, право устанавливать пароль для объектов типа User).

Делегировать полномочия просто. Достаточно выбрать лицо, которому будут делегированы полномочия, и указать, какие именно полномочия передаются. Интерфейс программы администрирования Active Directory позволяет без затруднений просматривать информацию о делегировании, определенную для контейнеров.

Наследование прав доступа означает, что информация об управлении доступом, определенная в высших слоях контейнеров в каталоге, распространяется ниже - на вложенные контейнеры и объекты-листья. Существуют две модели наследования прав доступа: динамическая и статическая. При динамическом наследовании права определяются путем оценки разрешений на доступ, назначенных непосредственно для объекта, а также для всех родительских объектов в каталоге. Это позволяет эффективно управлять доступом к части дерева каталога, внося изменения в контейнер, влияющий на все вложенные контейнеры и объекты-листья. Обратная сторона такой гибкости - недостаточно высокая производительность из-за времени определения эффективных прав доступа при запросе пользователя.

В Windows NT реализована статическая форма наследования прав доступа, иногда также называемая наследованием в момент создания. Информация об управлении доступом к контейнеру распространяется на все вложенные объекты контейнера. При создании нового объекта наследуемые права сливаются с правами доступа, назначаемыми по умолчанию. Любые изменения наследуемых прав доступа, выполняемые в дальнейшем на высших уровнях дерева, должны распространяться на все дочерние объекты. Новые наследуемые права доступа распространяются на объекты Active Directory в соответствии с тем, как эти новые права определены. Статическая модель наследования позволяет увеличить производительность.

9.3.2. ЭЛЕМЕНТЫ БЕЗОПАСНОСТИ СИСТЕМЫ

Рассмотрим вопросы реализации политики безопасности: управление учетными записями пользователей и групп, исполнение и делегирование административных функций.

Учетные записи пользователей и групп. Любой пользователь Windows NT характеризуется определенной учетной записью. Под учетной записью понимается совокупность прав и дополнительных параметров, ассоциированных с определенным пользователем. Кроме того, пользователь принадлежит к одной или нескольким группам. Принадлежность к группе позволяет быстро назначать права доступа и полномочия.

К встроенным учетным записям пользователей относятся:

- Guest- учетная запись, фиксирующая минимальные привилегии гостя;
- Administrator - встроенная учетная запись для пользователей, наделенных максимальными привилегиями;
- Krbtgt - встроенная учетная запись, используемая при начальной аутентификации Kerberos.

Кроме них имеются две скрытые встроенные учетные записи:

- System - учетная запись, используемая операционной системой;
- Creator owner - создатель (файла или каталога).

Перечислим встроенные группы:

- локальные (Account operators; Administrators; Backup operators; Guests; Print operators; Replicator; Server operators; Users);
- глобальные (Domain guests - гости домена; Domain Users - пользователи домена; Domain Admins - администраторы домена).

Помимо этих встроенных групп имеется еще ряд специальных групп:

- Everyone - в эту группу по умолчанию включаются вообще все пользователи в системе;
- Authenticated users - в эту группу включаются только аутентифицированные пользователи домена;
- Self - сам объект.

Для просмотра и модификации свойств учетной записи достаточно щелкнуть имя пользователя или группы и на экране появится диалоговое окно User Properties:
General - общее описание пользователя;
Address - домашний и рабочий адрес пользователя;

Account - обязательные параметры учетной записи;
Telephone/notes - необязательные параметры;
Organization - дополнительные необязательные сведения;
Membership - обязательная информация о принадлежности пользователя к группам;
Dial-in - параметры удаленного доступа;
Object - идентификационные сведения о пользовательском объекте;
Security - информация о защите объекта.
Локальная политика безопасности регламентирует правила безопасности на локальном компьютере. С ее помощью можно распределить административные роли, конкретизировать привилегии пользователей, назначить правила аудита.
По умолчанию поддерживаются следующие области безопасности:

- политика безопасности - задание различных атрибутов безопасности на локальном и доменном уровнях; так же охватывает некоторые установки на машинном уровне;
- управление группами с ограничениями - позволяет управлять членством в группах, которые, по мнению администратора, "чувствительны" с точки зрения безопасности системы;
- управление правами и привилегиями - позволяет редактировать список пользователей и их специфических прав и привилегий;
- деревья объектов - включают три области защиты: объекты каталога Active Directory, ключи реестра, локальную файловую систему; для каждого объекта в дереве шаблоны безопасности позволяют конфигурировать и анализировать характеристики дескрипторов защиты, включая владельцев объекта, списки контроля доступа и параметры аудита;
- системные службы (сетевые или локальные) - построенные соответствующим образом дают возможность независимым производителям программного обеспечения расширять редактор конфигураций безопасности для устранения специфических проблем.

Конфигурирование безопасности. Для конфигурирования параметров безопасности системы используются шаблоны.
Управление доступом к реестру. Реестр - это дерево объектов. Доступ к каждому объекту в дереве должен быть регламентирован. Выбрав в окне обзорного просмотра ветвь, соответствующую шаблону Custom, щелкните папку Registry. В правой части окна появится список ветвей реестра, доступ к которым можно ограничивать. В шаблоне, поставляемом с редактором, приведена ветвь MACHINE\HARDWARE, которую надо истолковывать как HKEY_LOCAL_MACHINE\Hardware. Чтобы добавить к дереву новые ветви, их надо в явном виде прописать в шаблоне с помощью любого текстового редактора. Для разграничения доступа к выбранной ветви реестра дважды щелкните ее имя и укажите нужный тип доступа и имя соответствующей учетной записи. Изменения будут занесены в шаблон.

9.4. ЗАЩИТА В ОПЕРАЦИОННОЙ СИСТЕМЕ UNIX

Операционная система Unix относится к категории многопользовательских многопрограммных ОС, работающих в режиме разделения времени. Богатые возможности, заложенные в ОС Unix, сделали ее наиболее популярной в мире. ОС Unix поддерживается практически на всех типах ЭВМ.
Организация работ в ОС Unix основана на понятии последовательного процесса как единицы работы, управления и потребления ресурсов. Взаимодействие процессов внутри ядра (процесс вызывает ядро как подпрограмму) происходит по принципу сопрограмм. Последовательность вычислений внутри процесса строго выдерживается: процесс, в

частности, не может активизировать ввод-вывод и продолжать вычисление параллельно с ним. В этом случае требуется создать параллельный процесс.

Ядро ОС Unix состоит из двух основных частей: управления процессами и управления устройствами. Управление процессами резервирует ресурсы, определяет последовательность выполнения процессов и принимает запросы на обслуживание. Управление устройствами контролирует передачу данных между ОП и периферийными устройствами.

В любой момент времени выполняется либо программа пользователя (процесс), либо команда ОС. В каждый момент времени лишь один пользовательский процесс активен, а все остальные приостановлены. Ядро ОС Unix служит для удовлетворения потребностей процессов.

Процесс - это программа на этапе выполнения. В некоторый момент времени программе могут соответствовать один или несколько процессов, или не соответствовать ни одного. Считается, что процесс является объектом, учтенным в специальной таблице ядра системы. Наиболее важная информация о процессе хранится в двух местах: в таблице процессов и в таблице пользователя, называемой также контекстом процесса. Таблица процессов всегда находится в памяти и содержит на каждый процесс по одному элементу, в котором отражается состояние процесса: адрес в памяти или адрес свопинга, размер, идентификаторы процесса и запустившего его пользователя. Таблица пользователя существует для каждого активного процесса и к ней могут непосредственно адресоваться только программы ядра (ядро резервирует по одному контексту на каждый активный процесс). В этой таблице содержится информация, требуемая во время выполнения процесса: идентификационные номера пользователя и группы, предназначенные для определения привилегий доступа к файлам, ссылки на системную таблицу файлов для всех открытых процессом файлов, указатель на индексный дескриптор текущего каталога в таблице индексных дескрипторов и список реакций на различные ситуации. Если процесс приостанавливается, контекст становится недоступным и немодифицируемым.

Каталоги файловой системы ОС Unix "спрятаны" от пользователей и защищены механизмами ОС. Скрытой частью файловой организации в ОС Unix является индексный дескриптор файла, который описывает расположение файла, его длину, метод доступа к файлу, даты, связанные с историей создания файла, идентификатор владельца и т.д.

Рис.

Рис. 9.6. Структура данных ядра ОС Unix

Работа с таблицами является привилегией ядра, что обеспечивает сохранность и безопасность системы. Структура данных ядра ОС, обеспечивающих доступ к файлам, приведена на рис. 9.6.

При взаимодействии с ОС Unix пользователь может обращаться к большому числу информационных объектов или файлов, объединенных в каталоги. Файловая система ОС Unix имеет иерархическую структуру.

В ОС Unix используется четыре типа файлов: обычные, специальные, каталоги, а в некоторых версиях ОС и FIFO- файлы (First In - First Out). Обычные файлы содержат данные пользователей. Специальные файлы предназначены для организации взаимодействия с устройствами ввода-вывода. Доступ к любому устройству реализуется как обслуживание запроса к специальному (дисковому) файлу. Каталоги используются системой для поддержания файловой структуры. Особенность каталогов состоит в том, что пользователь может читать их содержимое, но выполнять записи в каталоги (изменять структуру каталогов) может только ОС. В ОС Unix, организуются именованные

программные каналы, являющиеся соединительным средством между стандартным выводом одной программы и стандартным вводом другой.
Схема типичной файловой системы ОС Unix приведена на рис. 9.7. Рассмотрим основные механизмы защиты данных,

Рис.

Рис. 9.7. Схема файловой системы ОС Unix
Управление доступом к системе. При включении пользователя в число абонентов ему выдается регистрационное имя (идентификатор) для входа в систему и пароль, который служит для подтверждения идентификатора пользователя. В отдельных версиях ОС Unix, помимо идентификатора и пароля, требуется ввод номера телефона, с которого выполняется подключение к системе. Администратор системы и пользователь могут изменить пароль командой passwd. При вводе этой команды ОС запрашивает ввод текущего пароля, а затем требует ввести новый пароль. Если предложенный пароль не удовлетворяет требования системы, то запрос на ввод пароля может быть повторен. Если предложенный пароль удовлетворителен, ОС просит ввести его снова, чтобы убедиться в корректности ввода пароля.
Пользователи, которым разрешен вход в систему, перечислены в учетном файле пользователей /etc/passwd. Этот текстовый файл содержит следующие данные: имя пользователя, зашифрованный пароль, идентификатор пользователя, иден

тификатор группы, начальный текущий каталог и имя исполняемого файла, используемого в качестве интерпретатора команд. Пароль шифруется, как правило, с использованием DES-алгоритма.
Управление доступом к данным. Операционная система Unix поддерживает для любого файла комплекс характеристик, определяющих санкционированность доступа, тип файла, его размер и точное местоположение на диске. При каждом обращении к файлу система проверяет право пользоваться им. Операционная система Unix допускает выполнение трех типов операций над файлами: чтение, запись и выполнение. Чтение файла означает, что доступно его содержимое, а запись - что возможны изменения содержимого файла. Выполнение приводит либо к загрузке файла в ОП, либо к выполнению содержащихся в файле команд системного монитора Shell. Разрешение на выполнение каталога означает, что в нем допустим поиск с целью формирования полного имени на пути к файлу. Любой из файлов в ОС Unix имеет определенного владельца и привязан к некоторой группе. Файл наследует их от процесса, создавшего файл. Пользователь и группа, идентификаторы которых связаны с файлом, считаются его владельцами.
Идентификаторы пользователя и группы, связанные с процессом, определяют его права при доступе к файлам. По отношению к конкретному файлу все процессы делятся на три категории:
1) владелец файла (процессы, имевшие идентификатор пользователя, совпадающий с идентификатором владельца файла);
2) члены группы владельца файла (процессы, имеющие идентификатор группы, совпадающий с идентификатором группы, которой принадлежит файл);
3) прочие (процессы, не попавшие в первые две категории).
Владелец файла обладает одними привилегиями на доступ к нему, члены группы, в которую входит файл - другими, все остальные пользователи - третьими. Каждый файл содержит код защиты, который присваивается файлу при его создании. Код защиты располагается в индексном дескрипторе файла и содержит десять символов, причем первый символ определяет тип файла, а последующие девять - право на доступ к нему. Три вида операций (чтение, запись и выполнение) и три категории (уровни привилегий на

доступ: владельцев, групп и прочих пользователей) дают в совокупности девять возможных вариантов разрешений или запретов на доступ к файлу. Первые три символа определяют возможности чтения (r), записи (w) и выполнения (e) на уровне владельца, следующие три - на уровне группы, в которую входит владелец, и последние три - на уровне остальных пользователей. Наличие символов r, w и e указывает на соответствующее разрешение.

Если процесс требует доступа к файлу, то сначала определяется категория, в которую по отношению к этому файлу он попадает. Затем из кода защиты выбираются те три символа, которые соответствуют данной категории, и выполняется проверка: разрешен ли процессу требуемый доступ. Если доступ не разрешен, системный вызов, посредством которого процесс сделал запрос на доступ, отвергается ядром ОС.

По соглашению, принятому в ОС Unix, привилегированный пользователь имеет идентификатор, равный нулю. Процесс, с которым связан нулевой идентификатор пользователя, считается привилегированным. Независимо от кода защиты файла привилегированный процесс имеет право доступа к файлу для чтения и записи. Если в коде защиты хотя бы одной категории пользователей (процессов) есть разрешение на выполнение файла, привилегированный процесс тоже имеет право выполнять этот файл.

С помощью специальных команд владелец файла (привилегированный пользователь) может изменять распределение привилегий. Команда Change mode позволяет изменить код защиты, команда Change owner меняет право на владение файлом, а команда Change group - принадлежность к той или иной группе. Пользователь может изменять режимы доступа только для файлов, которыми он владеет.

Защита хранимых данных. Для защиты хранимых данных в составе ОС Unix имеется утилита crypt, которая читает данные со стандартного ввода, шифрует их и направляет на стандартный вывод. Шифрование применяется при необходимости предоставления абсолютного права владения файлом.

Восстановление файловой системы. Операционная система Unix поддерживает три основных набора утилит копирования: программы volcopy/labelit, dump/restor и cpio. Программа volcopy целиком переписывает файловую систему, проверяя с помощью программы labelit соответствие меток требуемых томов. Программа dump обеспечивает копирование лишь тех файлов, которые были записаны позднее определенной даты (защита накоплением). Программа restor может анализировать данные, созданные программой dump, и восстанавливать отдельные файлы или всю файловую систему полностью. Программа cpio предназначена для создания одного большого файла, содержащего образ всей файловой системы или какой-либо ее части.

Для восстановления поврежденной, например, в результате сбоев в работе аппаратуры файловой системы используются программы fsck и fsdb.

За сохранность файловой системы, адаптацию программного обеспечения к конкретным условиям эксплуатации, периодическое копирование пользовательских файлов, восстановление потерянных данных и другие операции ответственность возложена на администратора системы.

Усложненное управление доступом. В составе утилит ОС Unix находится утилита cron, которая предоставляет возможность запускать пользовательские программы в определенные моменты (промежутки) времени и, соответственно, ввести временные параметры для ограничения доступа пользователей.

Для управления доступом в ОС Unix также применяется разрешение установки идентификатора владельца. Такое разрешение дает возможность получить привилегии владельца файла на время выполнения соответствующей программы. Владелец файлов может установить режим, в котором другие пользователи имеют возможность назначать собственные идентификаторы режима.

Доступ, основанный на полномочиях, использует соответствие меток. Для этого вводятся метки объектов (файлов) и субъектов (процессов), а также понятия доминанты и

равенства меток (для выражения отношения между метками). Создаваемый файл наследует метку от создавшего его процесса. Вводятся соотношения, определяющие права процессов по отношению к файлам.

Интерфейс дискретного доступа существенно детализирует имеющиеся механизмы защиты ОС Unix. Вводимые средства можно разделить на следующие группы:

1) работа со списками доступа при дискретной защите;
2) проверка права доступа;
3) управление доступом на основе полномочий;
4) работа привилегированных пользователей.

В рамках проекта Posix создан интерфейс системного администратора. Указанный интерфейс определяет объекты и множества действий, которые можно выполнить над объектами. В качестве классов субъектов и объектов предложены пользователь, группа пользователей, устройство, файловая система, процесс, очередь, вход в очередь, машина, система, администратор, программное обеспечение и др. Определены атрибуты таких классов, операции над классами и события, которые могут с ними происходить.

9.5. ЗАЩИТА В ОПЕРАЦИОННОЙ СИСТЕМЕ NOVELL NETWARE

Авторизация доступа к данным сети. В NetWare реализованы три уровня защиты данных (рис. 9.8).

Здесь под аутентификацией понимается:

1) процесс подтверждения подлинности клиента при его подключении к сети,
2) процесс установления подлинности пакетов, передаваемых между сервером и рабочей станцией.

Права по отношению к файлу (каталогу) определяют, какие операции пользователь может выполнить с файлом (каталогом). Администратор может для каждого клиента сети определить права по отношению к любому сетевому файлу или каталогу.

Клиент 1 Клиент 2 Уровни

Рис.

Рис. 9.8. Уровни защиты данных в Novell NetWare

Атрибуты определяют некоторые системные свойства файлов (каталогов). Они могут быть назначены администратором для любого сетевого файла или каталога.

Например, чтобы записать данные в файл, клиент должен:

1) знать свой идентификатор и пароль для подключения к сети,
2) иметь право записи данных в этот файл,
3) файл должен иметь атрибут, разрешающий запись данных.

Следует отметить, что атрибуты файла (каталога) имеют более высокий приоритет, чем права пользователей по отношению к этому файлу.

Аутентификация пользователей при подключении к сети. Подключение к сети выполняется с помощью утилиты LOGIN.EXE. Эта программа передает на сервер идентификатор, введенный пользователем.

По этому идентификатору NetWare выполняет поиск соответствующего объекта пользователя в системной базе данных сетевых ресурсов. Если в базе данных хранится значение пароля для этого клиента, то NetWare посылает на рабочую станцию зашифрованный с помощью пароля открытый ключ (симметричное шифрование). На рабочей станции этот ключ расшифровывается с помощью пароля, введенного пользователем, и используется для получения подписи запроса (пакета) к серверу о продолжении работы. Сервер расшифровывает эту подпись с помощью закрытого ключа (асимметричное шифрование), проверяет ее и посылает подтверждение на рабочую станцию. В дальнейшем каждый NCP-пакет снабжается подписью, получаемой в

результате кодирования открытым ключом контрольной суммы содержимого пакета и случайного числа Nonce. Это число генерируется для каждого сеанса. Поэтому подписи пакетов не повторяются для разных сеансов, даже если пользователь выполняет те же самые действия
1
2
3

NCP-пакеты могут подписываться и рабочими станциями, и файловым сервером. Для инициирования включения подписи в NCP-пакеты администратор может задать один из следующих уровней:
0 - сервер не подписывает пакет;
1 - сервер подписывает пакет, если этого требует клиент (уровень на станции больше или равен 2);
2 - сервер подписывает пакет, если клиент также способен это сделать (уровень на станции больше или равен 1);
3 - сервер подписывает пакет и требует этого от всех клиентов (иначе подключение к сети невозможно).
Права пользователей по отношению к каталогам и файлам. Права, которые могут быть предоставлены пользователю (или группе пользователей) по отношению к каталогу или файлу, перечислены в табл. 9.3.
Права и фильтры (маски) наследуемых прав назначаются администратором сети с помощью утилит NetWare. Но назначение прав для каждого пользователя по отношению ко всем требуемым файлам и каталогам - это утомительная задача. В NetWare предлагается механизм наследования прав. Прежде всего введем некоторые определения.
Опекун (Trustees) - это пользователь (группа пользователей, другой объект), которому администратор с помощью утилиты (например, FILER) явно назначает права по отношению к какому-либо файлу или каталогу. Такие права называются опекунскими назначениями.
Фильтр наследуемых прав (IRF - Inherited Right Filter) - это свойство файла (каталога), определяющее, какие права данный файл (каталог) может унаследовать от родительского каталога. Фильтр назначается администратором с помощью утилиты (например, FILER).
Наследуемые права - права, передаваемые (распространяемые) от родительского каталога.
Эффективные права - права, которыми пользователь реально обладает по отношению к файлу или каталогу.
9.3. Список возможных прав по отношению к каталогу или файлу

Право	Обозначение	Описание
Supervisor	S	Предоставляет все права по отношению к каталогу или файлу, включая возможность назначения этого права другим пользователям. Не блокируется фильтром наследуемых прав IRF. Это право не может быть удалено ниже по дереву каталогов
Read	R	Чтение существующего файла (просмотр содержимого текстового файла, просмотр записей в файле базы данных и т.д.)
Write	W	Запись в существующий файл (добавление, удаление частей текста, редактирование записей базы данных)
Create	C	Создание в каталоге новых файлов (и запись в них) и подкаталогов. На уровне файла позволяет восстанавливать файл, если он был ошибочно удален
Erase	E	Удаление существующих файлов и каталогов
Modify	M	Изменение имен и атрибутов (файлов и каталогов), но не содержимого файлов

File Scan F Просмотр в каталоге имен файлов и подкаталогов. По отношению к файлу - возможность видеть структуру каталогов от корневого уровня до этого файла (путь доступа)
Access
Control A Возможность предоставлять другим пользователям все права, кроме Supervisor. Возможность изменять фильтр наследуемых прав IRF
Права доступа к объектам NDS и их свойствам. Системная база данных сетевых ресурсов (СБДСР) представляет собой совокупность объектов, их свойств и значений этих свойств. В NetWare 4.x эта база данных называется NDS (NetWare Directory Services), а в NetWare 3.x - Bindery. Объекты NDS связаны между собой в иерархическую структуру, которую часто называют деревом NDS. На верхних уровнях дерева (ближе к корню [Root]) описываются логические ресурсы, которые принято называть контейнерными объектами. На самом нижнем (листьевом) уровне располагаются описания физических ресурсов, которые называют оконечными объектами.
В качестве контейнерных объектов используются объекты типа [Root] (корень), C (страна), O (организация), OU (организационная единица). Оконечные объекты - это User (пользователь), Group (группа), NetWare Server (сервер NetWare), Volume (том файлового сервера), Directories (директория тома) и т.д. Оконечные объекты имеют единое обозначение - CN.

В NetWare 4.x разработан механизм защиты дерева NDS. Этот механизм очень похож на механизм защиты файловой системы, который был рассмотрен ранее. Чтобы облегчить понимание этого механизма, оконечный объект можно интерпретировать как файл, а контейнерный объект - как каталог, в котором могут быть созданы другие контейнерные объекты (как бы подкаталоги) и оконечные объекты (как бы файлы). На рис. 9.9 представлена схема дерева NDS. Здесь символами [Root], C, O, OU обозначены контейнерные объекты, а символами CN - оконечные объекты.
В отличие от файловой системы здесь права по отношению к какому-либо объекту можно предоставить любому контейнерному или оконечному объекту дерева NDS. В частности допустимо рекурсивное назначение прав объекта по отношению к этому же объекту.
Права, которые могут быть предоставлены объекту по отношению к другому или тому же самому объекту, перечислены в табл. 9.4.
[Root]

0

0U
-CN
-CN
0U
-CN
-CN

Рис. 9. Схема дерева NDS 9.4. Список возможных прав по отношению к объекту

Рис. 9.
Право Обозна
чение Описание
Supervisor S Гарантирует все привилегии по отношению к объекту и его свойствам. В отличие от файловой системы это право может быть блокировано фильтром наследуемых прав IRF, который может быть назначен для каждого объекта

Browse B Обеспечивает просмотр объекта в дереве NDS
Create C Это право может быть назначено только по отношению к контейнерному объекту (контейнеру). Позволяет создавать объекты в данном и во всех дочерних контейнерах
Delete D Позволяет удалять объект из дерева NDS
Rename R Позволяет изменять имя объекта
C

Администратор сети может для каждого объекта в дереве NDS определить значения свойств этого объекта. Для объекта User - это имя Login, требования к паролю, пароль пользователя, пользовательский сценарий подключения и т.д.

Заключение

Контрольные вопросы

Смотри руководство по организации самостоятельной работы магистрантов.