50 1190 0101

Утвержден

РУСБ.10015-01-УД

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл.	

ОПЕРАЦИОННАЯ СИСТЕМА СПЕЦИАЛЬНОГО НАЗНАЧЕНИЯ «ASTRA LINUX SPECIAL EDITION»

Описание применения

РУСБ.10015-01 31 01

Листов 36

2015

АННОТАЦИЯ

Настоящий документ является описанием применения операционной системы специального назначения «Astra Linux Special Edition» РУСБ.10015-01 (далее по тексту — ОС).

В документе описаны назначение ОС, условия ее применения, описание задачи, приведены входные и выходные данные. Также приведены сведения по получению обновлений ОС.

СОДЕРЖАНИЕ

1. НАЗНАЧЕНИЕ ПРОГРАММЫ

1.1. Назначение

ОС предназначена для построения автоматизированных систем в защищенном исполнении, обрабатывающих информацию, содержащую сведения, составляющие государственную тайну с грифом не выше «совершенно секретно».

1.2. Основные характеристики

В состав ОС входят следующие компоненты:

– ядро ОС;
– средства установки и настройки ОС;
– системные и сервисные утилиты;
– базовые сетевые службы;
– средства организации единого пространства пользователей (ЕПП);
– программы защищенной графической подсистемы;
– средства управления программными пакетами;
– средства резервного копирования и восстановления данных;
– защищенный комплекс программ печати и учета документов;
– защищенный комплекс программ гипертекстовой обработки данных;
– защищенная система управления базами данных;
– защищенный комплекс программ электронной почты;
– пакет офисных программ.

1.3. Возможности

ОС предоставляет следующие возможности:

– установку и функционирование на современных серверах и рабочих станциях на платформах с процессорной архитектурой x86-64, а также поддержку современного периферийного оборудования;
– поддержку основных сетевых протоколов (TCP/IP, DHCP, DNS, FTP, TFTP, SMTP, IMAP, HTTP, NTP, SSH, NFS, SMB);
– организацию сетевого домена с централизованным хранением учетных записей;
– работу с мультимедийными данными;
– работу с реляционными БД;
– работу с электронной почтой;
– работу с гипертекстовыми данными;
– обработку текстовых документов и электронных таблиц различных форматов.

2. УСЛОВИЯ ПРИМЕНЕНИЯ

2.1. Требования к техническим средствам

Для функционирования ОС необходима следующая минимальная конфигурация оборудования:

– аппаратная платформа — процессор с архитектурой x86-64 (AMD, Intel);
– оперативная память — от 1 ГБ;
– объем свободного дискового пространства — от 4 ГБ;
– устройство чтения DVD-дисков;
– стандартный монитор SVGA 15”.

2.2. Совместимость с оборудованием

Штатное, предусмотренное документацией, функционирование ОС обеспечивается только на рекомендованном изготовителем ОС совместимом оборудовании. Перечень рекомендуемого к применению оборудования, а также регламент сертификации на совместимость опубликованы на сайте `http://astra-linux.ru`.

3. ПОРЯДОК ОБНОВЛЕНИЯ ОС

В целях реализации функций безопасности по управлению обновлениями функций СЗИ для ОС предусмотрен плановый выпуск очередных обновлений (новых версий) и выпуск внеочередных обновлений.

3.1. Новая версия ОС

Очередное обновление ОС (новая версия) решает следующий комплекс задач:

– обеспечение поддержки современного оборудования;

– реализация новых функциональных возможностей программных средств (компонент) из состава ОС;

– устранение ошибок и повышение уровня защищенности;

– повышение удобства использования и управления компонентами ОС.

Лицензиаты (потребители) оповещаются о возможности и порядке получения очередного обновления ОС, как с использованием контактной информации, указанной в заключенных ранее лицензионных договорах (дополнениях к лицензионным договорам), так и путем размещения соответствующей информации на сайте разработчика `http://astra-linux.ru`.

Получение очередного обновления (новой версии) ОС осуществляется установленным порядком при заключении соответствующего договора (дополнения к имеющемуся лицензионному договору).

Верификация потребителями очередного обновления ОС (входной контроль) осуществляется посредством подсчета контрольных сумм DVD-дисков. Значения контрольных сумм и порядок их вычисления определены в документе РУСБ.10015-01 30 01 «Операционная система специального назначения «Astra Linux Special Edition». Формуляр».

Дополнительная верификация файлов, входящих в состав очередного обновления ОС СН, осуществляется автоматически средствами создания замкнутой программной среды в соответствии с описанием, приведенным в документе РУСБ.10015-01 95 01-1 «Операционная система специального назначения «Astra Linux Special Edition». Руководство администратора. Часть 1».

3.2. Внеочередное обновление ОС

При получении сведений о наличии в компоненте ОС уязвимости, которая может быть использована для нарушения установленных правил разграничения доступа к информации, настроенной в соответствии с требованиями эксплуатационной документации и/или ограничениями эксплуатации, разработчик ОС описывает организационно-технические мероприятия, устраняющие выявленную уязвимость на объектах эксплуатации.

В случае невозможности определить организационно-технических мероприятия, устраняющие выявленную уязвимость ОС на объектах эксплуатации, либо их неэффективности разработчик выпускает внеочередное обновление.

Источником внеочередных обновлений является официальный сайт разработчика ОС http://astra-linux.ru, на котором в соответствующем разделе http://astra-linux.ru/wiki/index.php/AstraLinux_Wiki размещаются внеочередные обновления, доступные в виде:

1) отдельных инструкций, содержащих сведения об обязательных к проведению при эксплуатации ОС СН организационно-технических мероприятиях;

2) отдельных файлов программ, инструкций по их установке и настройке, а также информации, содержащей сведения о контрольных суммах всех файлов внеочередного обновления;

3) отдельных пакетов программ (комплекта пакетов программ), инструкций по их установке и настройке, а также информации, содержащей сведения о контрольных суммах всех файлов внеочередного обновления;

4) методических указаний по настройке и особенностям эксплуатации ОС СН с установленными внеочередными обновлениями.

Лицензиаты (потребители) оповещаются о возможности и порядке получения внеочередного обновления ОС, как с использованием контактной информации, указанной в заключенных ранее лицензионных договорах (дополнениях к лицензионным договорам), так и путем размещения соответствующей информации на сайте разработчика http://astra-linux.ru.

Верификация внеочередных обновлений осуществляется с помощью контрольных сумм, рассчитанных с использованием программ подсчета контрольных сумм gostsum (для файлов) и gostsum_from_deb (для deb-пакетов) из состава ОС, а также утилиты vimdiff из состава ОС.

Дополнительная верификация файлов, входящих в состав очередного обновления ОС, осуществляется автоматически средствами создания замкнутой программной среды в соответствии с описанием, приведенным в документе РУСБ.10015-01 95 01-1 «Операционная система специального назначения «Astra Linux Special Edition». Руководство администратора. Часть 1»).

Установка и настройка обновлений осуществляется по инструкции, поставляемой совместно с обновлением, или по методическим указаниям, приведенным на сайте разработчика ОС http://astra-linux.ru.

4. ОПИСАНИЕ ЗАДАЧИ

Основная задача, решаемая ОС в процессе своего функционирования, — обеспечение интерфейса для доступа ПО к устройствам вычислительной системы посредством управления устройствами, вычислительными процессами и эффективного распределения вычислительных ресурсов между вычислительными процессами в соответствии с требованиями руководящих документов по обеспечению защиты информации, содержащей сведения, составляющие государственную тайну с грифом не выше «совершенно секретно».

4.1. Классы решаемых задач

Для решения основной задачи функционирования ОС она декомпозируется на следующие классы задач:

– загрузка программ в ОП и управление их выполнением;

– обеспечение многозадачного режима функционирования (одновременного выполнения множества процессов);

– распределение ресурсов вычислительной системы между процессами;

– управление распределением ОП между процессами и организация виртуальной памяти;

– обеспечение доступа к данным на энергонезависимых носителях (НЖМД, оптические диски и пр.), организованным в виде некоторой ФС;

– выполнение по запросу программ низкоуровневых операций (ввод-вывод данных, выделение и освобождение памяти, запуск и завершение программ и т. д.);

– предоставление стандартизованного доступа программ к периферийным устройствам (устройствам ввода-вывода);

– поддержка стеков сетевых протоколов;

– обеспечение многопользовательского режима работы;

– обеспечение пользовательского интерфейса (4.1.1);

– идентификация и аутентификация пользователей (4.1.2);

– организация ЕПП (4.1.3);

– дискреционное разграничение доступа процессов к ресурсам (4.1.4);

– мандатное разграничение доступа процессов к ресурсам (4.1.5);

– организация надежных вычислений (изоляция адресных пространств процессов) (4.1.6);

– обеспечение взаимодействия между процессами (4.1.6);

– регистрация событий (протоколирование) (4.1.7);

– очистка оперативной и внешней памяти (4.1.8);

– контроль целостности (4.1.9);

– создание замкнутой программной среды (4.1.10);

– маркировка документов при выводе на печать (4.1.11);

– обеспечение надежного восстановления (4.1.12);

– обеспечение доступа к БД в соответствии с требованиями для разграничения доступа к информации, содержащей сведения, составляющие государственную тайну с грифом не выше «совершенно секретно» (4.1.13);

– обеспечение доступа к информации через сервер гипертекстовой обработки данных в соответствии с требованиями для разграничения доступа к информации, содержащей сведения, составляющие государственную тайну с грифом не выше «совершенно секретно» (4.1.14);

– обеспечение обмена сообщениями электронной почты в соответствии с требованиями для разграничения доступа к информации, содержащей сведения, составляющие государственную тайну с грифом не выше «совершенно секретно» (4.1.15).

4.1.1. Обеспечение пользовательского интерфейса

Решение задачи обеспечения графического пользовательского интерфейса основано на использовании системы X Window, которая имеет архитектуру «клиент-сервер». X-сервер отвечает за взаимодействие с дисплеем и устройствами ввода. Клиенты соединяются с X-сервером локально (с использованием сокетов) или удаленно (TCP/IP).

Для предотвращения реализации угроз нарушения конфиденциальности и целостности информации в обход мандатного разграничения доступа (4.1.5), в т. ч. с использованием механизмов «копирования-вставки» и «буксировки» (copy-paste и drag-and-drop), для переноса информации из секретного документа (окна) в несекретный в графической системе ОС реализован подход на основе полного разделения в соответствии с мандатным контекстом (сочетанием уровня и категорий). Подобный подход означает, что для каждого мандатного контекста запускается собственный X-сервер, а значит и графический сеанс. При графическом входе в систему пользователю предлагается в специальном диалоге выбрать мандатный контекст из доступных пользователю уровней и/или категорий. Далее графическая сессия будет выполняться в выбранном мандатном контексте. Одновременно пользователем может быть выполнено несколько входов с разными мандатными контекстами. Сессии изолированы, и передача информации между ними невозможна.

Графическая подсистема ОС все же позволяет внутри графической сессии, выполняемой в определенном мандатном контексте, запускать приложения с иным мандатным контекстом. При этом для предотвращения реализации угроз нарушения конфиденциальности и целостности информации в обход мандатного разграничения доступа используется специальный модуль-расширение X-сервера — `XPARSEC`. В модуле используется набор «перехватчиков», предоставляемый встроенным расширением X-сервера — `XACE`. При

получении запросов от клиента «перехватчики» («hooks») XACE передают управление и параметры в XPARSEC, который анализирует аргументы запросов и в соответствии с установленными правилами разграничения доступа разрешает или запрещает выполнение запросов клиента. В ОС мандатный контекст считывается при каждом запросе клиента.

Для обеспечения возможности работы привилегированного клиента (менеджера окон), которому необходимо выполнять некоторые запросы к X-серверу, независимо от мандатного контекста своей метки в специальном файле (/etc/X11/trusted) размещается информация с указанием полного пути запуска. При локальном соединении X-сервер получает PID (идентификатор процесса) клиента, определяет путь запуска и привилегии клиента. Менеджер окон может получать метки окон и на основе реализованного в ОС специального расширения X-протокола выполнять привилегированные операции.

В состав графической подсистемы ОС входит рабочий стол пользователя Fly, интегрированный с внедренными в X-сервер механизмами защиты информации и обеспечивающий отображение:

– мандатного контекста сессии в системном лотке (трей);
– мандатного уровня каждого окна;
– мандатного уровня во всех приложениях рабочего стола;
– запуска приложения с разными мандатными контекстами;
– уровня доверенности окна для локальных и удаленных приложений (в удаленном режиме будут цветная рамка, соответствующая мандатной метке, и пунктирная).

Графическая подсистема ОС готова к работе с соблюдением мандатного разграничения доступа непосредственно после установки ОС без проведения дополнительных настроек.

4.1.2. Идентификация и аутентификация пользователей

Решение задачи идентификации и аутентификации пользователей в ОС основывается на использовании механизма PAM, который представляет собой набор разделяемых библиотек (модулей), с помощью которых администратор может организовать процедуру аутентификации (подтверждение подлинности) пользователей прикладными программами. Каждый модуль реализует собственный механизм аутентификации. Изменяя набор и порядок следования модулей, можно построить сценарий аутентификации. Подобный подход позволяет изменять процедуру аутентификации без изменения исходного кода и повторного компилирования PAM. Сценарии аутентификации описываются в конфигурационных файлах.

Если ОС не настроена для работы в ЕПП (4.1.3), то аутентификация осуществляется с помощью локальной БД пользователей. При использовании ЕПП аутентификация пользователей осуществляется централизованно по протоколу Kerberos.

В ОС реализована возможность хранения аутентификационной информации пользователей, полученной с использованием хэш-функции по ГОСТ Р 34.11-94 и по ГОСТ Р 34.11-2012.

4.1.3. Организация единого пространства пользователей

Решение задачи организации ЕПП (создание домена) обеспечивает:

– сквозную аутентификацию в сети;

– централизацию хранения информации об окружении пользователей;

– централизацию хранения настроек системы защиты информации на сервере;

– интеграцию в домен защищенных серверов СУБД (4.1.13), электронной почты (4.1.15), гипертекстовой обработки данных (4.1.14) и печати (4.1.11);

– централизованную настройку правил регистрации событий безопасности в рамках домена;

– централизованный учет подключаемых устройств.

Сетевая аутентификация и централизация хранения информации об окружении пользователя подразумевает использование двух основных механизмов: поддержки кросс-платформенных серверных приложений для обеспечения безопасности NSS и PAM (см. 4.1.2).

Для реализации удаленной аутентификации используется служба каталогов LDAP в качестве источника данных для базовых системных служб на основе механизмов NSS и PAM. В результате вся служебная информация пользователей сети может располагаться на выделенном сервере в распределенной гетерогенной сетевой среде. Добавление новых сетевых пользователей в этом случае производится централизованно на сервере службы каталогов. Сетевые службы, поддерживающие возможность аутентификации пользователей, могут вместо локальных учетных записей использовать каталог LDAP. Администратор может централизованно управлять конфигурацией сети, включая разграничение доступа к сетевым службам.

Благодаря предоставлению информации LDAP в иерархической древовидной форме, разграничение доступа в рамках службы каталогов LDAP может быть основано на введении доменов. В качестве домена в данном случае будет выступать поддерево службы каталогов LDAP. Служба каталогов LDAP позволяет разграничивать доступ пользователей к разным поддеревьям каталога, хотя по умолчанию в ОС реализуется схема одного домена.

Сквозная доверенная аутентификация реализуется технологией Kerberos.

Централизация хранения информации об окружении пользователей подразумевает так же и централизованное хранение домашних каталогов пользователей. Для этого используется сетевая защищенная ФС CIFS.

В среде ОС пользователю поставлен в соответствие ряд атрибутов, характеризующих его мандатные права. Концепция ЕПП подразумевает хранение системной информации о пользователе (включая доступные мандатные уровни и категории) централизованно. В данном случае вся информация хранится в службе каталогов LDAP.

Информация о мандатных атрибутах пользователей (4.1.5) хранится локально в соответствующих конфигурационных файлах. При изменении конфигурации системы для использования в сетевом контексте мандатные права пользователей должны переместиться вслед за окружением пользователя (идентификаторы пользователей, групп, домашние каталоги и пр.) в службу каталогов LDAP. Доступ к мандатным атрибутам пользователей осуществляется с использованием программного интерфейса подсистемы безопасности PARSEC. Данный интерфейс позволяет получить из соответствующего конфигурационного файла информацию об источнике данных для мандатных СЗИ системы. По умолчанию используются локальные текстовые файлы. При работе ОС в сетевом контексте в качестве источника данных выступает служба каталогов LDAP. Переключение контекста производится путем правки соответствующего конфигурационного файла.

Для управления ЕПП в ОС реализована служба ALD, которая является надстройкой над технологиями LDAP, Kerberos, CIFS и обеспечивает автоматическую настройку всех необходимых файлов конфигурации служб, реализующих перечисленные технологии, а также предоставляет интерфейс управления и администрирования.

4.1.4. Дискреционное разграничение доступа процессов к ресурсам

В ОС механизм дискреционного разграничения доступа обеспечивает проверку дискреционных ПРД, формируемых в виде базовых ПРД ОС семейства Linux, формируемых в виде идентификаторов субъектов (идентификатор пользователя (UID) и идентификатор группы (GID), имеющих доступ к объекту (чтение, запись, исполнение). Кроме того, для формирования дискреционных ПРД в ОС используются списки контроля доступа (ACL) и механизм системных привилегий ОС семейства Linux.

В состав ОС входят защищенные комплексы программ: СУБД, электронной почты и гипертекстовой обработки данных.

В защищенных комплексах программ электронной почты и гипертекстовой обработки данных защищаемыми объектами являются объекты ФС. Таким образом, дискреционное разграничение доступа к ним обеспечивается так же, как и к прочим объектам ФС.

4.1.5. Мандатное разграничение доступа процессов к ресурсам

Решение задачи мандатного разграничения доступа процессов к ресурсам основано на реализации соответствующего механизма в ядре ОС. При этом, принятие решения о запрете или разрешении доступа субъекта к объекту принимается на основе типа операции

(чтение, запись, исполнение), мандатного контекста безопасности, связанного с каждым субъектом, и мандатной метки, связанной с объектом.

Правила принятия решения могут быть записаны следующим образом. Пусть контекст безопасности субъекта содержит уровень `L0` и категории `C0`, а мандатная метка объекта содержит уровень `L1` и категории `C1`. В ОС определены следующие операции сравнения уровней и категорий:

– уровень `L0` меньше уровня `L1` (`L0<L1`), если численное значение `L0` меньше численного значения `L1`;

– уровень `L0` равен уровню `L1` (`L0=L1`), если численные значения `L0` и `L1` совпадают;

– категории `C0` меньше категорий `C1` (`C0<C1`), если все биты набора `C0` являются подмножеством набора бит `C1`;

– категории `C0` равны категориям `C1` (`C0=C1`), если значения `C0` и `C1` совпадают.

Таким образом, в механизме мандатного разграничения доступа действуют следующие правила:

– операция записи разрешена, если `L0=L1` и `C0=C1`;

– операция чтения разрешена, если `L0>=L1` и `C0>=C1`;

– операция исполнения разрешена, если `L0>=L1` и `C0>=C1`.

В остальных случаях анализируются полномочия и тип мандатной метки. Тип метки может использоваться для того, чтобы изменять ее эффективное действие. Ненулевой тип метки может быть установлен только привилегированным процессом.

Механизм мандатного разграничения доступа затрагивает следующие подсистемы:

– механизмы IPC;

– стек TCP/IP (IPv4);

– слой виртуальной ФС;

– ФС Ext2/Ext3/Ext4;

– сетевая защищенная ФС CIFS;

– ФС proc, tmpfs.

При создании субъектом объекта, относящегося к любой из вышеприведенных подсистем, объект наследует метку на основе мандатного контекста безопасности процесса. При этом тип метки устанавливается в 0. Если ФС поддерживает только нулевые метки (например, VFAT), то на ней невозможно создание объектов с меткой, отличной от нулевой.

В ОС система привилегий Linux, предназначенная для передачи отдельным пользователям прав выполнения определенных административных действий, расширена следующими привилегиями, относящимися к подсистеме безопасности PARSEC и обеспечивающими работу с механизмом мандатного разграничения доступа:

– PARSEC_CAP_SIG — право посылать сигналы процессам, игнорируя дискреционные и мандатные права;

– PARSEC_CAP_SETMAC — право изменять мандатную метку и устанавливать другие привилегии;

– PARSEC_CAP_CHMAC — право менять мандатные метки файлов;

– PARSEC_CAP_AUDIT — право управления политикой аудита;

– PARSEC_CAP_READSEARCH — право игнорировать мандатную политику при чтении и поиске файлов (но не при записи);

– PARSEC_CAP_PRIV_SOCK — право создавать привилегированный сокет и менять его мандатную метку. Привилегированный сокет позволяет осуществлять сетевое взаимодействие, игнорируя мандатную политику;

– PARSEC_CAP_UPDATE_ATIME — право изменять время доступа к файлу;

– PARSEC_CAP_IGNMACLVL — право игнорировать мандатную политику по уровням;

– PARSEC_CAP_IGNMACCAT — право игнорировать мандатную политику по категориям;

– PARSEC_CAP_FILE_CAP — право устанавливать привилегии на файлы;

– PARSEC_CAP_CAP — право устанавливать любой непротиворечивый набор привилегий для другого процесса;

– PARSEC_CAP_MAC_SOCK — право смены мандатной точки соединения.

Привилегии наследуются процессами от своих родителей. Процессы, запущенные от имени администратора, независимо от наличия у них привилегий имеют возможность осуществлять все перечисленные привилегированные действия.

В качестве основной сетевой защищенной ФС используется CIFS, которая является расширением SMB, поддерживает атрибуты ФС UNIX и имеет ограниченную поддержку расширенных атрибутов. Кроме того, она широко распространена и работает в гетерогенных сетях (поддерживается многими ОС). Поддерживает аутентификацию средствами PAM и Kerberos (см. 4.1.2).

Сетевые соединения могут рассматриваться как IPC, поэтому должны подвергаться мандатному контролю доступа. Для этого в сетевые пакеты протокола IPv4 в соответствии со стандартом RFC1108 внедряются мандатные метки, соответствующие метке объекта — сетевое соединение (сокет). При этом метка сокета наследуется от субъекта (процесса). Прием сетевых пакетов подчиняется мандатным ПРД. Следует отметить, что метка сокета может иметь тип, позволяющий создавать сетевые сервисы, принимающие соединения с любыми уровнями секретности.

В рамках стандарта RFC1108 метка снабжается классом 0xAB (Unclassified), при этом последующий битовый список (последовательность байт, в которых младший бит ука-

зывает на наличие следующего байта в потоке) опции представляет собой упакованную в соответствии со стандартом структуру мандатного контекста, где уровень занимает 8 бит, а категории — 64 бита (порядок байт — от младших к старшим). Последние (старшие) нулевые биты в соответствии со стандартом могут быть отброшены.

Отсутствие метки на объекте доступа является синонимом нулевой мандатной метки. Таким образом, ядро ОС, в которой все объекты и субъекты доступа имеют нулевой уровень, функционирует аналогично стандартному ядру ОС Linux.

Для ряда сетевых сервисов (например, сервер LDAP, DNS, Kerberos) необходимо обеспечить возможность их работы с клиентами, имеющими разный мандатный контекст безопасности.

Обеспечение мандатного разграничения доступа в защищенных комплексах программ гипертекстовой обработки данных и электронной почты реализовано на основе программного интерфейса библиотек подсистемы безопасности PARSEC. На серверах комплексов программ гипертекстовой обработки данных и электронной почты при обработке запросов на соединение выполняется получение мандатного контекста соединения, унаследованного от субъекта (процесса). Сокет сервера, ожидающий входящих запросов на соединение, работает в контексте процесса, имеющего привилегию для приема соединений с любыми уровнями секретности.

После установки соединения и успешного прохождения процедуры идентификации и аутентификации пользователя процесс сервера, обрабатывающий запросы пользователя, переключается в контекст безопасности пользователя, сбрасывает привилегии, обрабатывает запросы пользователя и завершается.

В комплексе программ гипертекстовой обработки данных (4.1.14) пользователь получает доступ к ресурсам, являющимся объектами ФС. Комплекс программ электронной почты (4.1.15) использует технологию maildir, обеспечивающую хранение почтовых сообщений в виде отдельных объектов ФС. Создаваемые файлы почтовых сообщений маркируются мандатными метками, унаследованными от процесса-создателя. Таким образом, в обоих комплексах программ ресурсы, к которым осуществляется доступ от имени серверных процессов, обрабатывающих запросы пользователей, являются объектами ФС. Следовательно, доступ к защищаемым ресурсам при приеме и обработке запросов пользователей в процессе функционирования серверов комплексов программ гипертекстовой обработки данных и электронной почты подчиняется мандатным ПРД.

4.1.6. Изоляция адресных пространств процессов

Решение задачи изоляции адресных пространств процессов основано на архитектуре ядра ОС, которое обеспечивает для каждого процесса в системе собственное изолированное адресное пространство. Данный механизм изоляции основан на страничном

механизме защиты памяти, а также механизме трансляции виртуального адреса в физический, поддерживаемый модулем управления памятью. Одни и те же виртуальные адреса (с которыми и работает процессор) преобразуются в разные физические для разных адресных пространств. Процесс не может несанкционированным образом получить доступ к пространству другого процесса, т. к. непривилегированный пользовательский процесс лишен возможности работать с физической памятью напрямую.

Механизм разделяемой памяти является санкционированным способом получить нескольким процессам доступ к одному и тому же участку памяти и находится под контролем дискреционных и мандатных ПРД (см. 4.1.4 и 4.1.5).

Адресное пространство ядра защищено от прямого воздействия пользовательских процессов с использованием механизма страничной защиты. Страницы пространства ядра являются привилегированными и доступ к ним из непривилегированного кода вызывает исключение процессора, которое обрабатывается корректным образом ядром ОС. Единственным санкционированным способом доступа к ядру ОС из пользовательской программы является механизм системных вызовов, который гарантирует возможность выполнения пользователем только санкционированных действий.

4.1.7. Регистрация событий

В ОС реализована расширенная подсистема протоколирования, осуществляющая регистрацию событий в двоичные файлы с использованием сервиса `parlogd`.

В библиотеках подсистемы безопасности PARSEC реализован программный интерфейс для протоколирования событий с использованием расширенной подсистемы протоколирования. Данный программный интерфейс применен для регистрации событий в СУБД PostgreSQL (4.1.13.3).

4.1.8. Очистка оперативной и внешней памяти

Решение задачи очистки ОП основано на архитектуре ядра ОС, которое гарантирует, что обычный непривилегированный процесс не получит данные чужого процесса, если это явно не разрешено ПРД. Это означает, что средства взаимодействия между процессами контролируются с помощью ПРД, и процесс не может получить неочищенную память (как оперативную, так и дисковую).

Решение задачи очистки памяти на внешних носителях основано на реализации механизма, который очищает неиспользуемые блоки ФС непосредственно при их освобождении. Работа названного механизма снижает скорость выполнения операций удаления и усечения размера файла. Механизм является настраиваемым и позволяет обеспечить работу ФС ОС (Ext2/Ext3/Ext4) в одном из следующих режимов:

– данные любых удаляемых/урезаемых файлов в пределах заданной ФС предва-

рительно очищаются маскирующей последовательностью;

– данные ФС освобождаются обычным образом (без предварительного маскирования).

Режим работы ФС может быть выбран администратором ОС и задан в виде параметра монтирования ФС.

Кроме того, в ОС реализован механизм включения очистки активных разделов страничного обмена.

4.1.9. Контроль целостности

Решение задач контроля целостности основано на использовании библиотеки `libgost`, в которой реализованы функции хэширования в соответствии с ГОСТ Р 34.11-94, ГОСТ Р 34.11-2012 с длиной хэш-кода 256 бит и ГОСТ Р 34.11-2012 с длиной хэш-кода 512 бит. Данная библиотека используется в средствах контроля целостности дистрибутива и средствах контроля целостности ФС.

Контроль целостности дистрибутива обеспечивается методом расчета его контрольной суммы и сравнения полученного значения с эталонным значением контрольной суммы.

Контроль целостности ОС, прикладного ПО и СЗИ обеспечивается набором программных средств, который предоставляет возможность периодического (с использованием системного планировщика заданий `cron`) вычисления контрольных сумм файлов и соответствующих им атрибутов с последующим сравнением вычисленных значений с эталонными. В указанном наборе программных средств реализовано использование библиотеки `libgost` и контроль целостности связанных с файлами атрибутов расширенной подсистемы безопасности PARSEC (мандатных атрибутов и атрибутов расширенной подсистемы протоколирования).

4.1.10. Создание замкнутой программной среды

Средства создания замкнутой программной среды предоставляют возможность внедрения цифровой подписи в исполняемые файлы формата ELF, входящие в состав устанавливаемого СПО и в расширенные атрибуты файловой системы.

Механизм контроля целостности исполняемых файлов и разделяемых библиотек формата ELF при запуске программы на выполнение реализован в модуле ядра ОС `digsig_verif`, который является не выгружаемым модулем ядра Linux, и может функционировать в одном из следующих режимов:

1) исполняемым файлам и разделяемым библиотекам с неверной ЭЦП, а также без ЭЦП загрузка на исполнение запрещается (штатный режим функционирования);

2) исполняемым файлам и разделяемым библиотекам с неверной ЭЦП, а также

без ЭЦП загрузка на исполнение разрешается, при этом выдается сообщение об ошибке проверки ЭЦП (режим для проверки ЭЦП в СПО);

3) ЭЦП при загрузке исполняемых файлов и разделяемых библиотек не проверяется (отладочный режим для тестирования СПО).

Механизм контроля целостности файлов при их открытии на основе ЭЦП в расширенных атрибутах файловой системы также реализован в модуле ядра ОС `digsig_verif` и может функционировать в одном из следующих режимов:

1) запрещается открытие файлов, поставленных на контроль файлов, с неверной ЭЦП или без ЭЦП;

2) открытие файлов, поставленных на контроль файлов, с неверной ЭЦП или без ЭЦП разрешается, при этом выдается сообщение об ошибке проверки ЭЦП (режим для проверки ЭЦП в в расширенных атрибутах файловой системы);

3) ЭЦП при загрузке исполняемых файлов и разделяемых библиотек не проверяется.

4.1.11. Маркировка документов при выводе на печать

Решение задачи маркировки документов при выводе на печать основано на использовании в ОС защищенного сервера печати CUPS, который обеспечивает маркировку выводимых на печать документов. Мандатные атрибуты автоматически связываются с заданием для печати на основе мандатного контекста получаемого сетевого соединения. Вывод на печать документов без маркировки субъектами доступа, работающими в ненулевом мандатном контексте, невозможен.

Для разрешения серверу CUPS обрабатывать задания печати, формируемые в ненулевом мандатном контексте, необходимо от имени администратора выполнить определенные действия, определяющие возможный мандатный контекст, в котором могут формироваться задания для печати на конкретном принтере.

Маркировка документов осуществляется на основе следующих модифицируемых файлов шаблонов:

– файл шаблона, содержащий информацию об атрибутах маркировки и их положении на странице при печати документа;

– файл шаблона, содержащий информацию об атрибутах маркировки оборота последнего листа документа и их положении на странице при печати пяти и менее экземпляров документа;

– файл шаблона, содержащий информацию об атрибутах маркировки оборота последнего листа документа и их положении на странице при печати более пяти экземпляров документа.

4.1.12. Обеспечение надежного восстановления

Для решения задачи надежного восстановления в результате сбоев и отказов оборудования в ОС реализованы:

– автоматическое выполнение в процессе перезагрузки после сбоя программы проверки и восстановления ФС;

– средства резервного копирования и восстановления ОС;

– средства резервного копирования и восстановления СУБД.

Более подробная информация приведена в документах РУСБ.10015-01 95 01-1 «Операционная система специального назначения «Astra Linux Special Edition». Руководство администратора. Часть 1» и РУСБ.10015-01 97 01-1 «Операционная система специального назначения «Astra Linux Special Edition». Руководство по КСЗ. Часть 1».

4.1.13. Обеспечение доступа к БД

Решение задачи обеспечения доступа к БД основано на использовании защищенного комплекса программ СУБД на основе объектно-реляционной СУБД PostgreSQL, в котором на низком уровне данные хранятся в отношениях (таблицах, видах), и доступ к данным разграничивается в понятиях реляционной СУБД.

Данные в реляционной БД хранятся в отношениях (таблицах), состоящих из строк и столбцов. При этом единицей хранения и доступа к данным является строка, состоящая из полей, идентифицируемых именами столбцов. Кроме таблиц, существуют другие типы объектов БД (виды, процедуры и пр.), которые предоставляют доступ к данным, хранящимся в таблицах.

Таким образом, в защищенном комплексе программ СУБД определены типы объектов, с каждым из которых ассоциируется определенный набор типов доступа (возможных операций). Для каждого объекта явно задается список разрешенных для каждого из поименованных субъектов БД (пользователей, групп или ролей) ACL. И в дальнейшем при разборе запроса к БД осуществляется проверка возможности предоставления доступа субъекта к объекту типа, соответствующего запросу.

4.1.13.1. Дискреционное разграничение доступа в защищенной СУБД

В общем случае отдельная строка таблицы не является однозначно идентифицируемым объектом (каждая строка идентифицируется только набором содержимого своих полей, но без специальных действий, например создания первичного ключа или физического уникального идентификатора строки в БД, такая идентификация не является уникальной), и соответственно дискреционные ПРД к ней применены быть не могут. Столбцы могут являться объектами дискреционного разграничения доступа, поскольку могут быть однозначно идентифицированы по составному имени объекта и столбца (имя столбца внутри объекта является уникальным).

В рамках дискреционных ПРД определены следующие операции над таблицами и хранящимися в них данными:

– SELECT — чтение данных из таблицы;

– INSERT — вставка новых данных в таблицу;

– DELETE — удаление некоторых/всех данных в таблице;

– UPDATE — изменение данных в таблице;

– REFERENCES — использование данных таблицы для внешних ключей;

– TRIGGER — создание и назначение для таблицы триггеров;

– TRUNCATE — очистка таблицы (удаление всех данных).

Для объектов «вид» (способ организации предварительно подготовленных запросов) определены следующие операции:

– SELECT — чтение данных из вида;

– INSERT — вставка новых данных в вид;

– DELETE — удаление некоторых/всех данных в виде;

– UPDATE — изменение данных в виде.

Для объектов «последовательность» (способ получения уникальных значений — счетчик) определены следующие операции:

– SELECT — чтение значения счетчика;

– UPDATE — установка значения счетчика;

– USAGE — выполнение функций манипулирования счетчиком.

Для объектов «база данных» (способ организации области данных, содержащих все остальные объекты, определенные в СУБД) определены следующие операции:

– CREATE — создание БД;

– CONNECT — установка соединения с БД;

– TEMPORARY/TEMP — создание временных таблиц в БД.

Для объектов «функция» (программный код манипулирования данными на сервере) определена операция EXECUTE (выполнение функции).

Для объектов «язык» (язык написания функций на сервере) определена операция USAGE (использование языка для написания функций).

Для объектов «схема» (способ организации объектов в пределах отдельной БД) определены следующие операции:

– CREATE — создание объектов в указанной схеме;

– USAGE — использование объектов указанной схемы.

Для объектов «табличное пространство» (способ организации БД в ФС ОС) определена операция CREATE (создание объектов в указанном табличном пространстве).

Для контроля выполнения всех перечисленных операций дискреционных ПРД су-

ществуют соответствующие права доступа. Право на предоставление прав доступа к объектам не может быть предоставлено другим пользователям и доступно только администратору СУБД (при соответствующих настройках сервера может быть предоставлено и владельцу объекта).

Кроме рассмотренных (делегируемых) прав доступа, существует ряд прав, которые всегда принадлежат владельцам объектов и администраторам СУБД. Эти права не могут быть делегированы или отменены средствами СУБД. К таким правам относятся: удаление и модификация объекта и назначение пользователям делегируемых прав доступа к объектам.

Сразу же после создания объекта только его владелец и администраторы СУБД могут использовать его каким-либо образом. Для того чтобы с этим объектом могли работать другие пользователи, владелец объекта или администратор СУБД должен явно предоставить им соответствующие дискреционные права доступа.

Модификация метаданных возникает каждый раз при изменении структуры БД, что включает в себя создание, модификацию и удаление объектов БД.

Разграничение доступа к перечисленным операциям на уровне СУБД так же реализуется применением дискреционных ПРД. Для этого используется право владения объектом, право на создание объектов. Право владения объектом предоставляет владельцу объекта возможность модифицировать и удалять объект. Как правило, владельцем является создатель объекта или администратор ОС (администратор СУБД). Право на создание (`CREATE`) существует к объектам БД, являющихся контейнерами для других объектов, а именно: непосредственно сама БД, схема, табличное пространство.

При выполнении любого запроса пользователя (субъекта БД) к защищаемому ресурсу (объекту БД) выполняется дискреционное разграничение доступа на основе установленных пользователю прав. Для каждой выполняемой операции производится проверка наличия права у пользователя на выполнение данной конкретной операции.

Дискреционные ПРД применяются после разбора запроса пользователя и построения плана его выполнения.

Дискреционные ПРД к столбцам объекта применяются только при отсутствии явного разрешения на доступ к самой таблице. Таким образом, права доступа к объекту являются доминирующими. При этом, при отсутствии явно заданных прав на объект нельзя сказать определенно о предоставлении доступа до тех пор, пока не будут проверены права на столбцы объекта.

В СУБД PostgreSQL параметр конфигурации `ac_enable_trusted_owner` позволяет администратору запретить владельцам объектов передавать права на доступ к ним другим пользователям СУБД. В случае установки значения этой переменной конфигурации

в FALSE распределение прав доступа к объектам БД разрешено только администраторам СУБД.

Параметр конфигурации ac_enable_truncate позволяет администратору запретить владельцам объектов и любым пользователям, обладающим соответствующим правом TRUNCATE, выполнять удаление всех записей из таблиц. В случае установки значения этой переменной конфигурации в FALSE, выполнение команды TRUNCATE запрещено всем пользователям.

4.1.13.2. Мандатное разграничение доступа в защищенной СУБД

В основе мандатного механизма разграничения доступа лежит управление доступом к защищаемым ресурсам БД на основе иерархических и не иерархических меток доступа. Это позволяет реализовать многоуровневую защиту с обеспечением разграничения доступа пользователей к защищаемым ресурсам БД и управление потоками информации. В качестве иерархических и не иерархических меток доступа при использовании СУБД в ОС используются метки конфиденциальности или метки безопасности ОС.

СУБД PostgreSQL не имеет собственного механизма назначения, хранения и модификации меток пользователей и использует для этого механизмы ОС.

В реляционной модели в качестве структуры, обладающей меткой, необходимо выбрать кортеж, поскольку именно на этом уровне детализации осуществляются операции чтения-записи информации в СУБД. При этом местом хранения метки может быть выбран только сам кортеж, так метка будет неразрывно связана с данными, содержащимися в кортеже. Кроме этого, метка так же может быть определена для таких объектов БД, к которым применимы виды доступа на чтение-запись данных, а именно таблицы и виды. В этом случае метки объектов располагаются в записи системной таблицы, непосредственно описывающей защищаемый объект. Так как мандатный контроль доступа может быть определен только для видов доступа на чтение и на запись информации, все множество операций с данными в защищаемых объектах приводится к ним следующим образом:

– INSERT — доступ на запись;

– UPDATE, DELETE — последовательное выполнение доступа на чтение и запись информации;

– SELECT — доступ на чтение.

При обращении пользователя к БД определяются его допустимый диапазон меток и набор специальных мандатных атрибутов. Если пользователю не присвоена метка, то он получает по умолчанию нулевую метку, соответствующую минимальному уровню доступа. Максимальная метка определяется по метке, заданной при регистрации пользователя в ОС. Поскольку сервер БД так же может иметь метку, при превышении метки пользователя метки сервера ему будут разрешены только операции чтения. Текущая метка пользователя

определяется по установленному соединению и может быть установлена в пределах его допустимого диапазона мандатных атрибутов при наличии соответствующей привилегии.

Применение мандатных ПРД осуществляется на уровне доступа к объектам БД и на уровне доступа непосредственно к данным (на уровне записей).

Проверка мандатных прав доступа к таблицам и видам осуществляется одновременно с проверкой дискреционных прав доступа к ним, после разбора и построения плана запроса, непосредственно перед его выполнением, когда определены все необходимые для проверки данные и проверяемые объекты. Таким образом, доступ предоставляется только при одновременном санкционировании дискреционным ПРД.

Проверка мандатных прав доступа к записям таблиц осуществляется в процессе выполнения запроса при последовательном или индексном сканировании данных.

Все записи, помещаемые в таблицы, для которых установлена защита на уровне записей, наследуют текущую метку пользователя. Обновляемые записи сохраняют свою метку при изменении. Доступ к существующим записям и возможность их обновления и удаления определяются установленными мандатными правилами.

Для администратора СУБД предусмотрены системные привилегии игнорирования мандатного разграничения доступа. Только таким образом можно производить регламентные работы с БД (например, восстановление резервной копии), т. к. это требует установки меток данных, сохраненных ранее.

В связи с тем, что в PostgreSQL объектами защиты являются столбцы, выполнена реализация мандатных ПРД для столбцов объектов. В этом случае метки столбцов объектов так же располагаются в записи соответствующей системной таблицы, непосредственно описывающей защищаемый столбец. Мандатные ПРД столбцов и самого объекта не могут быть применены одновременно. Режим применения мандатных ПРД только к самому объекту или только к его столбцам может быть задан для каждого объекта в отдельности. Защита на уровне записей может использоваться в любом случае.

В ОС каждый пользователь может иметь множество меток, которое задается минимальной и максимальной метками диапазона. Чтобы поддержать эту модель в СУБД PostgreSQL каждой сессии пользователя назначается три метки: максимальная, минимальная и текущая. Их начальная инициализация осуществляется по следующему алгоритму:

– после прохождения пользователем стандартной процедуры аутентификации сервер считывает из ОС значения максимальной и минимальной меток пользователя и принимает их как максимальную и минимальную метки сессии. При этом, если запись о метках для пользователя не найдена, то максимальная и минимальная метки принимаются равными нулю. Следовательно, пользователи, зарегистрированные только на сервере СУБД PostgreSQL и не имеющие учетной записи на сервере ОС,

всегда имеют минимальный уровень доступа к информации;

– если параметр конфигурации `ac_ignore_socket_maclabel` установлен в `FALSE`, считывается метка входящего соединения, и, если она попадает в диапазон меток, считанных из ОС, то максимальная метка сессии устанавливается равной метке входящего соединения. При этом, если минимальная метка такого пользователя в ОС сервера выше метки входящего соединения, то он вообще не будет допущен к работе с БД;

– если параметр конфигурации `ac_ignore_server_maclabel` установлен в `FALSE`, то считывается метка серверного процесса и, если она не совместима с максимальной меткой сессии, то процесс аутентификации прерывается;

– текущей меткой сессии становится максимальная метка сформированного таким образом диапазона.

Если на любом из этих этапов возникает ситуация с несовместимостью меток или выходом за пределы диапазона, то процесс аутентификации клиента прерывается и доступ к БД блокируется.

Если пользователь имеет мандатный атрибут `ac_capable_setmac`, то он может изменять свою текущую мандатную метку в диапазоне от минимальной до максимальной.

СУБД PostgreSQL предоставляет пользователям возможность создавать функции (и, следовательно, триггеры), указывая при этом, будут ли они выполняться с уровнем доступа пользователя, прямо или косвенно вызвавшего функцию (`SECURITY INVOKER`), или с уровнем доступа пользователя, создавшего эту функцию (`SECURITY DEFINER`). При этом в понятие «уровня доступа» входят как дискреционный уровень доступа, так и мандатный, который в данном случае определяется текущими мандатными атрибутами пользователя СУБД, вызвавшего или создавшего функцию, соответственно. При этом метки текущей сессии пользователя, вызвавшего функцию, не изменяются.

Следует учитывать, что:

– при определении функции как `SECURITY DEFINER` она будет всегда вызываться с переустановкой мандатных атрибутов на атрибуты создавшего ее пользователя;

– при определении функции как `SECURITY INVOKER` она всегда будет выполняться без изменения текущего значения мандатных атрибутов;

– при вызове функции в качестве триггера выполняются следующие правила в дополнение к указанным:

– перед вызовом в качестве триггера встроенной в СУБД функции к текущим мандатным атрибутам всегда добавляются флаги `ac_capable_ignmaclvl` и `ac_capable_ignmaccat`, чтобы обеспечить полноценную проверку ссылочной целостности БД;

– перед вызовом в качестве триггера не встроенной функции в качестве текущих мандатных атрибутов всегда устанавливаются мандатные атрибуты пользователя, запустившего данную сессию (соединение) (т. е. пользователя с именем `SESSION_USER`). Это необходимо, чтобы предотвратить получение функцией-триггером пользователя с низким уровнем доступа высоких привилегий в случае каскадного вызова триггеров.

После возврата управления из функции значения текущих мандатных атрибутов всегда восстанавливаются в исходные (до вызова функции) значения.

Функции, написанные на языках низкого уровня, после их подключения имеют полный доступ ко всем внутренним структурам сервера СУБД PostgreSQL и могут произвольно их модифицировать. Кроме этого, поскольку они выполняются в рамках процесса сервера, они имеют соответствующие права доступа к объектам ОС в среде функционирования сервера. Именно поэтому права пользователя `postgres`, под которым запускается сервер, необходимо свести к необходимому минимуму, минуя какой-либо контроль с его стороны (включая текущие мандатные атрибуты).

При наличии мандатных меток на сам объект, его столбец и непосредственно строку возможны следующие варианты использования мандатных ПРД (на примере таблиц):

– метки отсутствуют — мандатные ПРД не применяются. В этом случае метка объекта не установлена, метки столбцов не установлены, а сам объект создан без защиты строк. СУБД функционирует в штатном режиме защиты с использованием только дискреционных ПРД;

– метками защищаются только записи. Метка объекта не установлена, метки столбцов не установлены, а сам объект создан с защитой строк. Дискреционные ПРД применяются перед выполнением запроса. Мандатные ПРД применяются только на уровне записей. Создание записей разрешено всем субъектам, при этом записи наследуют метку субъекта. Операции чтения и модификации осуществляются над множествами записей, доступных субъекту по мандатным ПРД. Проверка мандатных ПРД осуществляется после успешного применения дискреционных ПРД, нарушение безопасности не возникает;

– метками защищается только объект. Метка объекта установлена, метки столбцов не установлены, а сам объект создан без защиты строк. Мандатные ПРД применяются только на уровне объекта, все данные, содержащиеся в объекте, рассматриваются, как имеющие метку объекта. Создание записей разрешено субъектам с метками, над которыми доминирует метка объекта, при этом записи наследуют метку субъекта. Операции чтения и модификации осуществляются по мандатным ПРД к объекту. Мандатные ПРД применяются только в случае успешной проверки дис-

крецонных ПРД, которые к столбцам объекта применяются только при отсутствии явного разрешения на доступ к самой таблице;

– метками защищается объект и его записи. Метка объекта установлена, метки столбцов не установлены, а сам объект создан с защитой строк. Аналогично предыдущему варианту создание записей разрешено субъектам с метками, над которыми доминирует метка объекта, при этом записи наследуют метку субъекта. Мандатные ПРД применяются как на уровне объекта, так и на уровне записей. Операции модификации возможны только над данными, имеющими метку, равную метке таблицы;

– метками защищаются столбцы объекта. Метка объекта не установлена, метки столбцов установлены, а сам объект создан без защиты строк. При этом мандатные ПРД применяются на уровне столбцов. Субъект может читать из столбцов, над метками которых доминирует его метка, вставлять данные в столбцы, чьи метки доминируют над его, и модифицировать те, чьи метки равны его. Операции удаления невозможны при наличии разных меток на столбцы, т. к. операция применяется ко всей строке. Это связано с тем, что операция удаления интерпретируется как последовательное предоставление доступа на чтение и на запись, что возможно только при равенстве меток субъекта и объекта. В случае, когда столбцы имеют разные метки, данное условие выполниться не может. Операция удаления доступна только для администратора и пользователей, обладающих привилегиями игнорирования мандатного разграничения доступа;

– метками защищаются столбцы и записи объекта. В этом случае метка объекта не установлена, метки столбцов установлены, а сам объект создан с защитой строк. При этом мандатные ПРД применяются как на уровне столбцов, так и на уровне записей. Субъект может вставлять данные в столбцы, чьи метки доминируют над его, при этом записи наследуют метку субъекта. Операции чтения и модификации осуществляются над множеством записей, доступных субъекту по мандатным ПРД на записи, и только по столбцам, доступных по мандатным ПРД на столбцы.

Поскольку в процессе работы с данными в СУБД возможно изменение организации их хранения путем изменения схемы объектов БД (метаданных), к подобным операциям так же применяются ПРД.

Модификация метаданных возникает каждый раз при изменении структуры БД, что включает в себя создание, модификацию и удаление объектов БД.

Так как некоторые действия над объектами БД могут влиять на хранящихся в них данных (как правило, модификация или удаление объекта или его части), при использовании мандатного разграничения доступа к данным объекта необходимо разграничивать и доступ к изменению метаданных в части, относящейся к этому объекту.

Аналогично операциям с данными действия с объектами БД должны быть приведены к видам доступа на чтение и на запись информации для возможности применения к ним мандатных ПРД. Все множество операций с метаданными может быть приведено следующим образом:

– `CREATE`, `ADD` — доступ на запись;

– `ALTER`, `DROP` — последовательное выполнение доступа на чтение и запись информации;

– использование или обращение к объекту в других SQL-командах — доступ на чтение.

Проверка мандатных прав доступа к метаданным осуществляется одновременно с проверкой дискреционных прав доступа к ним, после разбора и построения плана запроса, непосредственно перед его выполнением, когда определены все необходимые для проверки данные и проверяемые объекты. Таким образом, доступ предоставляется только при одновременном санкционировании дискреционным ПРД.

Некоторые операции над объектами, такие как `DROP` всего объекта или его столбца и `TRUNCATE` влекут за собой удаление данных. В случае защиты метками записей объекта существуют ограничения на выполнение этих операций.

Операции удаления невозможны при наличии разных меток на записях, т. к. операция применяется ко множеству строк. Это связано с тем, что операция удаления интерпретируется как последовательное предоставление доступа на чтение и на запись, что возможно только при равенстве меток субъекта и объекта. В случае, когда строки имеют разные метки, данное условие выполниться не может.

Операция удаления доступна только для администратора и пользователей, обладающих привилегиями игнорирования мандатного разграничения доступа.

Для настройки работы сервера с мандатным разграничением доступа существует ряд конфигурационных параметров, указываемых в конфигурационном файле конкретного кластера данных:

– `ac_ignore_socket_maclabel` — определяет, будет ли сервер СУБД использовать метку входящего соединения. Если этот параметр установлен в `FALSE`, то метка входящего соединения будет учитываться при определении максимальной доступной метки сессии и после подключения будет доступна только информация с меткой не выше метки входящего соединения. При установке этого параметра в `TRUE` метки сеанса будут определяться максимальной меткой пользователя, полученной из ОС;

– `ac_ignore_server_maclabel` — определяет, будет ли сервер СУБД дополнительно использовать свою метку (метку пользователя `postgres`) при определении

прав пользователя на внесение, удаление и модификацию данных или нет. Если этот параметр установлен в FALSE, то метка сервера используется для блокирования внесения в БД информации с меткой, превышающей метку сервера. Если этот параметр установлен в TRUE, то метка сервера не учитывается;

– ac_enable_trusted_owner — определяет, могут ли владельцы объектов назначать права на доступ к ним другим пользователям. Если этот параметр установлен в значение FALSE, то право назначать права на доступ к любым объектам БД имеет только администратор. Это предотвращает неконтролируемое распространение прав на доступ к информации. Если этот параметр установлен в TRUE, то, кроме администратора, каждый владелец объекта может назначать права на доступ пользователей к «своему» объекту;

– ac_enable_truncate — блокирует (значение FALSE) или разблокирует (значение TRUE) возможность выполнения команды TRUNCATE;

– ac_enable_sequence_mac — если параметр конфигурации установлен в FALSE, то мандатный принцип контроля доступа на последовательности не применяется;

– ac_enable_copy_to_file — блокирует (значение FALSE) или разблокирует (TRUE) возможность выполнения команды COPY с выводом результатов в файл, доступный серверу СУБД;

– ac_debug_print — если установлен в TRUE, добавляет в журнал сервера отладочную информацию о работе механизмов защиты.

Система привилегий СУБД PostgreSQL предназначена для передачи отдельным пользователям прав выполнения определенных административных действий. Обычный пользователь системы не имеет дополнительных привилегий.

Привилегии являются подклассом атрибутов пользователя СУБД PostgreSQL.

Привилегии ОС, используемые в СУБД PostgreSQL, кроме атрибута ac_session_maclabel, не могут быть изменены с помощью средств СУБД ни пользователями, ни администраторами СУБД:

– ac_session_maclabel — текущая мандатная метка сессии пользователя СУБД. Эта метка определяет доступные пользователю объекты БД и является меткой по умолчанию для создаваемых пользователем объектов. При соединении пользователя с СУБД значение этого атрибута устанавливается равным метки соединения или ac_user_max_maclabel;

– ac_user_max_maclabel — максимально возможное значение для ac_session_maclabel;

– ac_user_min_maclabel — минимально возможное значение для ac_session_maclabel;

– ac_capable_ignmaclvl — позволяет пользователю игнорировать мандатный контроль по уровням;

– ac_capable_ignmaccat — позволяет пользователю игнорировать мандатный контроль по категориям;

– ac_capable_mac_readsearch — позволяет пользователю игнорировать мандатный контроль по уровням и категориям при чтении данных;

– ac_capable_setmac — позволяет пользователю изменять текущую метку своей сессии в пределах, заданных ее минимальным и максимальным значением;

– ac_capable_chmac — позволяет пользователю изменять метки объектов БД.

В случае, если пользователь СУБД не зарегистрирован в ОС на стороне сервера СУБД, все его мандатные атрибуты имеют нулевое значение. Администраторам СУБД дополнительно к их атрибутам из ОС всегда добавляются атрибуты ac_capable_ignmaclvl, ac_capable_ignmaccat и ac_capable_chmac.

4.1.13.3. Регистрация событий в защищенной СУБД

Решение задачи регистрации событий в защищенной СУБД обеспечивается на основе использования реализованной в ОС расширенной подсистемы протоколирования (см. 4.1.7). Настройка подсистемы регистрации событий в защищенной СУБД обеспечивается конфигурационным файлом pg_audit.conf конкретного кластера данных.

В этом конфигурационном файле можно задать списки успешных (success events mask) и неуспешных (failure events mask) типов запросов на доступ, которые будут регистрироваться в журнале СУБД и подсистеме аудита ОС для отдельных пользователей и по умолчанию. Списки типов запросов на доступ задаются в виде шестнадцатеричных чисел, в которых каждому типу запроса соответствует установленный (для регистрируемых запросов) или сброшенный (для не регистрируемых запросов) бит:

– на добавление/изменение/удаление пользователей и групп (SUBJECT) соответствует нулевой бит (шестнадцатеричное значение — 1);

– на изменение конфигурации, влияющей на доступ к данным (запрос на изменение значения переменной ac_session_maclabel) (CONFIGURATION) соответствует первый бит (шестнадцатеричное значение — 2);

– на изменение прав доступа к объектам БД (RIGHTS) соответствует второй бит (шестнадцатеричное значение — 4);

– на выборку информации из БД (SELECT) соответствует четвертый бит (шестнадцатеричное значение — 10);

– на добавление информации в БД (INSERT) соответствует пятый бит (шестнадцатеричное значение — 20);

– на изменение информации в БД (UPDATE) соответствует шестой бит (шестнадца-

теричное значение — 40);

– на удаление информации из БД (DELETE) соответствует седьмой бит (шестнадцатеричное значение — 80);

– на очистку данных (TRUNCATE) соответствует восьмой бит (шестнадцатеричное значение — 100);

– на задание колонки таблицы в качестве внешнего ключа (REFERENCES) соответствует десятый бит (шестнадцатеричное значение — 400);

– на добавление триггера к таблице (TRIGGER) соответствует одиннадцатый бит (шестнадцатеричное значение — 800);

– на запуск хранимой процедуры или триггера (EXECUTE) соответствует двенадцатый бит (шестнадцатеричное значение — 1000);

– на использование объекта БД (USAGE) соответствует тринадцатый бит (шестнадцатеричное значение — 2000);

– на создание объектов в БД (CREATE) соответствует шестнадцатый бит (шестнадцатеричное значение — 10000);

– на создание временных объектов в БД (CREATE) соответствует семнадцатый бит (шестнадцатеричное значение — 20000);

– на удаление объектов БД (DROP) соответствует восемнадцатый бит (шестнадцатеричное значение — 40000);

– на изменение объекта БД (ALTER) соответствует девятнадцатый бит (шестнадцатеричное значение — 80000).

Информация о соединении пользователей с БД (CONNECT) и разъединении с ней (DISCONNECT) регистрируется всегда.

4.1.14. Гипертекстовая обработка данных

Решение задачи гипертекстовой обработки данных основано на использовании защищенного комплекса программ гипертекстовой обработки данных, который включает web-сервер Apache2 и браузер Mozilla Firefox, доработанные для интеграции с ядром ОС и базовыми библиотеками с целью обеспечения мандатного разграничения доступа при организации удаленного доступа к информационным ресурсам в информационных и управляющих системах, в которых осуществляется хранение, обработка и передача конфиденциальной информации и информации, содержащей сведения, составляющие государственную тайну.

Web-сервер защищенного комплекса программ гипертекстовой обработки запускается как сервис ОС. При обслуживании запросов пользователей осуществляется переключение в мандатный контекст безопасности пользователя. Информационные ресурсы, к которым осуществляется доступ, хранятся как объекты ФС. Таким образом, доступ к защищаемой информации разграничивается средства расширенной подсистемы безопасности

PARSEC.

В защищенном комплексе программ гипертекстовой обработки обеспечено функционирование в ЕПП (идентификация и аутентификация доменных пользователей) (см. 4.1.3).

4.1.15. Обмен сообщениями электронной почты

Решение задачи обмена сообщениями электронной почты основано на использовании защищенного комплекса программ электронной почты, который включает сервер электронной почты, состоящий из агента передачи электронной почты Exim4, агента доставки электронной почты Dovecot и клиента электронной почты Mozilla Thunderbird, доработанных для реализации следующих дополнительных функциональных возможностей:

– интеграции с ядром ОС и базовыми библиотеками для обеспечения мандатного разграничения доступа к почтовым сообщениям, хранящимся с использованием формата Maildir;

– автоматической маркировки создаваемых пользователем почтовых сообщений, с использованием текущего мандатного контекста пользователя.

Агент передачи электронной почты использует протокол SMTP и обеспечивает решение следующих задач:

– доставку исходящей почты от авторизованных клиентов до сервера, который является целевым для обработки почтового домена получателя;

– прием и обработку почтовых сообщений доменов, для которых он является целевым;

– передачу входящих почтовых сообщений для обработки агентом доставки электронной почты.

Агент доставки электронной почты Dovecot предназначен для решения задач по обслуживанию почтового каталога и предоставления удаленного доступа к почтовому ящику по протоколу IMAP.

Сервер защищенного комплекса программ электронной почты использует в качестве формата почтового хранилища MailDir, т. к. формат mailbox не поддерживает работу с мандатными уровнями и категориями, отличными от нуля.

Клиент электронной почты — прикладное ПО, устанавливаемое на рабочем месте пользователя и предназначенное для получения, написания, отправки и хранения сообщений электронной почты пользователя.

В защищенном комплексе программ электронной почты обеспечено функционирование в ЕПП (идентификация и аутентификация доменных пользователей).

5. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ

5.1. Входными данными для ОС являются:

– обращение субъектов доступа (процессов и команд СУБД) к защищаемым именованным объектам доступа — файлам (программам, библиотекам, файлам с пользовательской и служебной информацией), каталогам, специальным файлам (устройствам, ссылкам, каналам FIFO и т. п.), БД и их элементам (таблицам, записям, полям записей, триггерам и т. п.), а также средствам IPC (портам, сокетам, семафорам);

– атрибуты, определяющие полномочия субъектов доступа и правила разграничения доступа к объектам доступа.

5.2. Выходными данными для ОС является результат использования субъектом доступа защищаемого объекта, предоставленного ему в соответствии с установленными ПРД. К таким результатам могут относиться: запуск программы, редактирование файла, создание сокетов, добавление данных в БД и т. п.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

БД — база данных
ЕПП — единое пространство пользователей
НЖМД — накопитель на жестком магнитном диске
ОП — оперативная память
ОС — операционная система
ПО — программное обеспечение
ПРД — правила разграничения доступом
СЗИ — средства защиты информации
СУБД — система управления базами данных
ФС — файловая система

ACL — Access Control List (список контроля доступа)
ALD — Astra Linux Directory (единое пространство пользователей)
CIFS — Common Internet File System (общий протокол доступа к файлам Интернет)
DHCP — Dynamic Host Configuration Protocol (протокол динамической конфигурации хоста)
DNS — Domain Name System (служба доменных имен)
FIFO — First-In, First-Out (первым пришел — первым обслужен — дисциплина очереди)
FTP — File Transfer Protocol (протокол передачи файлов)
GID — Group Identifier (идентификатор группы)
HTTP — HyperText Transfer Protocol (протокол передачи гипертекстовых файлов)
IP — Internet Protocol (протокол Интернет)
IPC — InterProcess Communication (межпроцессное взаимодействие)
IMAP — Internet Message Access Protocol (протокол доступа к сообщениям в сети Интернет)
LDAP — Lightweight Directory Access Protocol (легковесный протокол доступа к сервисам каталогов)
NFS — Network File System (сетевая файловая система)
NTP — Network Time Protocol (синхронизирующий сетевой протокол)
NSS — Name Service Switch (диспетчер службы имен)
PAM — Pluggable Authentication Modules (встраиваемые модули аутентификации)
PID — Process Identifier (идентификатор процесса)
SMB — Session Message Block (блок сессионных сообщений)
SMTP — Simple Mail Transfer Protocol (простой протокол электронной почты)

SSH — Secure Shell Protocol (протокол передачи информации в зашифрованном виде)

TCP — Transmission Control Protocol (протокол передачи данных)

TFTP — Trivial File Transfer Protocol (простейший протокол передачи файлов)

UID — User Identifier (идентификатор пользователя)

Лист регистрации изменений

Изм.	Номера листов (страниц)				Всего листов (страниц) в докум.	№ документа	Входящий № сопрово-дительного докум. и дата	Подп.	Дата
	измененных	замененных	новых	аннулированных					