**1 слайд**

Astra linux — это отечественный дистрибутив Linux на основе Debian, разработанный АО «НПО РусБИТех» (Русские базовые информационные технологии). Впервые Astra увидела свет в 2009 году. Пользовательский функционал закрыт по большей части компонентами из состава открытого программного обеспечения — стандартные механизмы Linux используются для выполнения основных задач, например, запуска приложений, виртуализации, поддержки аппаратного обеспечения. Компоненты собственной разработки в основном решают две ключевые задачи: обеспечения безопасности, а также взаимодействие графического интерфейса операционной системы и человека.

**2 слайд**

В Astra linux используются уникальные инструменты защиты информации:

* Встроенная система безопасности PARSEC
* Мандатное управление доступом
* Изоляция модулей
* Очистка оперативной и внешней памяти и гарантированное удаление файлов
* Маркировка документов
* Регистрация событий
* Защита информации в графической подсистеме
* Ограничение действий пользователя (режим «киоск»)
* Безопасность адресного пространства процессов
* Контроль замкнутости программной среды и целостности
* Средства организации единого пространства пользователей
* Защищенная среда виртуализации
* Защищенная реляционная СУБД Postgre SQL
* Защищенные комплексы программ электронной почты и гипертекстовой обработки данных

**3 слайд**

Astra linux соответствует требованиям безопасности всех регуляторов к ПО для импортозамещения:

* ФСТЭК РОССИИ - профиль защиты операционных систем типа «А» первого класса защиты, первый уровень доверия к операционным системам
* МИНЦИФРЫ РОССИИ - в реестре отечественного ПО Минцифры под номером 369 от 08.04.2016 г.
* МИНОБОРОНЫ РОССИИ - профиль защиты операционных систем типа «А» первого класса защиты, 1 уровень контроля отсутствия недекларированных возможностей, соответствие реальных и декларируемых возможностей

**4 слайд**

На данный момент существуют две разновидности системы, а именно "Common Edition" (релиз Орел) - вариант несертифицированной операционной системы. Рекомендован для применения в открытых сегментах инфраструктур, подключённых к сетям общего доступа, в образовательных учреждениях и других ИТ-системах, не требующих специальных средств защиты, а также для домашнего использования. Не может применяться в системах, обрабатывающих информацию ограниченного доступа, к которым предъявляются требования по защите информации.

И вторая версия "Special Edition" (релизы Смоленск и Воронеж) - не имеющая аналогов сертифицированная ОС со встроенными средствами защиты информации (СЗИ) для стабильных и безопасных ИТ-инфраструктур любого масштаба и бесперебойной работы с данными любой степени конфиденциальности.

Для различных процессорных архитектур разработаны специальные варианты исполнения с защитой всех компонентов и программных ресурсов.

**5 слайд**

Релиз Воронеж рекомендуется для обработки конфиденциальной информации в ГИС, в информационных системах персональных данных, а также в составе значимых объектов КИИ любого класса (уровня, категории) защищённости. Дополнительно используется в других информационных (автоматизированных) системах для обработки информации ограниченного доступа без содержания сведений, составляющих гостайну.

Релиз Смоленск рекомендуется для обработки информации любой категории доступа в ГИС, в информационных системах персональных данных, в составе значимых объектов КИИ, иных информационных (автоматизированных) системах, обрабатывающих информацию ограниченного доступа, в т.ч. содержащую сведения, составляющие гостайну до степени «особой важности» включительно.

По сути, релизы отличаются наличием мандатного управления доступом и классами АС по РД АС.

**6 слайд**

В новой версии Astra linux SE реализован ряд улучшений:

* 3 уровня защищенности ОС с удобным переходом с «Базового» и «Усиленного» режимов обеспечения безопасности на «Максимальный» — в ваших руках мощный функционал для достижения желаемой степени защиты данных в ИТ-инфраструктуре организации
* Полная поддержка контейнерной виртуализации с возможностью дополнительной изоляции и защиты контейнеров
* В обновленном интерфейсе — гибкая настройка уровней защиты под конкретные требования в части безопасности
* Обновленная и расширенная подсистема аудита (протоколирования)
* Основана на новой пакетной базе Debian 10

**7 слайд**

Все продукты Astra Linux входят в реестр Минкомсвязи России. Их используют крупные предприятия, государственные корпорации, министерства и структуры, например, Министерство обороны РФ, ФСБ, ФСО, МВД, РОСГВАРДИЯ, МЧС, Почта России, РОСКОСМОС, РУСГИДРО, ОКБ Сухой, РОСТЕХ, РОСНЕФТЬ, Дочерние предприятия РЖД, Дочерние предприятия Газпрома, Дочерние предприятия Росатома, Министерство развития Востока, Центр охраны здоровья животных, Пенсионный фонд России.

8 слайд

**Математическое описание базовой модели**

В МРОСЛ ДП-модели было обеспечено сочетание мандатного управления доступом, мандатного контроля целостности с перспективным ролевым управлением доступом. При этом особое внимание уделялось детальному описанию правил переходов системы из состояния в состояние, которые классифицированы на де-юре и де-факто правила.

9 слайд

В теории компьютерной безопасности важнейшим результатом исследования математической модели безопасности управления доступом и информационными потоками, как правило, считается обоснование в ее рамках условий безопасности или, наоборот, нарушения безопасности рассматриваемых систем. Для соответствия этому в МРОСЛ ДП-модели были приведены определения безопасного начального состояния системы (состояния, в котором отсутствуют запрещенные информационные потоки по памяти или по времени, фактическое владение субъект-сессиями друг другом, информационные потоки по памяти или доступы к сущностям, параметрически или функционально ассоциированным с субъект-сессиями, не нарушают правил мандатного контроля целостности) и определение трех смыслов нарушения безопасности системы:

10 слайд

* в смысле мандатного контроля целостности, позволяющее недоверенной субъект-сессии с низким уровнем целостности захватить контроль (фактическое владение) над доверенной субъект-сессии с высоким уровнем целостности;
* в смысле Белла–ЛаПадулы, результатом которого является создание запрещенного информационного потока по памяти «сверху-вниз»;
* в смысле контроля информационных потоков по времени — создание запрещенного информационного потока по времени «сверху-вниз» между сущностями.

Однако такое описание МРОСЛ ДП-модели имело достаточно существенный объем и являлось «монолитным», т. е. элементы модели давались в порядке, удобном для описания модели в целом. Из-за большого объема и монолитности модели, невозможности в таком виде ее поэтапной реализации затрудняется использование модели разработчиками ОССН, а также создание на ее основе новых моделей.

Все выше перечисленное стало причиной переработки МРОСЛ ДП-модели в ее иерархическое представление. Первоначально при его разработке в модель (по сравнению с ее «монолитным» представлением) не добавлялись новые элементы, а основной целью являлось формирование следующих четырех упорядоченных уровней:



11 слайд

* первый уровень (базовый) — модель системы ролевого управление доступом;
* второй уровень — модель системы ролевого управление доступом и мандатного контроля целостности;
* третий уровень — модель системы ролевого управление доступом, мандатного контроля целостности и мандатного управления доступом только с информационными потоками по памяти;
* четвертый уровень — модель системы ролевого управление доступом, мандатного контроля целостности и мандатного управления доступом с информационными потоками по памяти и по времени.

Каждый нижний уровень иерархического представления модели соответствует абстрактной системе, элементы которой не зависят от новых элементов, принадлежащих более высокому уровню модели, который, в свою очередь, наследует, а при необходимости корректирует или дополняет элементы нижнего уровня. Такой подход позволяет постепенно усложнять формулировки определений и утверждений модели по мере включения в нее соответствующих очередному рассматриваемому уровню элементов.

12 слайд

**Элементы состояния системы**

Основным элементам состояния системы в рамках модели, образующим конструкцию базового уровня иерархического представления МРОСЛ ДП-модели, соответствуют учетные записи пользователей, представляющие пользователей ОССН, функционирующих от их имени субъект-сессий — «активных» компонентов (процессов) ОССН, так называемые «пассивные» сущности, представляющие файлы, каталоги, сокеты и другие ресурсы ОССН, представляющие связи между пользователями и пассивными сущностями. Все эти элементы применяются для описания состояний рассматриваемой в рамках модели абстрактной системы, для чего используем следующие обозначения:

*U* — конечное непустое множество учетных записей пользователей (в любой ОССН задана хотя бы одна учетная запись пользователя);

*S* — конечное непустое множество субъект-сессий учетных записей пользователей (всех «активных» компонентов защищенной ОССН, при этом в любой ОССН есть хотя бы одна субъект-сессия, например, соответствующая системному процессу, реализующему процедуру входа пользователя в систему и запуска от имени его учетной записи процессов);

*user: S → U* — функция принадлежности субъект-сессии учетной записи пользователя, задающая для каждой субъект-сессии учетную запись пользователя, от имени которой она активизирована;

13 слайд

— конечное непустое множество сущностей (всех «пассивных» компонентов защищенной ОС, к которым назначаются права доступа, включающее файлы, каталоги, порты, сокеты, очереди, семафоры и другие объекты хранения данных, сетевого и межпроцессного взаимодействия), где — множество объектов (например, файлов), C — множество контейнеров (например, каталогов) и .

Также определим: *NAMES* — множество допустимых имен сущностей, ролей и административных ролей; entity\_name: C × E → 2NAMES — функция имен сущностей в составе сущностей-контейнеров. При этом для любых контейнеров c, cx ∈ C по определению выполняются условия:

* ;
* существует единственная сущность — «корневой контейнер» такая, что и, если , то существует единственная последовательность контейнеров ∈ C такая, что n 2 и .

В большинстве ОС сущности образуют иерархические, древовидные структуры, для представления которых в модели вводится определение иерархии сущностей.

Поэтому введем следующее определение:

14 слайд

**Определение.** Иерархией сущностей называется заданное на множестве сущностей *E* бинарное отношение «», удовлетворяющее условию: для двух сущностей , выполняется отношение , когда либо , либо существует последовательность сущностей , такая что В случае, когда для двух сущностей выполняется условие , будем говорить что сущность содержится в сущности-контейнере , и будем использовать обозначение . Определим функцию иерархии сущностей , где для выполняется

Аналогичная иерархия необходима и для построения структур из субъект-сессий, так как в ОССН для процесса может быть определен его родительский процесс и, если существуют, дочерние процессы, что в совокупности образует множество древовидных структур (лес).

15 слайд

**Определение.** Иерархией субъект-сессий называется заданное на множестве S отношение частичного порядка «», удовлетворяющее условию: если для субъект-сессии существуют субъект-сессии В случае, когда для двух субъект-сессий выполняются условия , будем говорить, что субъект-сессия является потомком , и будем использовать обозначение . Определим функцию иерархии субъект-сессий (сопоставляющую каждой субъект-сессии множество субъект-сессий , непосредственно в ней содержащихся), удовлетворяющую условиям:

* если субъект-сессия то , и не существует субъект-сессии такой, что .
* для любых субъект-сессий выполняется равенство

Заметим, что в рамках МРОСЛ ДП-модели иерархия сущностей задана не «абстрактной» функцией HE , а реализуемой явно в ОССН функцией имен сущностей в составе сущностей-контейнеров entity\_name, что, кроме того, позволит в дальнейшем описать правила предоставления содержимого контейнеров (например, списков файлов и каталогов, выдаваемых в реальной ОССН по команде ls) с учетом параметров мандатного управления доступом. При этом учтено наличие механизма создания «жестких» ссылок (hard link) в файловой системе ОССН, обеспечивающего возможность размещения сущностей-объектов одновременно в нескольких сущностях контейнерах, в том числе несколько «жестких» ссылок на одну сущность-объект в составе одной сущности-контейнере. Также описаны свойства корневой сущности-контейнера ROOT, как правило, соответствующей в реальных защищенных ОС корневому каталогу «/»

Еще одним важным отличием МРОСЛ ДП-модели от других ДП-моделей и классических моделей является то, что множество субъект-сессий не входит во множество сущностей. Это связно с тем, что в реальной ОССН функции управления доступом к субъект-сессиям (процессам) и сущностям (файлам, каталогам) реализуются раздельно. Таким образом, иерархия на множестве субъект-сессий определяется независимо от иерархии сущностей и при реализации в ОССН может быть задана двумя способами. Первый способ, когда существует явная связь между родительскими субъект-сессиями и их потомками (например, когда при завершении работы родительской субъект-сессии завершают работу ее субъект-сессии потомки, подчиненные родительской в иерархии). Второй способ, когда порожденная субъект-сессией другая субъект-сессия функционирует независимо, в этом случае подчинение ее в иерархии родительской субъект-сессии нецелесообразно.

16 слайд

**Ролевое управление доступом**

Ролевое управление доступом является развитием политики безопасности дискреционного управления доступом, при этом права доступа субъектов (сессий) системы к сущностям группируются с учетом специфики их применения, образуя роли. Задание ролей позволяет определить более четкие и понятные для пользователей компьютерных систем правила управления доступом, а также предоставлять возможность задать гибкие, изменяющиеся динамически в процессе функционирования компьютерной системы правила управления доступом.

Для введения ролей на базовом уровне иерархического представления МРОСЛ ДП-модели используем следующие обозначения:

*R* – множество ролей;

*AR –* множество административных ролей, при этом по определению (административные роли — особый вид ролей, предназначенный для изменения множеств прав доступа ролей, авторизации на роли, а также выполнения функций по администрированию системы, например управления мандатными уровнями конфиденциальности сущностей и субъект-сессий);

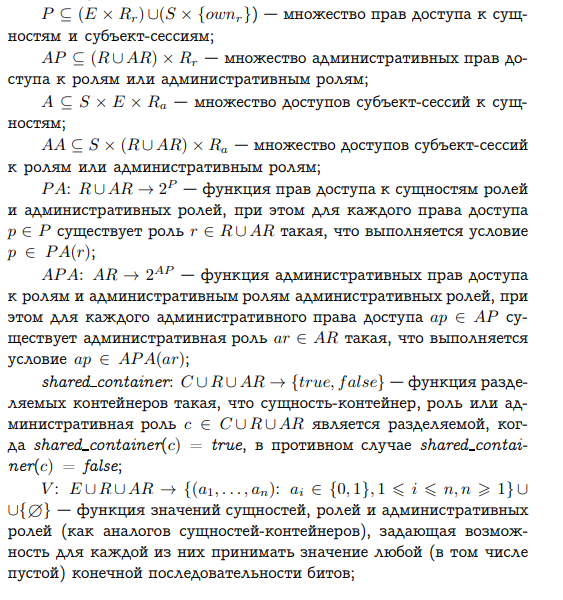
множество специальных ролей, которые не могут создаваться, удаляться, переименовываться, менять свои параметры в процессе функционирования системы;

*-* множество видов прав доступа;

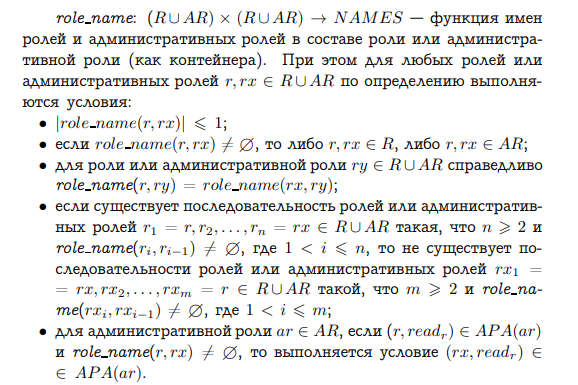
множество видов доступа.

Поскольку в ОССН реализованы три вида прав доступа к сущностям: на чтение, на запись и на выполнение, а также при управлении доступом к сущностям и субъект-сессиям учитывается наличие у каждой из них уникального владельца, имеющего право передавать права доступа к ним другим учетным записям пользователей, то соответственно в рамках МРОСЛ ДП-модели будем использовать виды прав доступа . Кроме того, так как в ОССН при получении субъект-сессиями доступов к сущностям они реализуют одну из двух (или обе) основных возможностей: читать или записывать в сущности данные (например, когда процесс открывает доступ к файлу на чтение или на запись), то в модели заданы следующие виды доступов :

17-18 слайд

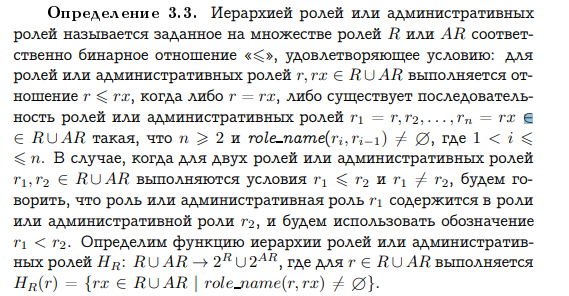


19 слайд



Роли также образуют иерархическую структуру, что было предусмотрено еще в классических ролевых моделях семейства RBAC. Это упрощает администрирование политики безопасности, делает ее более понятной для пользователей и администраторов компьютерной системы. Например, довольно часто ролевая иерархия задается в соответствии с должностной иерархией пользователей компьютерной системы.

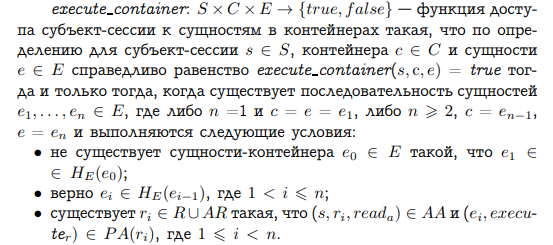
20 слайд

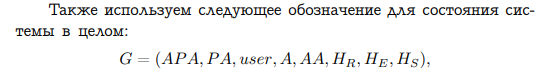


В отличие от других ролевых моделей МРОСЛ ДП-модель предлагает рассматривать роли как аналог сущностей-контейнеров, к которым субъект-сессии могут иметь (через административные роли) права доступа и получать доступы. При этом в рамках МРОСЛ ДП-модели иерархии ролей и административных ролей заданы (по аналогии с иерархией сущностей) не функцией HR, а функцией имен ролей в составе ролей-контейнеров role\_name. Таким образом, право доступа ownr — владелец роли, readr — право получать роль как текущую, просматривать ее параметры, writer — право изменять множество прав доступа роли, executer — право обращаться к ролям, подчиненным данной роли в иерархии ролей (по умолчанию предполагается, что такое право доступа к ролям имеется всегда); доступ reada — получение субъект-сессией роли как текущей, доступ writea — изменение прав доступа роли или состава ролей, подчиненных ей в иерархии. Имеющиеся в ОС привилегии целесообразно задать административными или «обычными» ролями, это обеспечит целостность механизма управления доступом в ОССН и, кроме того, позволит в дальнейшем присваивать ролям привилегиям уровни конфиденциальности и уровни целостности. В результате закладывается основа единого механизма мандатного управления доступом и мандатного контроля целостности для доступов к сущностям, получения в качестве текущих и администрирования ролей субъект-сессиями, с возможностью противодействия в дальнейшем запрещенным информационным потокам по памяти или по времени.

Поскольку в реальной ОССН любой доступ субъект-сессии к сущности сопровождается последовательной проверкой наличия у субъект-сессии прав доступа на выполнение executer ко всем контейнерам, начиная с корневого, в котором содержится сущность, то для удобства и краткости дальнейшего описания таких ситуаций определим функцию:

21 слайд

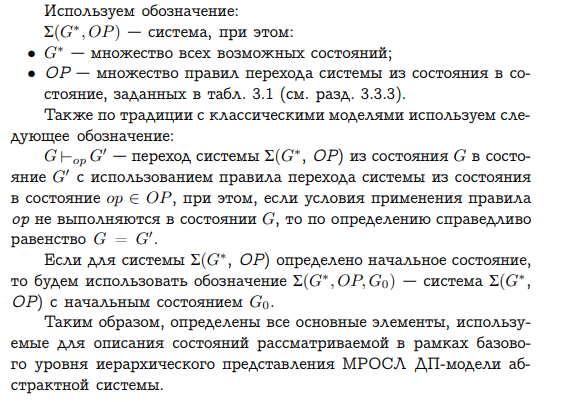




где:

* множества учетных записей пользователей U, сущностей E, субъект-сессий S, прав доступа к сущностям P, доступов субъект-сессий к сущностям A, доступов субъект-сессий к ролям и административным ролям AA;
* функции административных прав доступа к ролям административных ролей APA, прав доступа ролей и административных ролей PA, принадлежности субъект-сессий учетным записям пользователей user, иерархии ролей HR, иерархии сущностей HE, иерархии субъект-сессий HS.

**22 слайд**



23 слайд

**Условия консистентности модели**

Связи между элементами, задающими каждое состояние абстрактной системы в рамках МРОСЛ ДП-модели и определяющими условия функционирования механизма управления доступом реальной ОССН, строятся не произвольным образом. Имеются ограничения, которые по сути являются критериями корректного, консистентного состояния МРОСЛ ДП-модели, а также переходов из состояния в состояние. Кратко будем называть эти ограничения условиями консистентности модели. Таким образом, консистентность состояний и переходов системы на базовом уровне иерархического представления МРОСЛ ДП-модели состоит в выполнении 9 условий.

В условии 1 предполагается, что субъект-сессии не могут иметь доступов друг к другу, что соответствует условиям функционирования реальных защищенных ОССН.

Условие 2 обеспечивает потенциальную возможность получения доступа к роли вне зависимости от ее положения в иерархии ролей. Кроме того, оно не позволяет субъект-сессии, не обладающей соответствующими административными ролями, изменить иерархию ролей. Получая доступ на запись к некоторой роли (дающий возможность изменять множество ее прав доступа), такая субъект-сессия не может удалить роли, подчиненные данной роли

Условие 3 задает общий порядок получения доступов и назначения прав доступа ролей или административных ролей, ролей- «владельцев» к сущностям, субъект-сессиям, ролям или административным ролям.

Условия 4 и 5 задают порядок получения доступа к сущностям, их создания, переименования или удаления, получения параметров типичный для ОССН, при этом эти условия основаны на ролевом управлении доступом и использовании функции execute container.

В условии 6 указываются роли, требуемые для администрирования учетных записей пользователей, что в предшествующих ДП-моделях не допускалось. Типичные для ОССН условия создания или удаления субъект-сессий заданы в условии 7.

Для обеспечения возможности задания в ОССН прав доступа к сущностям, находящимся в архивах или на внешних устройствах (файловая система которых часто не позволяет хранить права доступа или другие параметры механизма управления доступом, например уровни конфиденциальности или целостности, в соответствии с требованиями МРОСЛ ДП-модели), в условии 8 используются косвенные метки сущностей, с помощью которых указывается, что права доступа ролей или административных ролей к сущности (в дальнейшем уровни конфиденциальности или целостности) наследуются от сущности-контейнера, являющейся «точкой монтирования» к файловой системе архива или внешнего устройства. Так как файловая система ОССН не позволяет создавать «жесткие» ссылки на сущности между файловыми системами различных устройств, то в соответствии с условием 2 такая «точка монтирования» определяется однозначно для каждой сущности с косвенной меткой, а, следовательно, однозначно задаются права доступа к ней.

В условиях 9 и 10 используются права доступа индивидуальных административных ролей, задаваемых функцией APA. Такой подход позволяет реализовать ролевое управление доступом — назначение прав доступа учетным записям пользователей только через роли.

24 слайд

**Де-юре правила перехода системы из состояния в состояние**

МРОСЛ ДП-модель, как и большинство классических моделей политик безопасности управления доступом, является автоматной, а значит, после задания состояний моделируемой абстрактной системы должна быть задана функция переходов, которая в модели по традиции определяется через описание правил перехода системы из состояния в состояние.

В рамках базового уровня МРОСЛ ДП-модели задано 31 де-юре правило перехода системы из состояния в состояние, условия и результаты применения которых соответствуют условиям консистентности модели. Эти правила предназначены для формального описания следующих основных функций механизма управления доступом защищенной ОС:

25 слайд

* создание, удаление, переименование, получение или изменение параметров учетных записей пользователей, ролей, административных ролей, сущностей или «жестких» ссылок на них, субъект-сессий;
* получение доступов субъект-сессий к сущностям, ролям или административным ролям;
* изменение прав доступа ролей или административных ролей к сущностям, субъект-сессиям, ролям или административным ролям;
* изменение иерархии сущностей, ролей или административных ролей.

26 слайд

Де-юре правила вида create\_user(x,u), delete\_user(x,u) и get\_user\_attr(x,u,z) позволяют субъект-сессии x создать, удалить или получить параметры учетной записи пользователя u.

27 слайд

Де-юре правила вида access\_read(x, y) и access\_write(x, y) позволяют субъект-сессии x, обладающей доступом на чтение к некоторой роли или административной роли r (текущей роли), содержащей соответствующее право доступа к сущности или административное право доступа к роли или административной роли y, получить к y соответствующий доступ. Де-юре правило delete\_access(x, y, ) позволяет субъект-сессии x, обладающей доступом к сущности или административным доступом к роли или административной роли y, удалить этот доступ.

Де-юре правила вида grant\_rights(x,r,{(y, ): 1 ≤ j ≤k}) и remove\_rights(x, r, {(y, ): 1 ≤ j ≤k}) позволяют субъект-сессии x добавить или удалить соответственно права доступа к сущности y из множества прав доступа (за исключением права доступа владения) роли или административной роли r.

Де-юре правила вида set\_entity\_owner(x,r,r′,y) и set\_subject\_owner(x, r, r′, y) позволяют субъект-сессии x либо изменить, либо задать единственную роль-«владелец» (имеющую право доступа владения) к сущности или субъект-сессии y соответственно с роли или административной роли r на роль или административную роль r'.

28 слайд

Де-юре правила вида grant\_admin\_rights(x, ar, {(r, ): 1 ≤ j ≤k}) и remove\_admin\_rights(x, ar, {(r, ): 11 ≤ j ≤k}) позволяют субъект-сессии x добавить или удалить соответственно права доступа на чтение или запись к роли или административной роли r из множества прав доступа административной роли ar, к которой x должна иметь административный доступ на запись.

Де-юре правила вида create\_object(x, y, yd, name, z), create\_container(x,y,yd,name,z) и delete\_entity(x,y,z) позволяют субъект- сессии x создать или удалить сущность-объект или сущность-контейнер y, входящую в состав сущности-контейнера z, к которой субъект-сессия x должна иметь доступ на запись и обладать текущей ролью или административной ролью, имеющей к z право доступа на выполнение $execute\_r$.

29 слайд

Де-юре правила вида create\_hard\_link(x,y,name,z) и delete\_hard\_link(x, y, name, z) позволяют субъект-сессии x создать или удалить соответственно в составе сущности-контейнера z «жесткую» ссылку на сущность-объект y.

30 слайд

Де-юре правила вида create\_role(x, r, name, rz), create\_hard\_link\_role(x,r,rz), delete\_role(x,r,rz) и delete\_hard\_link\_role(x,r,rz) позволяют субъект-сессии x создать или удалить роль или административную роль r или «жесткую» ссылку на нее (изменить иерархию ролей), входящую в состав роли или административной роли rz, к которой субъект-сессия x должна иметь доступ на запись и не являющейся индивидуальной административной, индивидуальной ролью учетной записи пользователя или общей ролью.

31 слайд

Де-юре правила вида rename\_entity(x, y, old name, name, z) и rename\_role(x, ry, name) позволяют субъект-сессии x переименовать сущность y, входящую в состав сущности-контейнера z, к которой субъект-сессия x должна иметь доступ на запись и обладать текущей ролью или административной ролью, имеющей к ней право доступа на выполнение $execute\_r$, или соответственно переименовать роль или административную роль ry во всех ролях-контейнерах, в которые она входит и к которым субъект-сессия x должна иметь административные доступы на запись.

Де-юре правило вида read\_container(x, y, z) позволяет субъект- сессии x «считать» в сущность-объект z к которой она должна иметь доступ на запись, содержимое сущности-контейнера, роли или административной роли y, к которой x должен иметь права доступа на чтение и выполнение, с учетом прав доступа к сущностям- контейнерам, содержащим y.

Де-юре правила вида get\_entity\_attr(x, y, z), get\_subject\_attr(x, y, z) и get\_role\_attr(x, y, z) позволяют субъект-сессии x «считать» в сущность-объект z (, к которой она должна иметь доступ на запись, атрибуты сущности, субъект-сессии, роли или административной роли y соответственно.

32 слайд

Де-юре правило вида set\_container\_attr(x, y, t) позволяет субъект-сессии x задать сущности-контейнеру y является ли она разделяемой или нет.

Де-юре правило вида create first\_subject(x, u, y, z) позволяет субъект-сессии x с использованием сущности y и учетной записи пользователя u создать от имени u новую субъект-сессию z.

Де-юре правила вида create\_subject(x, y, z) и delete\_subject(x,z) позволяют субъект-сессии x создать или удалить соответственно субъект-сессию z.

Таким образом, полностью определена рассматриваемая в рамках базового уровня иерархического представления МРОСЛ ДП- модели абстрактная система (автомат), а именно заданы элементы, используемые для описания ее состояний, и правила перехода системы из состояния в состояние.

**33 слайд**

При реализации МКЦ в AL ее сущность разделяется на уровни. Чем выше уровень целостности сущности —те выше ее важность для корректного функционирования ОС и тем выше требование доверия к процессу, модифицирующему данную сущность.

МКЦ не допускает модификацию объектов с высоким уровнем целостности субъектам с низким уровнем целостности вне зависимости от мандатных и дискреционных меток

По картинке мы видим, что:

1. если объект с субъектом находятся на одном уровне— то они могут взаимодействовать с обоими правами доступа.

2. Субъект с высоким уровнем целостности может лишь записывать в объект с низким уровнем целостности, и при попытке получения доступа на чтение система понижает доверие к этому пользователю

3. Субъект с низким уровнем целостности может лишь прочитать объект с высоким уровнем целостности, и при попытке получения прав на запись система понижает доверие объекта с высоким уровнем целостности и все субъекты с высоким уровнем целостности теряют возможность взаимодействия с таким файлом

**34 слайд**

Уровни целостности обьектов и субьектов

**35 слайд**

На примере подводной лодки

**36 слайд**

Спасибо за внимание