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»

Руководство по КСЗ. Часть 2

РУСБ.10015-01 97 01-2

Листов 38

2010

АННОТАЦИЯ

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

В документе приведена структура тестов КСЗ, описано проведение тестирования, а также приведены проверки идентификации и аутентификации, дискреционного и мандатного разграничения доступом, очистки памяти и изоляции модулей, маркировки документов, защиты ввода-вывода информации на отчуждаемый физический носитель и сопоставления пользователя с устройством, регистрации событий, надежного восстановления, контроля целостности КСЗ.

СОДЕРЖАНИЕ

1. СТРУКТУРА ТЕСТОВ

Тестирование КСЗ заключается в проверке функционирования подсистем реализующих функции по защите. В части, касающейся функций ядра ОС по защите от НСД, тестируется подсистема безопасности PARSEC, а в части, касающейся функций защиты СУБД, — СУБД PostgreSQL.

1.1. Подсистема безопасности PARSEC ОС

Тестирование подсистемы безопасности PARSEC производится тестовой системой parsec-tests. Она состоит из набора тестов для тестирования механизмов мандатного разграничения доступа к объектам файловой системы, IPC и сетевой подсистемы, механизмов изоляции модулей и очистки памяти и внешних носителей, метода безопасного удаления файлов, привилегий пользователей и регистрации событий. Все тесты имеют опцию «Число итераций», позволяющую настраивать число проходов теста, на каждом из которых на процесс устанавливается случайная мандатная метка.

1.1.1. Модуль тестирования механизма дискреционного разграничения доступа к объектам ФС

Реализован в виде теста rwx.sh.

Проверка чтения и записи в файл с данными правами доступа для пользователя, группы пользователя и остальных пользователей. Чтение и запись происходит от имени пользователя-владельца файла, пользователя, входящего в группу владельца файла, а также от имени пользователя, не являющегося владельцем файла и не входящим в группу пользователя-владельца файла.

1.1.2. Модули тестирования механизма мандатного разграничения доступа

1.1.2.1. Модуль тестирования механизма мандатного разграничения доступа к объектам ФС

Реализован в виде теста fmac. Осуществляет следующие проверки:

1) наследование файлом мандатной метки процесса при открытии файла;
2) установка мандатной метки на файл;
3) доступ на чтение, запись, чтение и запись процесса с мандатной меткой M1 к файлу с мандатной меткой M2. Для метки M1 проверяются все логически возможные ее комбинации с M2.

1.1.2.2. Модуль тестирования механизма мандатного разграничения доступа к объектам IPC

Реализован в виде теста ipc_mac. Осуществляет следующие проверки:

1) доступ процесса к данном объекту IPC (семафор, разделяемая память, очередь

сообщения) методом получения процессом идентификатора данного IPC;

2) посылка сигнала процессу (нити) с данной мандатной меткой `M1` от процесса (нити) с мандатной меткой `M2`.

1.1.2.3. Модуль тестирования механизма привилегий процесса

Реализован в виде теста `cap_mac`. Осуществляет следующие проверки:

1) наследование процессом своих LINUX- и PARSEC-привилегий при включенном флаге процесса `PR_KEEP_CAPS` после переключения с суперпользователя на обычного пользователя;

2) получение полных LINUX- и PARSEC-привилегий процессом при включенном флаге процесса `PR_KEEP_CAPS` после переключения обычного пользователя на суперпользователя;

3) наследование наследуемых LINUX- и PARSEC-привилегий процессом.

1.1.2.4. Модуль тестирования механизма мандатного разграничения доступа при сетевых взаимодействиях

Реализован в виде тестов `tcpip_mac` и `tcpip6_mac` для версий протокола IPv4 и IPv6, соответственно.

Для протоколов TCP, UDP и UNIX-сокетов (потоковых и датаграммных) осуществляется проверка возможности соединения и отправки/приема данных (для сокетов без соединения) с клиента с мандатной меткой `M1` на сервер с меткой `M2`. Проверяются все логически возможные комбинации `M1` и `M2`.

Аналогичные проверки осуществляются для привилегированного сокета.

1.1.3. Модуль тестирования механизмов работы с памятью и изоляции модулей

Осуществляется проверка изоляции модулей и очистки памяти. Проверка обнаружения ранее записанной в данный участок выделенной памяти сигнатуры после повторного выделения (после освобождения) данной памяти.

1.1.4. Модуль тестирования механизма очистки памяти внешних носителей

Реализован в виде теста `secdelrm.sh`.

Проверка наличия содержимого некоторого файла в тестовом разделе, смонтированном с опциями `secdel` или `secrm` после удаления данного файла. В тесте поддерживаются ФС Ext2/Ext3.

1.1.5. Модуль тестирования подсистемы регистрации событий

Реализован в виде теста `audit.sh`.

Проверка работоспособности службы аудита заключается в наличии сообщений аудита в log-файле системы аудита в результате срабатывания события аудита, которое

зарегистрировано ранее.

1.2. СУБД PostgreSQL

В КСЗ PostgreSQL реализованы следующие функции по защите от НСД:

– дискреционное разграничение доступа к объектам БД;

– мандатное разграничение доступа к объектам и данным БД;

– взаимодействие пользователя с КСЗ;

– идентификация и аутентификация;

– надежное восстановление;

– регистрация;

– тестирование.

В состав КСЗ PostgreSQL входит пакет `postgresql-se-test-8.4`, содержащий тесты и вспомогательные скрипты для проведения тестирования общей функциональности СУБД по управлению БД и механизмов разграничения доступа.

Тестирование общей функциональности СУБД по управлению БД основано на использовании программы `pg_regress`, входящей в пакет `postgresql-server-dev-8.4`. Сами тесты входят в пакет `postgresql-se-test-8.4`.

При установке сервера СУБД Postgres, в каталог `/usr/lib/postgresql/8.4/lib` помещается каталог `regress` из исходных текстов PostgreSQL, содержащий необходимые SQL-скрипты, входные данные, библиотеки функций и эталоны результатов для выполнения тестов. Тестирование осуществляется выполнением SQL-скритов с запросами, относящимися к тестируемой части функциональности, утилитой командной строки `psql`, предоставляющей доступ к БД.

Результат выполнения сохраняется в выходном файле и в дальнейшем сравнивается с эталонным файлом результатов выполнения. Решение об успешности прохождения теста принимается по результату сравнения. Тест считается выполненным успешно при отсутствии расхождения результатов с эталоном. Тестирование общей функциональности является стандартным для PostgreSQL и рассматривается в целом.

Для тестирования функций защиты применяется аналогичный подход. В каталог `/usr/share/postgresql/test/` помещается каталог `rbt`, содержащий необходимые SQL-скрипты, вспомогательные скрипты и эталоны результатов для выполнения тестов.

Каталог `rbt` имеет следующую структуру:

– `expected` — каталог, содержащий файлы эталонов результатов теста;

– `sql` — каталог, содержащий SQL-скрипты тестов;

– `support` — каталог, содержащий вспомогательные SQL-скрипты общих частей тестов, таких как: создание тестовой БД, инициализация объектов, настройка пара-

метров и завершение работы. Так же в этом каталоге располагается скрипт создания пользователей в ОС с назначением им соответствующих атрибутов безопасности;

– `runregressiontests` — вспомогательный скрипт для запуска процесса тестирования общей функциональности;

– `runsetests` — вспомогательный скрипт для запуска процесса тестирования функциональности.

Описание тестов приведено в таблице 1.

Таблица 1

Тест	Описание
`acl-column`	Проверка дискреционного метода контроля доступа к столбцам объектов БД
`acl-database`	Проверка дискреционного метода контроля доступа к БД
`acl-function`	Проверка дискреционного метода контроля доступа к функциям (хранимым процедурам)
`acl-language`	Проверка дискреционного метода контроля доступа к процедурным языкам
`acl-role`	Проверка дискреционного метода контроля доступа при использовании ролей (групп)
`acl-schema`	Проверка дискреционного метода контроля доступа к схемам
`acl-sequence`	Проверка дискреционного метода контроля доступа к последовательностям
`acl-table`	Проверка дискреционного метода контроля доступа к таблицам
`acl-tablespace`	Проверка дискреционного метода контроля доступа к областям хранения данных
`acl-view`	Проверка дискреционного метода контроля доступа к видам
`mac-alterenablecolmacs`	Проверка изменения режима использования мандатных меток столбцов
`mac-alter-meta`	Проверка мандатного метода контроля доступа при модификации метаданных
`mac-altersetcolumnmac`	Проверка изменения мандатных меток столбцов
`mac-altersetmac`	Проверка изменения мандатных меток объектов
`mac-column`	Проверка мандатного метода контроля доступа к столбцам
`mac-copy-file-deny`	Проверка запрета работы команды `COPY` при выводе данных в файл
`mac-copy-file`	Проверка работы команды ввода/вывода `COPY` при работе с файлами
`mac-copy-std`	Проверка работы команды ввода/вывода `COPY` при работе со стандартными потоками

Окончание таблицы 1

Тест	Описание
mac-create	Проверка создания объектов и автоматического назначения мандатных меток
mac-createtableas	Проверка создания объектов командой CREATE TABLE AS
mac-delete	Проверка мандатного метода контроля доступа при удалении
mac-insert	Проверка мандатного метода контроля доступа при вставке
mac-largeobjects	Проверка мандатного метода контроля доступа при работе с большими объектами
mac-plperl, mac-plperlu, mac-plpgsql, mac-plpythonu, mac-pltcl, mac-pltclu	Проверка мандатного метода контроля доступа в хранимых процедурах на языках PL/Perl, Untrusted PL/Perl, PL/pgSQL, Untrusted PL/Python, PL/Tcl и Untrusted PL/Tcl
mac-select	Проверка мандатного метода контроля доступа при выборке данных
mac-sequence	Проверка мандатного метода контроля доступа к последовательностям
mac-tableview	Проверка мандатного метода контроля доступа к таблицам и видам
mac-triggers-perl, mac-triggers-perlu, mac-triggers-pgsql, mac-triggers-pythonu, mac-triggers-tcl, mac-triggers-tclu	Проверка мандатного метода контроля доступа в триггерах на языках PL/Perl, Untrusted PL/Perl, PL/pgSQL, Untrusted PL/Python, PL/Tcl и Untrusted PL/Tcl
mac-update	Проверка мандатного метода контроля доступа при модификации данных
misc-altertable	Проверка сохранности правил разграничения доступа при изменении структуры таблицы
misc-cluster	Проверка сохранности правил разграничения доступа при оптимизации индексов таблицы
misc-config	Проверка конфигурационных параметров КСЗ
misc-indexes	Проверка работы индексов совместно с системой защиты
misc-maclabel	Проверка работы встроенных функций с типом maclabel («мандатная метка»)
misc-rules	Проверка мандатного метода контроля доступа при использовании правил
misc-sequence	Проверка работы последовательностей без применения мандатных меток
misc-vacuum	Проверка взаимодействия задач технического обслуживания данных с системой защиты
misc-VAL	Проверка механизма надежного восстановления

Для проведения тестирования необходимо наличие установленного сервера PostgreSQL и следующих пакетов:

– дополнительные возможности для PostgreSQL;
– процедурный язык PL/Perl для PostgreSQL 8.4;
– процедурный язык PL/Python для PostgreSQL 8.4;
– процедурный язык PL/Tcl для PostgreSQL 8.4.

При выполнении каждого теста создается отдельная БД, после чего в кластере создаются пользователи, соответствующие заведенным ранее в ОС. С помощью инструментов управления сервером осуществляются настройки сервера для используемого в тесте сочетания конфигурационных параметров. В процессе исполнения создаются необходимые объекты БД, изменяются ПРД и выполняются запросы от пользователей с разными атрибутами безопасности и наборами привилегий. При этом проверяется как успешность выполнения запросов, так и отказы доступа. В тестах проверки мандатного метода контроля доступа проверяется доступ пользователей с разными мандатными метками и наборами привилегий к защищенным мандатными метками данным (объектам, столбцам объектов и строкам). По завершении теста все созданные объекты, включая пользователей и БД, удаляются.

В ходе тестов непосредственно проверяются следующие механизмы:

– дискреционное разграничение доступа к объектам БД;
– мандатный разграничение доступа к объектам и данным БД;
– надежное восстановление.

Функции защиты, такие как взаимодействие пользователя с КСЗ, идентификация и аутентификация, регистрация, тестирование проверяются косвенным образом, т. к. в каждом тесте осуществляется доступ разных пользователей, используется их взаимодействие с КСЗ и регистрируются все попытки доступа к защищаемым объектам, создания и уничтожения объектов и действия по изменению ПРД.

1.2.1. acl-column

Проверка дискреционного разграничение доступа к столбцам объектов БД заключается в последовательном назначении и отборе прав доступа пользователя к столбцу объекта и выполнении запросов на чтение, вставку, модификацию, удаление данных и создание ограничения, ссылающегося на столбец. При этом оценивается соответствие результата предоставления доступа диспетчером доступа СУБД (успех или отказ) установленным правам пользователя.

1.2.2. acl-database

Проверка дискреционного разграничение доступа к БД заключается в последовательном назначении и отборе прав на создание схем и временных объектов и выполнении соответствующих запросов. При этом оценивается соответствие результата предоставления доступа диспетчером доступа СУБД (успех или отказ) установленным правам пользователя.

В связи с тем, что при отсутствии права на соединение с БД соединение не устанавливается, тест состоит из двух частей. В первой создается БД, и осуществляется попытка доступа при отсутствии права на установку соединения с этой БД. Во второй части пользователю предоставляется указанное право, и он осуществляет успешное соединение с БД. После этого последовательно проверяются остальные права, применимые к БД.

1.2.3. acl-function

Проверка дискреционного разграничение доступа к функциям (хранимым процедурам) заключается в последовательном назначении и отборе права исполнения функции и попытках ее выполнения. При этом оценивается соответствие результата предоставления доступа диспетчером доступа СУБД (успех или отказ) установленным правам пользователя.

1.2.4. acl-language

Проверка дискреционного разграничение доступа к процедурным языкам заключается в последовательном назначении и отборе права использования процедурного языка и попытках создания функции на этом языке. При этом оценивается соответствие результата предоставления доступа диспетчером доступа СУБД (успех или отказ) установленным правам пользователя.

1.2.5. acl-role

Проверка дискреционного разграничение доступа при использовании ролей (групп) заключается в создании ролей (групп), обладающих определенным набором прав доступа к объекту, и попытках выполнения запросов на чтение, вставку, модификацию и удаление данных. При этом меняется состав ролей пользователя, что влияет на проверяемый результат предоставления доступа диспетчером доступа СУБД (успех или отказ).

1.2.6. acl-schema

Проверка дискреционного разграничение доступа к схемам заключается в последовательном назначении и отборе прав на использование схемы и создание в ней объектов и выполнении соответствующих запросов. При этом оценивается соответствие результата предоставления доступа диспетчером доступа СУБД (успех или отказ) установленным правам пользователя.

1.2.7. acl-sequence

Проверка дискреционного разграничение доступа к последовательностям заключается в последовательном назначении и отборе прав доступа пользователя к последовательности и выполнении запросов на получение и изменение ее значения. При этом оценивается соответствие результата предоставления доступа диспетчером доступа СУБД (успех или отказ) установленным правам пользователя.

1.2.8. acl-table

Проверка дискреционного разграничение доступа к таблицам заключается в последовательном назначении и отборе прав доступа пользователя к таблице и выполнении запросов на чтение, вставку, модификацию, удаление данных и создании триггеров и ограничений целостности. При этом оценивается соответствие результата предоставления доступа диспетчером доступа СУБД (успех или отказ) установленным правам пользователя.

1.2.9. acl-tablespace

Проверка дискреционного разграничение доступа к областям хранения данных заключается в последовательном назначении и отборе права на создание объектов в области хранения данных и выполнении соответствующих запросов. При этом оценивается соответствие результата предоставления доступа диспетчером доступа СУБД (успех или отказ) установленным правам пользователя.

1.2.10. acl-view

Проверка дискреционного разграничение доступа к видам заключается в последовательном назначении и отборе прав доступа пользователя к виду и выполнении запросов на чтение, вставку, модификацию и удаление данных. При этом оценивается соответствие результата предоставления доступа диспетчером доступа СУБД (успех или отказ) установленным правам пользователя.

1.2.11. mac-alterenablecolmacs

Проверка изменения режима использования мандатных меток столбцов заключается в попытках установки и снятия признака использования мандатных меток столбцов у объекта. При этом оценивается возможность установки признака владельцем объекта, если он имеет привилегию изменения мандатной метки, и невозможность изменения признака у таблиц другого пользователя.

1.2.12. mac-alter-meta

Проверка мандатного разграничение доступа при модификации метаданных заключается в попытках модификации метаданных, а именно создании и изменении объектов БД, при этом оценивается возможность выполнения модификаций в зависимости от метки

объекта.

1.2.13. mac-altersetcolumnmac

Проверка изменения мандатных меток столбцов объекта заключается в попытках установки и изменения мандатных меток столбцов объекта. При этом оценивается возможность изменения метки владельцем объекта, если он имеет привилегию изменения мандатной метки, и невозможность изменения мандатной метки у столбцов объекта другого пользователя.

1.2.14. mac-altersetmac

Проверка изменения мандатных меток объектов заключается в попытках установки и изменения мандатных меток объекта. При этом оценивается возможность изменения метки владельцем объекта, если он имеет привилегию изменения мандатной метки, и невозможность изменения мандатной метки у объекта другого пользователя.

1.2.15. mac-column

Проверка разграничение доступа доступа к столбцам заключается в последовательном выполнении запросов на чтение, вставку, модификацию и удаление данных пользователями с разными мандатными метками и наборами привилегий к защищенным мандатным меткам столбцов. Создается таблица с защищенными мандатными метками строками. Таблица заполняется данными пользователей с разными мандатными метками сессий. Проверка мандатного метода контроля доступа к столбцам осуществляется при сброшенном и установленном признаке использования мандатных меток столбцов у объекта. При этом оценивается соответствие результата предоставления доступа диспетчером доступа СУБД (успех или отказ) установленным мандатным меткам и наборам привилегий пользователей.

1.2.16. mac-copy-file-deny

Проверка запрета работы команды `COPY` при выводе данных в файл заключается в попытках ввода/вывода в файл на сервере защищенных данных командой `COPY` при установленном запрете этой операции. При этом проверяется, что никакой пользователь, даже обладающий привилегиями суперпользователя БД, не может выполнить действия по вводу/выводу защищенных данных в файл на сервере.

1.2.17. mac-copy-file

Проверка работы команды ввода/вывода `COPY` при работе с файлами заключается в попытках ввода/вывода в файл на сервере защищенных данных командой `COPY` в случае разрешения этой операции. Проверяется, что операцию может выполнить только пользователь, обладающий привилегиями суперпользователя БД. Вычисленная в процессе вывода

максимальная мандатная метка данных назначается результирующему файлу с данными при наличии у системного пользователя `postgres` привилегии назначения мандатных меток.

1.2.18. mac-copy-std

Проверка работы команды ввода/вывода `COPY` при работе со стандартными потоками заключается в попытках ввода/вывода в стандартные потоки `stdin`/`stdout` защищенных данных командой `COPY`. Операция осуществляет вывод на стороне клиента. Проверяется, что при выполнении операции применяется мандатный метод контроля доступа.

1.2.19. mac-create

Проверка создания объектов и автоматического назначения мандатных меток заключается в создании объектов БД, для которых предусмотрена защита мандатным методом контроля доступа. При этом оценивается соответствие мандатной метки, назначенной создаваемым объектам, и текущей мандатной метки сессии пользователя.

1.2.20. mac-createtableas

Проверка создания объектов командой `CREATE TABLE AS` заключается в создании объектов БД данной командой на основе существующего объекта с данными. При этом оценивается соответствие мандатной метки, назначенной создаваемым объектам, и текущей мандатной метки сессии пользователя и заполнения создаваемых объектов данными в соответствии с правилами мандатного разграничения доступа.

1.2.21. mac-delete

Проверка мандатного разграничение доступа при удалении заключается в последовательных попытках удаления существующих данных, защищенных разными мандатными метками, пользователями с разными мандатными метками и наборами привилегий. При этом оценивается соответствие результата предоставления доступа диспетчером доступа СУБД (успех или отказ) установленным мандатным меткам и наборам привилегий пользователей.

1.2.22. mac-insert

Проверка мандатного разграничение доступа при вставке заключается в последовательных попытках вставки данных в защищенный мандатной меткой объект пользователями с разными мандатными метками и наборами привилегий. При этом оценивается соответствие результата предоставления доступа диспетчером доступа СУБД (успех или отказ) установленным мандатным меткам и наборам привилегий пользователей.

1.2.23. mac-largeobjects

Проверка мандатного разграничение доступа при работе с большими объектами заключается в создании большого объекта и в последовательных попытках выполнения операций с ним пользователями с разными мандатными метками и наборами привилегий. При этом оценивается соответствие мандатной метки, назначенной создаваемому большому объекту, и текущей мандатной метки сессии пользователя и результатов предоставления доступа диспетчером доступа СУБД (успех или отказ) установленным мандатным меткам и наборам привилегий пользователей.

1.2.24. mac-plperl, mac-plperlu, mac-plpgsql, mac-plpythonu, mac-pltcl, mac-pltclu

Проверка мандатного разграничение доступа в хранимых процедурах на языках PL/Perl, Untrusted PL/Perl, PL/pgSQL, Untrusted PL/Python, PL/Tcl и Untrusted PL/Tcl заключается в вызове пользователями функций в разных режимах исполнения — от имени вызывающего и от имени создателя. При этом в режиме исполнения функции от имени создателя при вызове функции должны использоваться мандатные атрибуты создателя, а в режиме исполнения от имени вызывающего — мандатные атрибуты вызывающего функцию пользователя.

1.2.25. mac-select

Проверка мандатного разграничение доступа при выборке заключается в последовательных попытках выборки существующих данных, защищенных разными мандатными метками, пользователями с разными мандатными метками и наборами привилегий. При этом оценивается соответствие результатов предоставления доступа диспетчером доступа СУБД (успех или отказ) установленным мандатным меткам и наборам привилегий пользователей.

1.2.26. mac-sequence

Проверка мандатного разграничение доступа к последовательностям заключается в последовательном выполнении запросов на получение и изменение значения последовательности, защищенной мандатной меткой, пользователями с разными мандатными метками и наборами привилегий при установленном конфигурационном параметре разрешения использования мандатных меток последовательностей. При этом оценивается соответствие результатов предоставления доступа диспетчером доступа СУБД (успех или отказ) установленным мандатным меткам и наборам привилегий пользователей.

1.2.27. mac-tableview

Проверка мандатного разграничение доступа к таблицам и видам заключается в последовательном выполнении запросов на чтение, вставку, модификацию и удаление

данных пользователями с разными мандатными метками и наборами привилегий к защищенным мандатными метками таблицам и видам. При этом рассматриваются варианты создания правил для операций с видом пользователями с разными наборами привилегий мандатного доступа, и оценивается соответствие результатов предоставления доступа диспетчером доступа СУБД (успех или отказ) установленным мандатным меткам и наборам привилегий пользователей.

1.2.28. mac-triggers-plperl, mac-triggers-plperlu, mac-triggers-plpgsql, mac-triggers-plpythonu, mac-triggers-pltcl, mac-triggers-pltclu

Проверка мандатного разграничение доступа в триггерах на языках PL/Perl, Untrusted PL/Perl, PL/pgSQL, Untrusted PL/Python, PL/Tcl и Untrusted PL/Tcl заключается в вызове функций триггеров в разных режимах исполнения — от лица вызывающего и от лица создателя. При этом в режиме исполнения функции триггера от имени создателя при вызове функции триггера должны использоваться мандатные атрибуты создателя, а в режиме исполнения — от имени вызывающего мандатные атрибуты вызывающего функцию пользователя. Так же проверяется реализация правил мандатного разграничения доступа при осуществлении доступа к данным объекта, который вызывает триггер и другим объектам БД, и возможность изменения мандатной метки обрабатываемых данных пользователем с соответствующими привилегиями мандатного доступа.

1.2.29. mac-update

Проверка мандатного разграничение доступа при модификации данных заключается в последовательных попытках модификации существующих данных, защищенных разными мандатными метками, пользователями с разными мандатными метками и наборами привилегий. При этом оценивается соответствие результатов предоставления доступа диспетчером доступа СУБД (успех или отказ) установленным мандатным меткам и наборам привилегий пользователей.

1.2.30. misc-altertable

Проверка сохранности прав доступа при изменении структуры таблицы заключается в модификации структуры таблицы добавлением и удалением нового столбца, при этом мандатные атрибуты существующих в таблице данных не должны изменяться, что проверяется выполнением тестов (см. 1.2.25) до и после выполнения модификаций.

1.2.31. misc-cluster

Проверка сохранности прав доступа при оптимизации индексов таблицы заключается в оптимизации индексов таблицы командой `CLUSTER`, что может приводить к физической реорганизации данных. При этом мандатные атрибуты существующих в таблице данных не должны изменяться, что проверяется выполнением тестов (см. 1.2.25) до и по-

сле выполнения операции.

1.2.32. misc-config

Проверка конфигурационных параметров КСЗ заключается в просмотре существующих конфигурационных параметров КСЗ PostgreSQL, попытках изменения значения тех параметров, для которых это предусмотрено. Так же проверяется, что записи системного каталога `pg_largeobject`, содержащего информацию о больших объектах, защищаются мандатной меткой. Отдельно проверяется возможность удаленного изменения конфигурационных параметров сервера, требующих его перезапуска.

1.2.33. misc-indexes

Проверка работы индексов совместно с системой защиты заключается в последовательном выполнении запросов на вставку и чтение данных пользователями с разными мандатными метками и наборами привилегий к защищенным мандатными метками таблицам, содержащим индексы. При этом индексы и механизм мандатного метода контроля доступа не должны влиять друг на друга.

1.2.34. misc-maclabel

Проверка работы встроенных функций с типом `maclabel` («мандатная метка») заключается в оценке результатов выполнения функций `session_maclabel`, `maclabel`, `level` и `category`, относящихся к КСЗ PostgreSQL и оперирующих типом `maclabel`.

1.2.35. misc-rules

Проверка мандатного разграничение доступа при использовании правил заключается в последовательном выполнении запросов на чтение, вставку, модификацию и удаление данных пользователями с разными мандатными метками и наборами привилегий к защищенным мандатными метками таблицам с заданными правилами изменения запроса. При этом оцениваются варианты создания правил пользователями с разными наборами привилегий мандатного доступа.

1.2.36. misc-sequence

Проверка работы последовательностей без применения мандатных меток заключается в последовательном выполнении запросов на получение и изменение значения последовательности, не защищенной мандатной меткой, пользователями с разными мандатными метками и наборами привилегий, при не установленном конфигурационном параметре разрешения использования мандатных меток последовательностей.

1.2.37. misc-vacuum

Проверка взаимодействия задач технического обслуживания данных с системой защиты заключается в выполнении операции `VACUUM`. При этом мандатные атрибуты суще-

ствующих в таблице данных не должны изменяться, что проверяется выполнением тестов (см. 1.2.25) до и после выполнения операции.

1.2.38. misc-VAL

Проверка механизмов надежного восстановления данных заключается в создании двух таблиц в рамках отдельных транзакций и заполнении их данными и последующем уничтожении процессов СУБД. При этом создание одной таблицы подтверждается завершением транзакции, а другой — нет, поэтому после уничтожения процессов СУБД и ее перезапуска в БД будет существовать только одна таблица.

2. ПРОВЕДЕНИЕ ТЕСТИРОВАНИЯ

2.1. Подсистема безопасности PARSEC ОС

Для запуска автоматической процедуры тестирования необходимо:

1) войти в систему как суперпользователь;

2) открыть терминал и установить пакет тестов командой:

```
apt-get install parsec-tests
```

3) зайти в каталог `/root/parsec-tests` и осуществить выполнение всего набора тестов запуском скрипта:

```
run.sh
```

(или с опцией `-v` для режима подробного вывода сообщений). При этом на экране будут появляться сообщения о прохождении и результатах выполнения тестов.

Подробная информация о результатах тестирования будет записана в файл `tests.log`, находящийся в каталоге `/root/parsec-tests`.

На экране последовательно будут появляться строки отчета о проведении отдельных пунктов проверок.

Если в тестах хотя бы одна проверка завершится с ошибкой, то вместо строки:

```
Тест ПРОШЕЛ
```

в файле будет содержаться строка:

```
[!] ОШИБКА тестирования
```

2.2. СУБД PostgreSQL

Для запуска автоматической процедуры тестирования необходимо:

1) войти в систему как суперпользователь;

2) проверить, используя менеджер пакетов `fly-admin-package`, наличие в системе установленных пакетов защищенной СУБД;

3) запустить окно терминала;

4) установить пакет `postgresql-se-test-8.4` командой:

```
apt-get install postgresql-se-test-8.4
```

5) перейти в каталог `/usr/share/postgresql/8.4/test/rbt/` командой:

```
cd /usr/share/postgresql/8.4/test/rbt/
```

6) запустить тесты командой:

```
./runtests
```

В ходе тестирования будут осуществлены необходимые подготовительные действия и запуск регрессионных тестов стандартных функциональных возможностей СУБД PostgreSQL и тестов дополнительных функциональных возможностей по разграничению доступа.

Успешность выполнения каждого теста подтверждается сообщением:

```
успех
```

После выполнения тестов выдается общая информация о количестве выполненных тестов и их ошибочности или успешности, например:

```
Всего = 48, запущено = 48, ошибочных = 2
```

Проверка по пункту считается успешной, если после выполнения программы на экране появится сообщение:

```
Все 121 тест(а,ов) выполнены успешно
```

для регрессионного тестирования и сообщение:

```
Всего = 48, запущено = 48, ошибочных = 0
```

для тестов разграничения доступа.

3. ПРОВЕРКА ИДЕНТИФИКАЦИИ И АУТЕНТИФИКАЦИИ

3.1. Идентификация и аутентификация

Для проверки идентификации и аутентификации необходимо:

1) войти в систему от имени суперпользователя;

2) добавить в систему пользователя `test` командой:

```
adduser test
```

3) задать пароль пользователю `test` командой:

```
passwd test
```

4) войти в систему от имени пользователя `test`;

5) набрать в терминале команду:

```
id
```

Будет показана информация о пользователе (его идентификатор, группы);

6) набрать в терминале:

```
macid
```

Будет показана информация о мандатном уровне и категориях пользователя;

7) выйти из системы и зайти от имени суперпользователя;

8) набрать команду в терминале:

```
tail /var/log/secure
```

Будет выведен фрагмент журнала безопасности. Факт аутентификации пользователя отражен напротив строки `login:`.

Информация о зарегистрированных пользователях системы содержится в файлах конфигурации. Изменять эти файлы может только суперпользователь.

3.2. Запрет на доступ несанкционированного пользователя

Для проверки запрета на доступ несанкционированного пользователя необходимо:

1) попытаться войти в систему от имени несуществующего пользователя, набрав произвольный идентификатор и/или пароль (например, `asdf`);

2) зайти в систему от имени суперпользователя и набрать команду:

```
tail /var/log/messages
```

Будет показан фрагмент системного журнала событий. Информация о неуспешном входе несанкционированного пользователя показана напротив строки `login:`.

3.3. Идентификация и аутентификация при работе с БД

Проверка идентификации и аутентификации при работе с БД осуществляется в автоматической процедуре тестирования СУБД (см. 2.2).

Проверка по данному пункту считается успешной при успешном выполнении автоматической процедуры тестирования СУБД, в частности теста доступа к БД (см. 1.2.2).

4. ПРОВЕРКА ДИСКРЕЦИОННОГО РАЗГРАНИЧЕНИЯ ДОСТУПА

4.1. Механизм дискреционного разграничения доступа к объектам ФС

Проверка механизма дискреционного разграничения доступа к объектам ФС осуществляется в ходе выполнения автоматической процедуры тестирования подсистемы безопасности PARSEC (см. 2.1).

Проверяется механизм дискреционного разграничения доступа к объектам ФС, включая создание файлов с дискреционными правами доступа, чтение файлов с установленными дискреционными правами доступа, запись в файлы с установленными дискреционными правами доступа.

Проверка по данному пункту считается успешной, если файл отчета tests.log содержит следующие строки:

```
---[rwx.sh]: начало теста
Проверка чтения файла владельцем...УСПЕШНО
Проверка записи файла владельцем...УСПЕШНО
Проверка чтения файла группой-владельцем...УСПЕШНО
Проверка записи файла группой-владельцем...УСПЕШНО
Проверка чтения файла сторонним пользователем...УСПЕШНО
Проверка записи файла сторонним пользователем...УСПЕШНО
Тест ПРОШЕЛ
---[rwx.sh]: завершение теста
```

4.2. Механизм дискреционного разграничения доступа к объектам БД

Проверка механизма дискреционного разграничения доступа к объектам БД осуществляется в ходе выполнения автоматической процедуры тестирования СУБД PostgreSQL (см. 2.2).

При этом проверяется доступ к объектам БД: базам данных (см. 1.2.2), таблицам (см. 1.2.8), представлениям (см. 1.2.10), столбцам таблиц и представлений (см. 1.2.1), языкам программирования (см. 1.2.4), функциям (см. 1.2.3), схемам (см. 1.2.6), последовательностям (см. 1.2.7) и табличным пространствам (см. 1.2.9). Так же проверяется дискреционное разграничение доступа при использовании ролей (групп) (см. 1.2.5).

Проверка по данному пункту считается успешной при успешном выполнении указанных тестов в составе автоматической процедуры тестирования СУБД.

5. ПРОВЕРКА МАНДАТНОГО РАЗГРАНИЧЕНИЯ ДОСТУПА

5.1. Механизм мандатного разграничения доступа к объектам ФС

Проверка механизма мандатного разграничения доступа к объектам ФС осуществляется в ходе выполнения автоматической процедуры тестирования подсистемы безопасности PARSEC (см. 2.1).

Проверяется механизм мандатного разграничения доступа к объектам ФС, включая создание файлов с мандатной меткой, чтение файлов с установленным мандатным уровнем, запись в файлы с установленным мандатным уровнем, запись и чтение файлов с установленной мандатной категорией.

Проверка по данному пункту считается выполненной, если файл отчета `tests.log` содержит следующие строки:

```
---[fmac]: start test
PARSEC FMAC TEST: INFO: начинаем...
progname = /root/parsec-tests/fmac
PARSEC FMAC TEST: INFO: Начинаем тест: mac inheritance test...
PARSEC FMAC TEST: INFO:         Итерация 0.
...
PARSEC FMAC TEST: INFO:         Итерация 9.
PARSEC FMAC TEST: INFO: mac inheritance test прошел успешно
PARSEC FMAC TEST: INFO: Начинаем тест: mac set-get test...
PARSEC FMAC TEST: INFO:         Итерация 0.
...
PARSEC FMAC TEST: INFO:         Итерация 9.
PARSEC FMAC TEST: INFO: mac set-get test прошел успешно
PARSEC FMAC TEST: INFO: Начинаем тест: mac access test...
PARSEC FMAC TEST: INFO:         Итерация 0.
...
PARSEC FMAC TEST: INFO:         Итерация 9.
PARSEC FMAC TEST: INFO: mac access test прошел успешно
PARSEC FMAC TEST: INFO: ТЕСТ УСПЕШЕН!ОБЩИЙ СТАТУС = 0
Тест ПРОШЕЛ
---[fmac]: stop test
```

5.2. Механизм мандатного разграничения доступа к объектам IPC

Проверка механизма мандатного разграничения доступа к объектам IPC осуществляется в ходе выполнения автоматической процедуры тестирования подсистемы безопасности PARSEC (см. 2.1).

Проверяется механизм мандатного разграничения доступа к объектам IPC, включая семафоры, разделяемую память, очереди сообщений.

Проверка по данному пункту считается выполненной, если файл отчета `tests.log` содержит следующие строки:

```
---[ipc_mac]: start test
PARSEC IPC/SIGNAL TEST: INFO: start...
progname = /root/parsec-tests/ipc_mac
PARSEC IPC/SIGNAL TEST: INFO: Начинаем тест: mac IPC test...
PARSEC IPC/SIGNAL TEST: INFO:   Итерация 0.
...
PARSEC IPC/SIGNAL TEST: INFO:   Итерация 9.
PARSEC IPC/SIGNAL TEST: INFO: mac IPC test прошел успешно
PARSEC IPC/SIGNAL TEST: INFO: Начинаем тест: mac SignalS test...
PARSEC IPC/SIGNAL TEST: INFO:   Итерация 0.
...
PARSEC IPC/SIGNAL TEST: INFO:   Итерация 9.
PARSEC IPC/SIGNAL TEST: INFO: mac SignalS test прошел успешно
PARSEC IPC/SIGNAL TEST: INFO: ТЕСТ УСПЕШЕН!ОБЩИЙ СТАТУС = 0
Тест ПРОШЕЛ
---[ipc_mac]: stop test
```

5.3. Механизм мандатного разграничения доступа для сетевых взаимодействий

Проверка механизма мандатного разграничения доступа для сетевых взаимодействий осуществляется в ходе выполнения автоматической процедуры тестирования подсистемы безопасности `PARSEC` (см. 2.1).

Проверяется механизм мандатного разграничения доступа для сетевых взаимодействий, включая взаимодействия с использованием протокола UDP семейства TCP/IP (4 и 6 версий), протокола TCP семейства TCP/IP (4 и 6 версий), UNIX-сокетов.

Проверка по данному пункту считается выполненной, если файл отчета `tests.log` содержит следующие строки (например, для IPv4):

```
---[tcpip_mac.sh]: start test
PARSEC TCP/IP TEST: INFO: start...
progname = /root/parsec-tests/tcpip_mac
PARSEC TCP/IP TEST: INFO: Начинаем тест: mac tcp socket test...
PARSEC TCP/IP TEST: INFO:       Итерация 0.
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1272
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:45798 ...
```

```
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
...
   PARSEC TCP/IP TEST: INFO:       Итерация 1.
...
   PARSEC TCP/IP TEST: INFO:       Итерация 2.
...
PARSEC TCP/IP TEST: INFO: mac tcp socket test прошел успешно

PARSEC TCP/IP TEST: INFO: Начинаем тест: mac udp socket test...
PARSEC TCP/IP TEST: INFO:       Итерация 0.
PARSEC TCP/IP TEST: INFO:       Итерация 1.
PARSEC TCP/IP TEST: INFO:       Итерация 2.
PARSEC TCP/IP TEST: INFO: mac udp socket test прошел успешно

PARSEC TCP/IP TEST: INFO: Начинаем тест: mac unix stream socket test...
PARSEC TCP/IP TEST: INFO:       Итерация 0.
PARSEC TCP/IP TEST: INFO: ждем соединения на ƒайле unixfile
PARSEC TCP/IP TEST: INFO: соединение установлено...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
...
PARSEC TCP/IP TEST: INFO:       Итерация 2.
...
PARSEC TCP/IP TEST: INFO: mac unix stream socket test прошел успешно

PARSEC TCP/IP TEST: INFO: Начинаем тест: mac unix dgram socket test...
PARSEC TCP/IP TEST: INFO:       Итерация 0.
PARSEC TCP/IP TEST: INFO: ждем данные на ƒайле unixfile
...
PARSEC TCP/IP TEST: INFO:       Итерация 2.
...
PARSEC TCP/IP TEST: INFO: Начинаем тест: mac privilege socket set-get test...
PARSEC TCP/IP TEST: INFO:       Итерация 0.
PARSEC TCP/IP TEST: INFO:       Итерация 1.
PARSEC TCP/IP TEST: INFO:       Итерация 2.
PARSEC TCP/IP TEST: INFO: mac privilege socket set-get test прошел успешно

PARSEC TCP/IP TEST: INFO: Начинаем тест: mac privilage socket accept test...
PARSEC TCP/IP TEST: INFO:       Итерация 0.
```

```
PARSEC TCP/IP TEST: INFO: Начинаем тест для TCP...
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1275
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:35596 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
...
PARSEC TCP/IP TEST: INFO:       Итерация 1.
...
PARSEC TCP/IP TEST: INFO:       Итерация 2.
...
PARSEC TCP/IP TEST: INFO: mac privilage socket accept test прошел успешно


PARSEC TCP/IP TEST: INFO: ТЕСТ УСПЕШЕН!ОБЩИЙ СТАТУС = 0
Тест ПРОШЕЛ
---[tcpip_mac.sh]: stop test
```

5.4. Механизм мандатного разграничения доступа к объектам БД

Проверка механизма мандатного разграничения доступа к объектам БД осуществляется в ходе выполнения автоматической процедуры тестирования СУБД PostgreSQL (см. 2.2).

При этом проверяется разграничение доступа при выполнении операций работы с данными: выборки (см. 1.2.25, 1.2.16, 1.2.17, 1.2.18), добавления (см. 1.2.22), модификации (см. 1.2.29) и удаления (см. 1.2.21). Так же проверяется доступ к объектам БД: таблицам и представлениям (см. 1.2.27), столбцам таблиц и представлений (см. 1.2.15), большим объектам (см. 1.2.23) и последовательностям (см. 1.2.26, 1.2.36). Тестами (см. 1.2.19, 1.2.20, 1.2.30) проверяется создание и модификация объектов БД, а тестами (см. 1.2.11, 1.2.12, 1.2.13, 1.2.14, 1.2.32) — управление мандатными метками объектов БД и правилами разграничения доступа. Дополнительно проверяется сохранение мандатных меток при выполнении служебных функций СУБД (см. 1.2.31, 1.2.33, 1.2.35, 1.2.37). Поскольку СУБД позволяет пользователю создавать хранимые процедуры обработки данных, производится проверка мандатного разграничения доступа в хранимых процедурах и триггерах на языках PL/Perl, Untrusted PL/Perl, PL/pgSQL, Untrusted PL/Python, PL/Tcl и Untrusted PL/Tcl (см. 1.2.24, 1.2.28).

В ходе выполнения ряда перечисленных тестов осуществляется проверка автоматической маркировки объектов БД и защищаемых записей объектов БД, отражающих уровень их конфиденциальности. Проверка по данному пункту считается успешной при успешном выполнении указанных тестов в составе автоматической процедуры тестирования СУБД.

6. ПРОВЕРКА ОЧИСТКИ ПАМЯТИ И ИЗОЛЯЦИИ МОДУЛЕЙ

6.1. Механизмы работы с ОП

Проверка механизма работы с ОП осуществляется в ходе выполнения автоматической процедуры тестирования подсистемы безопасности PARSEC (см. 2.1).

Проверяются механизмы работы с ОП, включая копирование при записи и ее очистку и наличие для каждого процесса в системе собственного изолированного адресного пространства.

Проверка по данному пункту считается выполненной, если файл отчета tests.log содержит следующие строки:

```
---[mem_test]: начало теста
Тестирование механизма очистки памяти...Текущая граница сегмента
данных: 08064000
Новая граница сегмента данных: 08068000
Сигнатура 'Hello world!' @ 8067ff3
Граница сегмента после повторного выделения: 8068000
Сигнатура '' @ 8067ff3
УСПЕШНО
Тестирование механизма COW (копирование при записи)...Сигнатура 'Hello from
world #0' @ 0804b4a0, A
Сигнатура 'Hello from world #1' @ 0804b4a0, B
Сигнатура 'Hello from world #0' @ 0804b4a0, A (после завершения процесса B)
УСПЕШНО
Тест ПРОШЕЛ
---[mem_test]: завершение теста
```

6.2. Механизм очистки памяти внешних носителей

Проверка механизма очистки памяти внешних носителей осуществляется в ходе выполнения автоматической процедуры тестирования подсистемы безопасности PARSEC (см. 2.1).

Проверяется механизм очистки памяти внешних носителей, включая создание файла внутри ФС, проверку содержимого файла, удаление созданного файла и проверку наличия содержимого файла на жестком диске.

Проверка по данному пункту считается выполненной, если файл отчета tests.log содержит следующие строки:

```
---[secdelrm.sh]: начало теста
Создание образа диска...УСПЕШНО
Создание файла в файловой системе...УСПЕШНО
```

```
Проверка присутствия содержимого файла на диске...УСПЕШНО
Удаление файла и поиск содержимого файла на диске...УСПЕШНО
Тест ПРОШЕЛ
---[secdelrm.sh]: завершение теста
```

7. ПРОВЕРКА МАРКИРОВКИ ДОКУМЕНТОВ

Для проверки маркировки документов необходимо:

1) войти в систему как пользователь `root`;
2) запустить окно терминала;
3) ввести команды:

```
cd /opt/ossn-test/marker-test
./test.sh
```

4) ознакомиться с содержимым журнала пользователя с помощью команды:

```
userlog
```

Результат тестирования считается положительным, если после выполнения программы на экран выведен документ с напечатанной меткой, а в журнале содержится необходимая информация о факте печати.

8. ПРОВЕРКА ЗАЩИТЫ ВВОДА-ВЫВОДА ИНФОРМАЦИИ НА ОТЧУЖДАЕМЫЙ ФИЗИЧЕСКИЙ НОСИТЕЛЬ И СОПОСТАВЛЕНИЕ ПОЛЬЗОВАТЕЛЯ С УСТРОЙСТВОМ

Для проверки защиты ввода-вывода информации на отчуждаемый физический носитель и сопоставления пользователя с устройством необходимо:

1) войти в систему от имени пользователя `root`;

2) запустить окно терминала;

3) подключить съемный USB-носитель;

4) выполните команду:

```
dmesg | grep "Attached SCSI" | tail –n1
```

и определить имя файла устройства в каталоге `/dev`, соответствующего подключенному USB-носителю;

5) открыть файл `/etc/fstab` в редакторе командой:

```
mcedit /etc/fstab
```

6) добавить строку, предоставляющую пользователям право монтировать ФС подключенного USB-носителя:

```
/dev/sdb /media/floppy auto rw,user,noauto 0 0
```

7) создать ФС на USB-носителе командой:

```
mkfs.ext3 /dev/sdc
```

8) смонтировать ФС на USB-носителе во временную папку `/mnt` командой:

```
mount /dev/sdc /mnt
```

9) установить на корневой каталог ФС на USB-носителе требуемую мандатную метку и владельца командами:

```
chmac Уровень:Категория /mnt
chown Пользователь:Группа /mnt
```

10) размонтировать ФС на USB-носителе командой:

```
umount /mnt
```

11) войти в систему от имени пользователя;

12) запустить окно терминала;

13) смонтировать USB-носитель командой:

```
mount /media/floppy
```

14) убедиться в выполнении правил мандатного и дискреционного разграничения доступа для ФС подготовленного съемного USB-носителя, выполняя команды по созданию и удалению объектов ФС в точке монтирования USB-носителя и ниже.

Результат тестирования считается положительным, если не выявлено фактов нарушения правил разграничения доступа к объектам ФС на подготовленном съемном носителе.

При монтировании ФС носителя, которая не поддерживает хранение мандатных меток, точке монтирования носителя и всем вложенным объектам ФС присваивается метка с минимальным уровнем и пустым списком категорий `0:0`. Владельцем назначается пользователь, смонтировавший ФС.

9. ПРОВЕРКА РЕГИСТРАЦИИ СОБЫТИЙ

9.1. Система регистрации событий

Проверка система регистрации событий осуществляется в ходе выполнения автоматической процедуры тестирования подсистемы безопасности PARSEC (см. 2.1).

Проверяется система регистрации событий, включая установки флагов аудита на файл, создания событий аудита, запуска и остановки системы регистрации событий.

Проверка по данному пункту считается выполненной, если файл отчета tests.log содержит следующие строки:

```
---[audit.sh]: начало теста
Установка флагов аудита на файл /tmp/file-21096...УСПЕШНО
Создание события аудита /tmp/file-21096...УСПЕШНО
Остановка службы аудита...УСПЕШНО
Поиск сообщений аудита в журнале...УСПЕШНО
Запуск службы аудита...УСПЕШНО
Тест ПРОШЕЛ
---[audit.sh]: завершение теста
```

9.2. Регистрация событий при работе с БД

Тестирование системы регистрации событий (аудита) СУБД PostgreSQL проводится в полуавтоматическом режиме. Тестированию подвергается требование к регистрации событий и фиксируемой в сообщениях аудита информации, а также к наличию средств выборочного ознакомления с информацией.

При выполнении тестирования (см. 2.2) генерируются следующие виды событий:

– использование механизма идентификации и аутентификации;

– попытки доступа;

– действия выделенных пользователей;

– запрос на доступ к защищаемому ресурсу;

– создание и уничтожение объекта;

– действия по изменению ПРД.

Для просмотра сообщений аудита СУБД необходимо:

1) войти в систему как пользователь root;

2) запустить окно терминала;

3) выполнить команду:

```
userlog -e pgsql | more
```

Сообщение аудита, которое может выдать СУБД PostgreSQL:

```
Mar  7 18:44:01 eutm postgres[8015]: [3-0] AUDIT: SUCCESS, CONNECT, 127.0.0.1,
```

```
"template1", SU = "postgres" (1), CU = "postgres" (1): successfull connection
```

Часть записи «Mar 7 18:44:01 eutm postgres[8015]:» генерируется сервисом `syslog` и сообщает общую информацию о дате и времени наступления события, узле, переславшем сообщение, названии программы (в данном случае это имя исполняемого модуля сервера СУБД), идентификаторе процесса СУБД (PID), обслуживающего клиентское соединение (здесь 8015).

Часть «[3-0]» указывает на «степень разделения сообщения». Сервис `syslog` накладывает ограничение на длину передаваемого сообщения, поэтому некоторые сообщения приходится делить на части. Число 3 здесь означает порядковый номер выданного пользователю сообщения, а число 0 — порядковый номер составной части сообщения. Если второе число есть 0, то сообщение не билось на части. Если же сообщение бьется на части, то нумерация второго числа начинается с 1.

Часть «AUDIT:» указывает на то, что сообщение от процесса с именем `postgres` есть сообщение аудита, а не какая-либо отладочная системная информация. После «AUDIT:» следует типовая информация, а именно:

– успешность осуществления события (`SUCCESS` или `FAILURE`);

– тип события (`CONNECT`, `DISCONNECT`, `SUBJECT`, `RIGHTS` и т. д.);

– хост, с которого пришел клиентский запрос (`127.0.0.1`);

– имя БД, с которой работают (`template1`);

– имя авторизованного пользователя и идентификатор (`SU = "postgres" (1)`);

– имя текущего пользователя и идентификатор при смене идентификатора пользователя (`CU = "postgres" (1)`);

– информация об объекте или действии (`successfull connection`).

При проведении тестирования можно также наблюдать генерацию сообщения аудита PostgreSQL в интерактивном режиме, если в консоли пользователя `root` выполнить следующую команду:

```
tail -f /var/log/messages | grep -e postgres -e AUDIT
```

При дальнейшей передаче SQL-команд в СУБД все сообщения аудита от СУБД PostgreSQL будут выдаваться в эту консоль практически в реальном режиме времени.

10. ПРОВЕРКА НАДЕЖНОГО ВОССТАНОВЛЕНИЯ

10.1. Механизм надежного восстановления ФС

Для восстановления ФС в результате аппаратного сбоя (например, в случае проблем с электропитанием) используется программа `fsck`. В случае аварийного завершения работы системы при следующей загрузке запуск этой программы будет произведен автоматически. Для проверки работы механизма надежного восстановления необходимо:

1) загрузить систему;
2) выключить питание (имитация сбоя электропитания);
3) включить питание системы. Дождаться завершения проверки дисковой подсистемы. Если это необходимо, следуя дальнейшим инструкциям, произвести вход в систему и повторить проверку.

Факт возможности входа в систему после выключения и включения питания и последующего автоматического восстановления системы программой `fsck` означает успешное завершение данного теста.

10.2. Механизм надежного восстановления БД

Проверка механизма надежного восстановления БД осуществляется в ходе выполнения автоматической процедуры тестирования СУБД PostgreSQL (см. 1.2.38).

Проверка по данному пункту считается успешной при успешном выполнении указанного теста в составе автоматической процедуры тестирования СУБД.

11. ПРОВЕРКА КОНТРОЛЯ ЦЕЛОСТНОСТИ КСЗ

Для контроля целостности объектов ФС используется программа `afick`. Для проверки работы механизма, осуществляющего контроль за целостностью СВТ, необходимо:

1) войти в систему как пользователь `root`
2) запустить окно терминала;
3) выполнить команду:

```
afick -i
```

Необходимо подождать, пока построится первоначальная БД;

4) произвести намеренные изменения в ФС:

```
chmac 1:0 /etc/fstab
cp /bin/su /bin/su.bak
echo asdf >> /bin/su
```

5) запустить программу контроля целостности в режиме проверки с помощью команды:

```
afick -k
```

Результат тестирования считается положительным, если после выполнения программы на экран выведена информация об измении файла `/bin/su` и об изменении мандатной метки у файла `/etc/fstab`.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

БД — база данных
КСЗ — комплекс средств защиты
НСД — несанкционированный доступ
ОП — оперативная память
ОС — операционная система
ПРД — правила разграничения доступа
СУБД — система управления базами данных
ФС — файловая система

IP — Internet Protocol (протокол Интернет)
IPC — InterProcess Communication (межпроцессное взаимодействие)
MAC — Mandatory Access Control (мандатное управление доступом)
PID — Process Identifier (идентификатор процесса)
SQL — Structured Query Language (язык структурированных запросов)
TCP — Transmission Control Protocol (протокол передачи данных)
UDP — User Datagram Protocol (протокол пользовательских дейтаграмм)

Лист регистрации изменений									
Изм.	Номера листов (страниц)				Всего листов (страниц) в докум.	№ документа	Входящий № сопроводительного докум. и дата	Подп.	Дата
	измененных	замененных	новых	аннулированных					