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 63

2024

Литера $О_1$

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 управления доступом, защищенной СУБД, 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. ПОРЯДОК ПРОВЕДЕНИЯ ТЕСТИРОВАНИЯ

Тестирование предназначено для проверки корректности работы функций КСЗ в рамках регламентных работ.

Перед проведением тестирования информационная (автоматизированная) система должна быть выведена из эксплуатации, т.к. в процессе тестирования меняются параметры работы средств защиты информации, параметры объектов и субъектов доступа, что может вызвать ошибки и сбои в работе системного и прикладного ПО, угрозы нарушения конфиденциальности и доступности информации.

После завершения тестирования все параметры возвращаются в исходное состояние и система может быть введена в эксплуатацию.

2. СТРУКТУРА ТЕСТОВ

Тестирование КСЗ заключается в проверке функционирования подсистем, реализующих функции безопасности ОС. В части, касающейся функций ядра ОС по защите от НСД, тестируется подсистема безопасности PARSEC, а в части, касающейся функций защиты СУБД, — СУБД Tantor (в исполнении Basic), доработанная в соответствии с требованием интеграции с ОС в части защиты информации, в том числе мандатного управления доступом.

2.1. Порядок проведения тестирования

Тестирование предназначено для полной проверки функций КСЗ.

Тестирование должно выполняться для проверки корректности работы функций КСЗ в процессе регламентных работ, при этом система должна быть выведена из эксплуатации.

Не допускается проведение тестирования на эксплуатируемой системе, т.к. в процессе тестирования меняются параметры работы системы, а также параметры объектов и субъектов системы, что может вызвать ошибки в работе прикладного ПО, а также угрозы информационной безопасности. После завершения тестирования все параметры возвращаются в исходное состояние и система может быть введена в эксплуатацию.

2.2. Подсистема безопасности PARSEC

Тестирование подсистемы безопасности PARSEC производится тестовой системой `parsec-tests`. Она состоит из набора тестов для тестирования механизмов:

1) дискреционного управления доступом к объектам ФС;
2) мандатного управления доступом к объектам ФС, IPC и при сетевых взаимодействиях;
3) работы с памятью и изоляции модулей;
4) очистки памяти внешних носителей;
5) привилегий процесса;
6) подсистемы регистрации событий.

Все тесты имеют опцию «Число итераций», позволяющую настраивать число проходов теста, на каждом из которых на процесс устанавливается случайная метка безопасности.

2.2.1. Модуль тестирования механизма дискреционного управления доступом к объектам ФС

Реализован в виде сценариев `rwx.sh` и `acl.sh`.

В сценарии `rwx.sh` реализована 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 входящего в группу пользователя-владельца файла.

В сценарии `acl.sh` реализована проверка чтения и записи в объекты ФС с правами доступа предоставляемыми через списки контроля доступа (access control list — ACL) для пользователя, групп пользователей и остальных пользователей в том числе с ограничением распространения прав через маску.

2.2.2. Модуль тестирования механизма мандатного управления доступом к объектам ФС

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

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

2.2.3. Модуль тестирования механизма дискреционного управления доступом к объектам IPC

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

1) доступ процесса к данному объекту IPC (семафор, разделяемая память, очередь сообщения) методом получения процессом идентификатора данного IPC;
2) посылку сигнала из процесса (нити), функционирующего в контексте владельца объекта IPC или в контексте иного пользователя.

2.2.4. Модуль тестирования механизма мандатного управления доступом к объектам IPC

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

1) доступ процесса к данному объекту IPC (семафор, разделяемая память, очередь сообщения) методом получения процессом идентификатора данного IPC;
2) посылку сигнала процессу (нити) с данной меткой безопасности `M1` от процесса (нити) с меткой безопасности `M2`.

2.2.5. Модуль тестирования механизма мандатного управления доступом при сетевых взаимодействиях

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

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

Аналогичные проверки осуществляются для привилегированного сокета.

2.2.6. Модуль тестирования механизмов работы с памятью и изоляции процессов

Реализован в виде теста `mem_test`.

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

2.2.7. 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 `secdelrnd` после удаления данного файла. В тесте поддерживаются ФС ext2/ext3/ext4/xfs.

2.2.8. 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 из администратора в обычного пользователя;
2) получение полных Linux- и PARSEC-привилегий процессом при включенном флаге процесса `PR_KEEP_CAPS` после переключения обычного пользователя в администратора;
3) наследование наследуемых Linux- и PARSEC-привилегий процессом.

2.2.9. Модуль тестирования подсистемы регистрации событий

Реализован в виде тестов `audit_file.sh` и `audit_proc.sh`.

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

2.2.10. Модуль тестирования механизма мандатного контроля целостности

Реализован в виде теста `mictest.sh`. Осуществляет следующие проверки:

1) разграничение доступа к файлам в соответствии с правилами МКЦ;
2) функционирование привилегии `PARSEC_CAP_IGNMACINT`;
3) ограничение отправки сигналов между процессами;
4) изменение уровня МКЦ;
5) функционирование метки целостности на юнитах systemd.

2.2.11. Модуль тестирования механизма управления метками безопасности

Реализован в виде теста `chlbl.sh`. Осуществляет проверки установки/снятия меток безопасности объектов в зависимости от:

1) типа объекта;
2) наличия прав суперпользователя `root`;
3) наличия привилегии `PARSEC_CAP_CHMAC`;
4) состояния МКЦ в системе (включено/выключено).

2.2.12. Модуль тестирования механизма фильтрации списка содержимого каталогов

Реализован в виде теста `iterate_dir.sh`. Проверка фильтрации списка содержимого каталогов для объектов ФС ext4/xfs, имеющих ненулевую классификационную метку.

2.2.13. Модуль тестирования механизма преобразования меток безопасности

Реализован в виде теста `pdpl_test.sh`. Проверяет корректность преобразования меток безопасности из формата, в котором они хранятся на диске (в расширенных атрибутах файлов), в формат внутреннего представления.

2.3. СУБД

В СУБД реализованы следующие функции по защите от НСД:

1) дискреционное управление доступом к объектам БД;
2) мандатное управление доступом к объектам и данным БД;
3) взаимодействие пользователя с КСЗ;

4) идентификация и аутентификация;

5) надежное восстановление;

6) регистрация;

7) тестирование.

В состав СУБД входит пакет `postgresql-se-test-x.x` (где в качестве `x.x` используется конкретная версия СУБД), содержащий тесты дополнительных функциональных возможностей по разграничению доступа. При его установке в каталог `/usr/share/postgresql/x.x/test/` помещается каталог `pgacext`, содержащий необходимые SQL-скрипты, вспомогательные скрипты и эталоны результатов для выполнения тестов.

Тестирование осуществляется путем создания тестового кластера БД и выполнения SQL-скриптов с запросами, относящимся 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.

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

- `runtests` — вспомогательный скрипт для запуска процесса тестирования;

- `tests.lst` — полный список тестов, доступных для текущей версии СУБД. Может быть изменен для проведения выборочного тестирования;

- `runsetests` — вспомогательный скрипт для запуска процесса тестирования дополнительных функциональных возможностей.

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

Продолжение таблицы 1

Тест	Описание
acl-dblink	Проверка дискреционного метода контроля доступа при работе с расширением dblink
acl-foreign	Проверка дискреционного метода контроля доступа при работе с внешними таблицами
acl-function	Проверка дискреционного метода контроля доступа к функциям (хранимым процедурам)
acl-language	Проверка дискреционного метода контроля доступа к процедурным языкам
acl-largeobject	Проверка дискреционного метода контроля доступа к бинарным объектам
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-type	Проверка дискреционного метода контроля доступа к типам и доменам
acl-tablespace	Проверка дискреционного метода контроля доступа к областям хранения данных
acl-view	Проверка дискреционного метода контроля доступа к видам
mac-alter-meta	Проверка мандатного метода контроля доступа при модификации метаданных
mac-altermac	Проверка изменения мандатных атрибутов объектов
mac-chmac-constraints	Проверка работы ограничений целостности мандатных атрибутов CHMAC для внешних ключей
mac-chmac	Проверка работы команды санкционированного изменения меток безопасности данных CHMAC, правил и триггеров для этой операции
mac-copy-file-deny	Проверка запрета работы команды COPY при выводе данных в файл
mac-copy-file	Проверка работы команды COPY при работе с файлами
mac-copy-std	Проверка работы команды COPY при работе со стандартными потоками
mac-create	Проверка создания объектов и автоматического назначения меток безопасности
mac-createtableas	Проверка создания объектов командой CREATE TABLE AS
mac-dblink	Проверка мандатного метода контроля доступа при работе с расширением dblink
mac-delete	Проверка мандатного метода контроля доступа при удалении

Продолжение таблицы 1

Тест	Описание
mac-dp	Проверка ограничений ДП-модели
mac-foreign	Проверка мандатного метода контроля доступа при работе с внешними таблицами
mac-indexes	Проверка работы индексов совместно с системой защиты
mac-inherit	Проверка наследования мандатных атрибутов таблиц
mac-insert	Проверка мандатного метода контроля доступа при вставке
mac-joins	Проверка мандатных ПРД при соединениях таблиц (NestLoop, HashJoin, MergeJoin)
mac-largeobject	Проверка мандатного метода контроля доступа при работе с бинарными объектами
mac-materializedview	Проверка мандатного метода контроля доступа при доступе к материализованным видам
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-audit	Проверка работы модуля аудита в различных конфигурациях
misc-cluster	Проверка сохранности правил разграничения доступа при оптимизации индексов таблицы
misc-config	Проверка конфигурационных параметров КСЗ
misc-dump-restore	Проверка резервирования/восстановления с помощью утилит pg_dump/pg_restore
misc-dynamic-queries	Проверка сохранности правил разграничения доступа при использовании динамических запросов

Окончание таблицы 1

Тест	Описание
`misc-maclabel`	Проверка работы встроенных функций с типом `maclabel` («метка безопасности»)
`misc-memory-check`	Проверка очистки высвобождаемой памяти случайными значениями
`misc-policy`	Проверка механизма фильтрации строк средствами системы фильтрации (POLICY)
`misc-psql-d`	Проверка работы метакоманды `\d` утилиты `psql`
`misc-role-control`	Проверка контроля ролей: проверка механизма временного пароля, проверка механизма ограничения количества сессий, проверка механизма блокировки роли входа
`misc-roles`	Проверка ролевого разграничения доступа
`misc-rules`	Проверка мандатного метода контроля доступа при использовании правил
`misc-sql-types`	Проверка SQL-типов данных
`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 x.x;
- процедурный язык PL/Perl для PostgreSQL x.x;
- процедурный язык PL/Python для PostgreSQL x.x;
- процедурный язык PL/Tcl для PostgreSQL x.x.

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 метками безопасности и наборами привилегий к защищенным метками безопасности 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 управление доступом к объектам БД;

- мандатное управление доступом 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.

Так же проверяются системные функции СУБД, такие как индексирование, резервирование/восстановление, процедуры и триггеры. При этом системные функции и механизмы контроля доступа не должны влиять друг на друга.

2.3.1. acl-column

Проверка дискреционного управления доступом 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.

2.3.2. acl-database

Проверка дискреционного управления доступом 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.

2.3.3. acl-dblink

Проверка дискреционного управления доступом при работе с расширением dblink заключается в последовательном назначении и отборе прав на использование расширения, а также прав на исполнение процедур, входящие в него. При этом оценивается соответствие результата предоставления доступа диспетчером доступа СУБД (успех или отказ) установленным правам пользователя.

2.3.4. acl-foreign

Проверка дискреционного управления доступом при работе с внешними таблицами заключается в последовательном назначении и отборе на использование внешних объектов (серверов, оберток, таблиц). При этом оценивается соответствие результата предоставления доступа диспетчером доступа СУБД (успех или отказ) установленным правам пользователя.

2.3.5. acl-function

Проверка дискреционного управления доступом 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.

2.3.6. acl-language

Проверка дискреционного управления доступом 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.

2.3.7. acl-largeobject

Проверка дискреционного управления доступом к бинарным объектам заключается в создании бинарного объекта, последовательном назначении и отборе прав на бинарный объект и попытках выполнения операций с ним. При этом оценивается соответствие результата предоставления доступа диспетчером доступа СУБД (успех или отказ) установленным правам пользователя.

2.3.8. acl-role

Проверка дискреционного управления доступом 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).

2.3.9. acl-schema

Проверка дискреционного управления доступом 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.

2.3.10. acl-sequence

Проверка дискреционного управления доступом 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.

2.3.11. acl-table

Проверка дискреционного управления доступом 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.

2.3.12. acl-type

Проверка дискреционного управления доступом к типам и доменам заключается в последовательном назначении и отборе прав доступа пользователя к типам при создании, модификации типов и доменов. При этом оценивается соответствие результата предоставления доступа диспетчером доступа СУБД (успех или отказ) установленным правам пользователя.

2.3.13. acl-tablespace

Проверка дискреционного управления доступом 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.

2.3.14. acl-view

Проверка дискреционного управления доступом 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.

2.3.15. mac-alter-meta

Проверка мандатного управления доступом 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.

2.3.16. mac-altermac

Проверка изменения мандатных атрибутов объектов заключается в попытках установки и изменения меток безопасности объекта и мандатных атрибутов CCR (Container Clearance Required) контейнеров. При этом оценивается возможность изменения мандатных атрибутов владельцем объекта, если он имеет привилегию изменения метки безопасности, и невозможность изменения мандатных атрибутов у объекта другого пользователя.

2.3.17. mac-chmac-constraints

Проверка работы ограничений целостности `CHMAC` для внешних ключей заключается в создании таблиц, связанных ограничением целостности `ON CHMAC`, и попытках изменения меток безопасности командой `CHMAC`. При этом проверяется соответствие меток записей тестируемых таблиц, связанных ограничением целостности.

2.3.18. mac-chmac

Проверка изменения меток безопасности командой `CHMAC`, а так же правильность работы правил и триггеров для этой операции (`ON CHMAC`) заключается создании тестовых объектов, правил и триггеров и попытках установки и изменения меток безопасности записей. При этом оценивается возможность выполнения команды `CHMAC`, если он имеет привилегию изменения метки безопасности, и корректность работы механизмов правил и триггеров для этой операции.

2.3.19. 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.

2.3.20. 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 метка безопасности данных назначается результирующему файлу с данными при наличии у системного пользователя `postgres` привилегии назначения меток безопасности.

2.3.21. 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.

2.3.22. mac-create

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

2.3.23. mac-createtableas

Проверка создания объектов командой `CREATE TABLE AS` (и аналогичными: `CREATE TABLE ... (LIKE ...)`, `SELECT ... INTO ...`) заключается в создании объектов БД данными командами на основе существующего объекта с данными. При этом оценивается соответствие метки безопасности, назначенной создаваемым или изменяемым объектам, и текущей метки безопасности сессии пользователя, а также заполнение создаваемых объектов данными в соответствии с правилами мандатного управления доступом.

2.3.24. mac-dblink

Проверка мандатного метода контроля доступа при работе с расширением `dblink` заключается в проверке корректности передачи метки безопасности пользователя на удаленную базу данных и разграничении доступа в удаленной базе данных. При этом оценивается соответствие метки безопасности, назначенной создаваемым объектом, и текущей метки безопасности сессии пользователя.

2.3.25. mac-dp

Проверка ограничений ДП-модели заключается в последовательных попытках назначить метку безопасности объекта, превышающую метку безопасности контейнера, в котором он содержится и в попытках назначить метку безопасности контейнеру, меньшую, чем метка безопасности объекта, содержащегося в этом контейнере. При этом оценивается соответствие результата предоставления доступа диспетчером доступа СУБД (успех или отказ) установленным меткам безопасности и наборам привилегий пользователей.

2.3.26. mac-delete

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

2.3.27. mac-foreign

Проверка мандатного метода контроля доступа при работе с внешними таблицами заключается в последовательных попытках выбора, изменения, модификации и удаления данных таблиц, доступ к которым предоставляется посредством внешних таблицы. При этом оценивается соответствие результатов предоставления доступа диспетчером доступа СУБД (успех или отказ) установленным меткам безопасности и наборам привилегий пользователей.

2.3.28. mac-indexes

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

2.3.29. mac-inherit

Проверка наследования мандатных атрибутов таблиц заключается в проверке соответствия мандатных атрибутов родительской таблицы и мандатных атрибутов таблиц, унаследованных от родительской. При этом оценивается соответствие результатов предоставления доступа диспетчером доступа СУБД (успех или отказ) установленным меткам безопасности и наборам привилегий пользователей.

2.3.30. mac-insert

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

2.3.31. mac-joins

Проверка работы объединений таблиц в запросах заключается в последовательном выполнении запросов на вставку и чтение данных пользователями с разными метками безопасности

и наборами привилегий к защищенным и незащищенным метками безопасности таблицам. При этом оценивается соответствие результатов предоставления доступа диспетчером доступа СУБД установленным меткам безопасности и наборам привилегий пользователей при различных методах соединения таблиц (NextLoop, HashJoin, MergeJoin).

2.3.32. mac-largeobject

Проверка мандатного управления доступом при работе с бинарными объектами заключается в создании бинарного объекта и в последовательных попытках выполнения операций с ним пользователями с разными метками безопасности и наборами привилегий. При этом оценивается соответствие метки безопасности, назначенной создаваемому бинарному объекту, и текущей метки безопасности сессии пользователя и результатов предоставления доступа диспетчером доступа СУБД (успех или отказ) установленным меткам безопасности и наборам привилегий пользователей.

2.3.33. mac-materializedview

Проверка мандатного управления доступом при доступе к материализованным видам заключается в последовательном выполнении запросов на создание и обновление материализованных видов пользователями с разными метками безопасности и наборами привилегий. При этом оценивается соответствие результатов предоставления доступа диспетчером доступа СУБД (успех или отказ) установленным меткам безопасности и наборам привилегий пользователей.

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

Проверка мандатного управления доступом 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.

2.3.35. mac-select

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

2.3.36. mac-sequence

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

2.3.37. mac-tableview

Проверка мандатного управления доступом к таблицам и видам заключается в последовательном выполнении запросов на чтение, вставку, модификацию и удаление данных пользователями с разными метками безопасности и наборами привилегий к защищенным метками безопасности 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 меткам безопасности и наборам привилегий пользователей.

2.3.38. mac-triggers-perl, mac-triggers-perlu, mac-triggers-pgsql, mac-triggers-pythonu, mac-triggers-tcl, mac-triggers-tclu

Проверка мандатного управления доступом 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. Также проверяется реализация правил мандатного управления доступом при осуществлении доступа к данным объекта, который вызывает триггер, и другим объектам БД, и возможность изменения метки безопасности обрабатываемых данных пользователем с соответствующими привилегиями мандатного доступа.

2.3.39. mac-update

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

2.3.40. misc-audit

Проверка работы модуля аудита в различных конфигурациях заключается в проверке назначения маски аудита для пользователя в разных режимах работы модуля аудита: внешнем (конфигурационный файл `pg_audit.conf`), внутреннем (с помощью запросов SQL), смешанном, а также в неактивном режиме. При этом оценивается корректность назначения маски аудита согласно приоритету настроек и режиму работы модуля аудита.

2.3.41. 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. 2.3.35) до и после выполнения модификаций.

2.3.42. 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. 2.3.35) до и после выполнения операции.

2.3.43. 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 бинарных объектах, защищаются меткой безопасности. Отдельно проверяется возможность удаленного изменения конфигурационных параметров сервера, требующих его перезапуска.

2.3.44. misc-dump-restore

Проверка резервирования/восстановления с помощью утилит `pg_dump`/`pg_restore` заключается попытках резервирования существующих данных и их восстановления с последующим контролем выполнением тестов (см. 2.3.35). При этом резервирование осуществляется как в тестовом, так и в бинарном режимах.

2.3.45. misc-dynamic-queries

Проверка сохранности правил разграничения доступа при использовании динамических запросов заключается в создании хранимой процедуры, формирующей план выполнения трансформированного запроса, часть которого передается в качестве параметра, и после-

довательного исполнения ее от имени пользователей с разными метками безопасности и наборами привилегий, при этом подсчитывается количество получаемых записей. При этом оценивается соответствие результатов предоставления доступа диспетчером доступа СУБД установленным меткам безопасности и наборам привилегий пользователей.

2.3.46. misc-maclabel

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

2.3.47. misc-memory-check

Проверка механизма очистки высвобождаемой памяти случайными значениями включает в себя:

- выполнение запроса, содержащего контрольную фразу;
- сохранение высвобождаемых данных в файл `out.txt`;
- анализ файла `out.txt` на наличие контрольной фразы с использованием инструмента командной строки `grep`.

Проверка выполняется два раза:

- с выключенной функцией очистки высвобождаемой памяти (для конфигурационного параметра `ac_rand_free` задано значение `false`);
- с включенной функцией очистки высвобождаемой памяти (для конфигурационного параметра `ac_rand_free` задано значение `true`).

Проверяется, что при `ac_rand_free = false` вывод команды `grep` содержит один или более экземпляров контрольной фразы, а при `ac_rand_free = true` вывод команды `grep` пустой, так как файл `out.txt` содержит случайные значения.

2.3.48. misc-policy

Проверка механизма фильтрации строк средствами системы фильтрации (POLICY) (ROW LEVEL SECURITY). Ограничение набора строк таблицы, выдаваемых пользователю, может осуществляться с помощью дополнительной логики, определяемой для каждой таблицы. При этом оценивается правильность выдачи информации различным пользователям.

2.3.49. misc-psql-d

Проверка работы метакоманды `\d` утилиты `\psql`, выводящей сведения об объектах БД. Проверка заключается создании защищенных метками объектов и в получении информации о них с помощью указанной метакоманды.

2.3.50. misc-role-control

Проверка механизма временного пароля заключается в создании роли входа с временным паролем. При этом пользователю разрешается подключиться к базе до истечения времени действия пароля, а после — нет.

Проверка механизма ограничения количества сессий заключается в создании роли с ограничением по количеству сессий. При этом попытки создать дополнительные сессии будут завершаться ошибкой.

Проверка механизма блокировки роли входа заключается в создании пользователя, которому запрещается подключаться к базе. Попытка подключения с заблокированной ролью завершится ошибкой.

2.3.51. misc-roles

Проверка ролевого разграничения доступа заключается в последовательном выполнении запросов на управление ролями пользователями с привилегиями администрирования и без. При этом оценивается соответствие результатов предоставления доступа диспетчером доступа СУБД (успех или отказ) наборам привилегий пользователей и ролей.

2.3.52. misc-rules

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

2.3.53. misc-sql-types

Проверка поддержки СУБД типов данных SQL. Проверка заключается в попытках создания таблиц со столбцами проверяемых типов данных.

2.3.54. misc-vacuum

Проверка взаимодействия задач технического обслуживания данных с системой защиты заключается в выполнении операции `VACUUM`. При этом мандатные атрибуты существующих в таблице данных не должны изменяться, что проверяется выполнением тестов (см. 2.3.35) до и после выполнения операции.

2.3.55. 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.

2.3.56. mac-declarative-part

Проверка механизма мандатного управления доступом при декларативном секционировании заключается в создании таблицы с зависимыми от нее таблицами и изменении метки на родительской таблице. При этом проверяется наследование мандатной метки, признаков `CCR` и `MACS` родительской таблицы на дочерних таблицах при всех операциях изменения над родительской таблицей.

2.3.57. mac-files

Проверка установки мандатных атрибутов на файлы базы данных. При этом проверяется соответствие мандатных атрибутов объекта базы данных и всех файлов, принадлежащих данному объекту в файловой системе.

2.3.58. mac-replication-logical

Проверка мандатного управления доступом при логической репликации. При этом проверяются соблюдение правил разграничения доступа для объектов баз данных при взаимодействии двух серверов с помощью механизмов логической репликации.

2.3.59. misc-memory-wiping

Проверка механизмов очистки памяти заключается в добавлении, изменении и удалении данных в таблице и самой таблицы. При этом проверяется очистка данных из файлов при их изменении или удалении из таблицы.

2.3.60. misc-notify

Проверка работы механизмов уведомлений с сообщениями. При этом проверяется невозможность несанкционированной передачи информации между уровнями.

3. ПРОВЕДЕНИЕ ТЕСТИРОВАНИЯ

3.1. Подсистема безопасности PARSEC

Для запуска автоматической процедуры тестирования необходимо:

1) войти в систему от имени администратора;
2) зайти в каталог `/usr/lib/parsec/tests` и осуществить запуск скрипта:

```
sudo ./run.sh
```

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

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

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

```
Тест ПРОШЕЛ
```

в файле будет содержаться строка:

```
[!] ОШИБКА тестирования
```

3.2. СУБД

Для запуска автоматической процедуры тестирования необходимо:

1) войти в систему от имени администратора с высокой меткой целостности;
2) запустить окно терминала;
3) установить пакет тестирования выбранной версии СУБД командой:

```
sudo apt install postgresql-se-test-x.x
```

4) установить на каталог `/tmp` необходимые для выполнения тестирования мандатные атрибуты:

```
sudo pdpl-file 3:0:-1:ccnr /tmp
```

5) перейти в каталог `/usr/share/postgresql/x.x/test/pgacext/` командой:

```
cd /usr/share/postgresql/x.x/test/pgacext/
```

6) запустить тесты командой:

```
sudo ./runtests -all
```

7) сбросить мандатные атрибуты каталога `/tmp`:

```
sudo pdpl-file 0 /tmp
```

Скрипт запуска тестов СУБД `runtests.sh` имеет несколько вариантов запуска:

- `-all` — запуск всех тестов;
- `-acl` — запуск тестов проверки дискреционного управления доступом;
- `-mac` — запуск тестов проверки мандатного управления доступом;
- `-f ФАЙЛ` — запуск указанного теста.

Пример

Запуск теста проверки дискреционного управления доступом к базам данных:

```
sudo ./runtests -f sql/acl-database.sql
```

- `-i ФАЙЛ_СО_СПИСОМ ОПЦИЙ` — запуск только указанных в файле тестов. Полный список тестов находится в каталоге `/usr/share/postgresql/x.x/test/pgacext/tests.lst`. Если имя теста в файле закомментировано с помощью символа #, то данный тест исполняться не будет.

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

Успешность выполнения каждого теста подтверждается сообщением:

```
успех
```

Проверка по пункту считается успешной, если после выполнения программы на экране монитора появится сообщение (где `N` общее количество выполненных тестов):

```
Всего = N, запущено = N, ошибочных = 0
```

4. ПРОВЕРКА ИДЕНТИФИКАЦИИ И АУТЕНТИФИКАЦИИ

4.1. Идентификация и аутентификация

Для проверки идентификации и аутентификации необходимо:

1) войти в систему от имени администратора;
2) добавить в систему пользователя `test` командой:

```
sudo adduser test
```

3) задать пароль пользователю `test` командой:

```
sudo 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 администратора;
8) набрать команду в терминале:

```
sudo tail /var/log/auth.log
```

Будет выведен фрагмент журнала безопасности. Факт аутентификации пользователя отражен в строке:

```
pam_unix(login:session): session opened for user test
```

Информация о зарегистрированных пользователях системы содержится в файлах конфигурации. Изменять эти файлы может только администратор через механизм `sudo`.

4.2. 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 администратора и набрать команду:

```
sudo tail /var/log/auth
```

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

4.3. Идентификация и аутентификация при работе с БД

Проверка идентификации и аутентификации при работе с БД осуществляется в автоматической процедуре тестирования СУБД (см. 3.2).

Проверка по данному пункту считается успешной при успешном выполнении автоматической процедуры тестирования СУБД, в частности теста доступа к БД (см. 2.3.2).

5. ПРОВЕРКА ДИСКРЕЦИОННОГО УПРАВЛЕНИЯ ДОСТУПОМ

5.1. Механизм дискреционного управления доступом к объектам ФС

Проверка механизма дискреционного управления доступом к объектам ФС осуществляется в ходе выполнения автоматической процедуры тестирования подсистемы безопасности PARSEC (см. 3.1).

Проверяется механизм дискреционного управления доступом 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 считается успешной, если файл отчета `tests.log` содержит следующие строки:

```
---[rwx.sh]: start test
# Проверка механизма файловой системы RWX
Проверка чтения файла владельцем...УСПЕШНО
Проверка записи для владельца...УСПЕШНО
Проверка чтения для группы владельца...УСПЕШНО
Проверка записи для группы владельца...УСПЕШНО
Проверка чтения для других...УСПЕШНО
Проверка записи для других...УСПЕШНО
Test PASS
---[rwx.sh]: stop test

---[acl.sh]: start test
# Testing correct Acl RWX mechanism
Выставление ACL для владельца...проверка битовой маскиУСПЕШНО
Выставление битовой маски для владельца...проверка ACLУСПЕШНО
Выставление ACL для группы...проверка битовой маскиУСПЕШНО
Выставление битовой маски для группы...проверка ACLУСПЕШНО
Выставление ACL для прочих...проверка битовой маскиУСПЕШНО
Выставление битовой маски для прочих...проверка ACLУСПЕШНО
Test PASS
---[acl.sh]: stop test

---[ipc_dac]: start test
PARSEC IPC/SIGNAL TEST: INFO: start...
progname = /usr/lib/parsec/tests/ipc_dac

DAC IPC test: INFO: Начинаем тест: PARSEC IPC/SIGNAL TEST...

DAC IPC test: INFO:  Итерация 0.
```

```
DAC IPC test: INFO:  Итерация 1.

DAC IPC test: INFO:  Итерация 2.

DAC IPC test: INFO:  Итерация 3.

DAC IPC test: INFO:  Итерация 4.

DAC IPC test: INFO:  Итерация 5.

DAC IPC test: INFO:  Итерация 6.

DAC IPC test: INFO:  Итерация 7.

DAC IPC test: INFO:  Итерация 8.

DAC IPC test: INFO:  Итерация 9.
DAC IPC test: INFO: DAC IPC test прошел успешно

DAC Signals test: INFO: Начинаем тест: PARSEC IPC/SIGNAL TEST...

DAC Signals test: INFO:  Итерация 0.

DAC Signals test: INFO:  Итерация 1.

DAC Signals test: INFO:  Итерация 2.

DAC Signals test: INFO:  Итерация 3.

DAC Signals test: INFO:  Итерация 4.

DAC Signals test: INFO:  Итерация 5.

DAC Signals test: INFO:  Итерация 6.

DAC Signals test: INFO:  Итерация 7.

DAC Signals test: INFO:  Итерация 8.

DAC Signals test: INFO:  Итерация 9.
DAC Signals test: INFO: DAC Signals test прошел успешно

PARSEC IPC/SIGNAL TEST: INFO: ТЕСТ УСПЕШЕН!ОБЩИЙ СТАТУС = 0
Test PASS
```

```
---[ipc_dac]: stop test
```

5.2. Механизм дискреционного управления доступом к объектам БД

Проверка механизма дискреционного управления доступом к объектам БД осуществляется в ходе выполнения автоматической процедуры тестирования СУБД (см. 3.2).

При этом проверяется доступ к объектам БД: базам данных (см. 2.3.2), таблицам (см. 2.3.11), представлениям (см. 2.3.14), столбцам таблиц и представлений (см. 2.3.1), языкам программирования (см. 2.3.6), бинарным объектам (см. 2.3.7), функциям (см. 2.3.5), схемам (см. 2.3.9), последовательностям (см. 2.3.10), типам (см. 2.3.12), табличным пространствам (см. 2.3.13) и работа модуля `dblink` (см. 2.3.3). Также проверяется дискреционное управление доступом при использовании ролей (групп) (см. 2.3.8) и работа с внешними таблицами (см. 2.3.4).

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

6. ПРОВЕРКА МАНДАТНОГО УПРАВЛЕНИЯ ДОСТУПОМ

6.1. Механизм мандатного управления доступом к объектам ФС

Проверка механизма мандатного управления доступом к объектам ФС осуществляется в ходе выполнения автоматической процедуры тестирования подсистемы безопасности PARSEC (см. 3.1).

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

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

```
---[fmac]: start test
PARSEC FMAC TEST: INFO: начинаем...
progname = /usr/lib/parsec/tests/fmac
PARSEC FMAC TEST: INFO: Начинаем тест: PARSEC FMAC 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: PARSEC FMAC 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: PARSEC FMAC 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
```

6.2. Механизм мандатного управления доступом к объектам IPC

Проверка механизма мандатного управления доступом к объектам IPC осуществляется в ходе выполнения автоматической процедуры тестирования подсистемы безопасности PARSEC (см. 3.1).

Проверяется механизм мандатного управления доступом к объектам IPC, включая семафоры, разделяемую память, очереди сообщений.

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

```
---[ipc_mac]: start test
PARSEC IPC/SIGNAL TEST: INFO: start...
progname = /usr/lib/parsec/tests/ipc_mac
PARSEC IPC/SIGNAL TEST: INFO: Начинаем тест: PARSEC IPC/SIGNAL 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: PARSEC IPC/SIGNAL 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
```

6.3. Механизм мандатного управления доступом для сетевых взаимодействий

Проверка механизма мандатного управления доступом для сетевых взаимодействий осуществляется в ходе выполнения автоматической процедуры тестирования подсистемы безопасности PARSEC (см. 3.1).

Проверяется механизм мандатного управления доступом для сетевых взаимодействий, включая взаимодействия с использованием протокола UDP семейства TCP/IP 4 версии, протокола TCP семейства TCP/IP 4 версии, UNIX-сокетов.

Проверка 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 = /usr/lib/parsec/tests/tcpip_mac

PARSEC TCP/IP TEST: INFO: Начинаем тест: PARSEC TCP/IP TEST...

PARSEC TCP/IP TEST: INFO:  Итерация 0.
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1252
```

```
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:51738 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1252
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:51748 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1252
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:51750 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1252

PARSEC TCP/IP TEST: INFO:  Итерация 1.
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1253
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:56950 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1253

PARSEC TCP/IP TEST: INFO:  Итерация 2.
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1254
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:46762 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1254
PARSEC TCP/IP TEST: INFO: mac tcp socket test прошел успешно

PARSEC TCP/IP TEST: INFO: Начинаем тест: PARSEC TCP/IP 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: PARSEC TCP/IP TEST...

PARSEC TCP/IP TEST: INFO:  Итерация 0.
PARSEC TCP/IP TEST: INFO: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле unixfile
```

```
PARSEC TCP/IP TEST: INFO:  Итерация 1.
PARSEC TCP/IP TEST: INFO: ждем соединения на файле 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: ждем соединения на файле unixfile

PARSEC TCP/IP TEST: INFO:  Итерация 2.
PARSEC TCP/IP TEST: INFO: ждем соединения на файле 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: ждем соединения на файле unixfile
PARSEC TCP/IP TEST: INFO: mac unix stream socket test прошел успешно

PARSEC TCP/IP TEST: INFO: Начинаем тест: PARSEC TCP/IP TEST...

PARSEC TCP/IP TEST: INFO:  Итерация 0.
PARSEC TCP/IP TEST: INFO: ожидание данных из файла unixfile

PARSEC TCP/IP TEST: INFO:  Итерация 1.
PARSEC TCP/IP TEST: INFO: ожидание данных из файла unixfile

PARSEC TCP/IP TEST: INFO:  Итерация 2.
PARSEC TCP/IP TEST: INFO: ожидание данных из файла unixfile
PARSEC TCP/IP TEST: INFO: mac unix dgram socket test прошел успешно

PARSEC TCP/IP TEST: INFO: Начинаем тест: PARSEC TCP/IP 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: PARSEC TCP/IP 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 1255
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:43910 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1255
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:43912 ...
```

```
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1255
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:43914 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1255
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:43916 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1255
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:43918 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1255
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:43920 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1255
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:43922 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1255
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:43924 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1255
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:43926 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1255
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:43928 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1255
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:43930 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1255
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:43932 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!

PARSEC TCP/IP TEST: INFO: Начинаем тест для UNIX STREAM...
PARSEC TCP/IP TEST: INFO: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле unixfile

PARSEC TCP/IP TEST: INFO:  Итерация 1.

PARSEC TCP/IP TEST: INFO: Начинаем тест для TCP...
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1256
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:55890 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1256
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:55892 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1256
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:55894 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1256
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:55896 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1256
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:55898 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1256
```

```
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:55900 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1256
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:55902 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1256
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:55904 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1256
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:55906 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1256
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:55908 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1256
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:55910 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1256
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:55912 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!

PARSEC TCP/IP TEST: INFO: Начинаем тест для UNIX STREAM...
PARSEC TCP/IP TEST: INFO: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле unixfile

PARSEC TCP/IP TEST: INFO:  Итерация 2.

PARSEC TCP/IP TEST: INFO: Начинаем тест для TCP...
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1257
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:35632 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1257
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:35634 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1257
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:35636 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1257
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:35638 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1257
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:35640 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1257
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:35642 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1257
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:35644 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1257
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:35646 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1257
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:35648 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
```

```
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1257
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:35650 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1257
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:35652 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!
PARSEC TCP/IP TEST: INFO: ожидаю соединения на порте 1257
PARSEC TCP/IP TEST: INFO: соединение установлено 127.0.0.1:35654 ...
PARSEC TCP/IP TEST: INFO: ок! клиент получил верную строку от сервера!

PARSEC TCP/IP TEST: INFO: Начинаем тест для UNIX STREAM...
PARSEC TCP/IP TEST: INFO: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле 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: ждем соединения на файле unixfile
PARSEC TCP/IP TEST: INFO: mac privilage socket accept test прошел успешно

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

6.4. Механизм мандатного управления доступом к объектам БД

Проверка механизма мандатного управления доступом к объектам БД осуществляется в ходе выполнения автоматической процедуры тестирования СУБД (см. 3.2).

При тестировании проверяется:

1) мандатное управление доступом при выполнении операций работы с данными:
   а) выборки (см. 2.3.35, 2.3.19, 2.3.20, 2.3.21, 2.3.45);
   б) добавления (см. 2.3.30);
   в) модификации (см. 2.3.39);
   г) удаления (см. 2.3.26);
2) мандатное управление доступом к объектам БД:
   а) таблицам и представлениям (см. 2.3.37);
   б) бинарным объектам (см. 2.3.32);
   в) последовательностям (см. 2.3.36);
3) создание (автоматическая маркировка) и модификация объектов БД (см. 2.3.22, 2.3.23, 2.3.41, 2.3.15);
4) мандатное управление доступом в хранимых процедурах и триггерах на языках PL/Perl, Untrusted PL/Perl, PL/pgSQL, Untrusted PL/Python, PL/Tcl и Untrusted PL/Tcl (см. 2.3.34, 2.3.38);
5) сохранение меток безопасности при выполнении служебных функций СУБД (см. 2.3.42, 2.3.28, 2.3.52, 2.3.54, 2.3.44, 2.3.55);
6) конфигурационные параметры и типы данных (см. 2.3.43, 2.3.53, 2.3.46);
7) работа модуля `dblink` в условиях мандатного управления доступом (см. 2.3.24);
8) вывод мандатных атрибутов объектов БД метакомандой `\d` утилиты `psql` (см. 2.3.49).
9) управление мандатными атрибутами объектов БД (см. 2.3.16);
10) команда санкционированного изменения меток безопасности данных `CHMAC` и механизмы правил и триггеров для этой операции (см. 2.3.18);

11) работа ограничений целостности мандатных атрибутов `CHMAC` для внешних ключей (см. 2.3.17);

12) ограничения ДП-модели (см. 2.3.25);

13) работа с внешними таблицами в условиях мандатного управления доступом (см. 2.3.27);

14) наследование мандатных атрибутов таблиц при наследовании (см.2.3.29);

15) работа модуля аудита в различных конфигурациях (см. 2.3.40);

16) разграничение доступа при администрировании ролей (см.2.3.51).

В ходе выполнения ряда тестов осуществляется проверка автоматической маркировки объектов БД и защищаемых записей объектов БД, отражающих уровень их конфиденциальности.

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

7. ПРОВЕРКА ОЧИСТКИ ПАМЯТИ И ИЗОЛЯЦИИ ПРОЦЕССОВ

7.1. Механизмы работы с ОП

Проверка механизма работы с ОП осуществляется в ходе выполнения автоматической процедуры тестирования подсистемы безопасности PARSEC (см. 3.1).

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

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

```
---[mem_test]: start test
Тест очищения памяти...Граница текущего сегмента данных: 0000000000623000
Граница нового сегмента данных: 0000000000627000
Сигнатура 'Hello world!' @ 0000000000626ff3
Граница сегмента после 2го выделения памяти: 0000000000627000
Сигнатура '' @ 0000000000626ff3
УСПЕШНО
Тестирование механизма COW (копирование при записи)...Сигнатура 'Hello
from world #0' @ 0000000000601aa0, A
Сигнатура 'Hello from world #1' @ 0000000000601aa0, B
# Тест изоляции памяти
XXX = 8
Сигнатура 'Hello from world #0' @ 0000000000601aa0, A (после завершения
процесса B)
УСПЕШНО
# Тест изоляции памяти
XXX = 8
Test PASS
---[mem_test]: stop test
```

7.2. Механизм очистки памяти внешних носителей

Проверка механизма очистки памяти внешних носителей осуществляется в ходе выполнения автоматической процедуры тестирования подсистемы безопасности PARSEC (см. 3.1).

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 считается выполненной, если файл отчета `tests.log` содержит следующие строки:

```
---[secdelrm.sh]: start test
# Проверка гарантированного удаления на ФС ext2 и в режиме secdel
Создание образа диска...УСПЕШНО
Создание файла в файловой системе...УСПЕШНО
Проверка содержимого файла...УСПЕШНО
SECDEL удаление файла и поиск содержимого на диске...УСПЕШНО

# Проверка гарантированного удаления на ФС ext3 и в режиме secdel
Создание образа диска...УСПЕШНО
Создание файла в файловой системе...УСПЕШНО
Проверка содержимого файла...УСПЕШНО
SECDEL удаление файла и поиск содержимого на диске...УСПЕШНО

# Проверка гарантированного удаления на ФС ext4 и в режиме secdel
Создание образа диска...УСПЕШНО
Создание файла в файловой системе...УСПЕШНО
Проверка содержимого файла...УСПЕШНО
SECDEL удаление файла и поиск содержимого на диске...УСПЕШНО

# Проверка гарантированного удаления на ФС ext2 и в режиме secdelrnd
Создание образа диска...УСПЕШНО
Создание файла в файловой системе...УСПЕШНО
Проверка содержимого файла...УСПЕШНО
SECDEL удаление файла и поиск содержимого на диске...УСПЕШНО

# Проверка гарантированного удаления на ФС ext3 и в режиме secdelrnd
Создание образа диска...УСПЕШНО
Создание файла в файловой системе...УСПЕШНО
Проверка содержимого файла...УСПЕШНО
SECDEL удаление файла и поиск содержимого на диске...УСПЕШНО

# Проверка гарантированного удаления на ФС ext4 и в режиме secdelrnd
Создание образа диска...УСПЕШНО
Создание файла в файловой системе...УСПЕШНО
Проверка содержимого файла...УСПЕШНО
SECDEL удаление файла и поиск содержимого на диске...УСПЕШНО

Test PASS
---[secdelrm.sh]: stop test
```

8. ПРОВЕРКА МАРКИРОВКИ ДОКУМЕНТОВ

Для проверки маркировки документов необходимо:

1) настроить защищенный комплекс программ печати и маркировки документов, а также принтер для печати документов с ненулевым мандатным контекстом в соответствии с документами 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»;

2) выполнить графический вход в систему как пользователь `test` с уровнем 1;

3) создать текстовый документ в приложении LibreOffice Writer и отправить его на печать;

4) завершить сеанс пользователя;

5) выполнить графический вход в систему от имени учетной записи пользователя, входящего в группы `lpmac` и `lp`;

6) выполнить маркировку документа с использованием инструмента командной строки `makrjob`;

7) отправить сформированные задания на печать с использованием приложения `fly-admin-printer`;

Результат тестирования считается положительным если, после печати сформированных в результате маркировки заданий, на печать выведены страницы с соответствующими атрибутами.

9. ПРОВЕРКА КОНТРОЛЯ ПОДКЛЮЧЕНИЯ СЪЕМНЫХ МАШИННЫХ НОСИТЕЛЕЙ ИНФОРМАЦИИ И СОПОСТАВЛЕНИЯ ПОЛЬЗОВАТЕЛЯ С УСТРОЙСТВОМ

Для проверки контроля подключения съемных машинных носителей информации и сопоставления пользователя с устройством необходимо:

1) войти в систему от имени администратора;
2) запустить окно терминала;
3) подключить съемный USB-носитель;
4) выполнить команду:

```
sudo 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:

```
sudo mcedit /etc/fstab
```

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

```
/dev/sdc /mnt auto rw,user,noauto 0 0
```

7) создать ФС на USB-носителе командой:

```
sudo 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 метку безопасности и владельца командами:

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

10) размонтировать ФС на USB-носителе командой:

```
umount /mnt
```

11) войти в систему от имени обычного пользователя;
12) запустить окно терминала;
13) смонтировать USB-носитель командой:

```
mount /mnt
```

14) убедиться в выполнении правил мандатного и дискреционного управления доступом 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 меток безопасности, точке монтирования носителя и всем вложенным объектам ФС присваивается метка безопасности с минимальным уровнем и пустым списком категорий `0:0`. Владельцем назначается пользователь, смонтировавший ФС.

10. ПРОВЕРКА РЕГИСТРАЦИИ СОБЫТИЙ БЕЗОПАСНОСТИ

10.1. Система регистрации событий безопасности

Проверка системы регистрации событий безопасности осуществляется в ходе выполнения автоматической процедуры тестирования подсистемы безопасности PARSEC (см. 3.1).

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

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

```
---[audit_file.sh]: start test
Запуск системы протоколированияУСПЕШНО
# Проверка системы протоколирования
Установка параметров флагов аудита для каталога /tmp/tmp/file-757...УСПЕШНО
Установка флагов аудита для файла /tmp/file-757...УСПЕШНО
Создание события аудита open /tmp/file-757...УСПЕШНО
Создание события аудита chmod /tmp/file-757...УСПЕШНО
Создание события аудита chown /tmp/file-757...УСПЕШНО
Создание события аудита setfaud /tmp/file-757...УСПЕШНО
Создание события аудита setfacl /tmp/file-757...УСПЕШНО
Создание события аудита parsec_chmac /tmp/file-757...УСПЕШНО
Создание события аудита exec /tmp/file-757...УСПЕШНО
Создание события аудита unlink /tmp/file-757...УСПЕШНО
Остановка службы протоколирования...УСПЕШНО
Удаление флагов аудита с каталога /tmp...УСПЕШНО
Поиск событий open в журнале...
УСПЕШНО
Поиск событий exec в журнале...
УСПЕШНО
Поиск событий unlink в журнале...
УСПЕШНО
Поиск событий chmod в журнале...
УСПЕШНО
Поиск событий chown в журнале...
УСПЕШНО
Поиск событий setfacl в журнале...
УСПЕШНО
Поиск событий audit в журнале...
УСПЕШНО
Поиск событий mac в журнале...
```

```
УСПЕШНО
Поиск событий create в журнале...
УСПЕШНО
Запуск системы протоколированияУСПЕШНО
Test PASS
---[audit_file.sh]: stop test

---[audit_proc.sh]: start test
подготовка к тестам
добавление пользователя и выставление флагов аудита
тест аудита для пользователя : audittestuser
audittestuser ocxudnarmphew:ocxudnarmphew
завершено
подтест - uid gid
найдено событие uid...УСПЕШНО
найдено событие gid...УСПЕШНО
подтест - module
найдено событие init_module...УСПЕШНО
найдено событие delete_module...УСПЕШНО
подтест - создание файла
найдено событие - create...УСПЕШНО
подтест - mac
найдено событие mac...УСПЕШНО
подтест - mac
найдено событие mac...УСПЕШНО
подтест - открытие файла
найдено событие open...УСПЕШНО
подтест - запуск приложений
найдено событие exec...УСПЕШНО
подтест - удаление файла
найдено событие delete...УСПЕШНО
подтест - chmod
найдено событие chmod...УСПЕШНО
подтест - chown
найдено событие chown...УСПЕШНО
подтест - net
найдено событие net...УСПЕШНО
подтест - chroot
найдено событие chroot...УСПЕШНО
подтест - rename
найдено событие rename...УСПЕШНО
подтест - capabilities
найдено событие cap...УСПЕШНО
подтест - audit
найдено событие ch_audit...УСПЕШНО
```

```
подтест - acl
найдено событие acl...УСПЕШНО
подтест - mount
найдено событие mountУСПЕШНО
Check value
0
Тест завершен. Аудит процессов работает корректно
Test PASS
---[audit_proc.sh]: stop test
```

10.2. Регистрация событий при работе с БД

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

При выполнении тестирования (см. 3.2) 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 зарегистррованных событий СУБД необходимо:

1) войти в систему от имени администратора;
2) запустить окно терминала;
3) выполнить команду:

```
sudo cat /parsec/log/astra/events | grep postgres
```

Пример

Информация о событии СУБД:

```
{"PROGRAM":"postgres","PRIORITY":"info","PID":"38589","MSG":
     {"astra-postgres":{"unique_id":"1711614746.943:875","type_ru":
     "Идентификация и аутентификация пользователей в СУБД","type_en":
     "DBMS user identification and authentication","superuser":"no",
     "session_user_ID":"0","session_user":"UNDEFINED","result":"SUCCESS",
```

"name_ru":"Успешная аутентификация пользователя в СУБД","name_en":
"Successful user authentication in the DBMS","message_id":
"postgres_conn_success","message":"maclabel{0,0}","database":
"UNDEFINED","current_user_ID":"0","current_user":"UNDEFINED",
"cluster_name":"main","client_host":"[autovacuum]"}},"ISODATE":
"2024-03-28T11:32:26+03:00","HOST":"astra-80422","FACILITY":"local0"}

Из записи можно получить следующую информацию:

- тип события ("type_ru":);
- успешность осуществления события ("result":);
- хост, с которого отправлен клиентский запрос ("client_host":);
- имя кластера, к которому выполнено подключение ("cluster_name":);
- имя БД, к которой выполнено подключение ("database":);
- идентификатор сессии пользователя ("session_user_ID":);
- идентификатор текущего пользователя ("current_user_ID":).

11. ПРОВЕРКА НАДЕЖНОГО ФУНКЦИОНИРОВАНИЯ

11.1. 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.

11.2. Механизм надежного восстановления БД

Проверка механизма надежного восстановления БД осуществляется в ходе выполнения автоматической процедуры тестирования СУБД (см. 2.3.55).

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

12. ПРОВЕРКА РАБОТЫ МЕХАНИЗМА КОНТРОЛЯ ЦЕЛОСТНОСТИ

Контроль целостности объектов ФС осуществляется с использованием программы `afick`.

Для проверки работы механизма, осуществляющего контроль целостности объектов ФС, необходимо:

1) войти в систему от имени администратора с высокой меткой целостности;
2) запустить окно терминала (можно использовать комбинацию клавиш **<Alt+T>**);
3) выполнить команду:

```
sudo afick -i
```

4) дождаться построения первоначальной БД;
5) сделать резервную копию изменяемого файла, например, файлов `/bin/blkid` и `/sbin/sysctl`:

```
sudo cp /sbin/blkid /sbin/blkid.bak
sudo cp /sbin/sysctl /sbin/sysctl.bak
```

6) произвести намеренные изменения в ФС:

```
sudo -s
echo asdf >> /sbin/blkid
chmod 700 /sbin/sysctl
```

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

```
sudo afick -k
```

Результат тестирования считается положительным, если в результате выполнения программы `afick` на экран монитора выведена информация об изменении файла `/sbin/blkid` и об изменении метки безопасности у файла `/sbin/sysctl`;

8) после завершения тестирования восстановить исходное состояние файлов, выполнив команды:

```
sudo cp /sbin/blkid.bak /sbin/blkid
sudo cp /sbin/sysctl.bak /sbin/sysctl
```

13. ДОПОЛНИТЕЛЬНЫЕ ПРОВЕРКИ СЗИ

13.1. Библиотечные функции libpdac++

Проверка библиотечных функций `libpdac++` осуществляется с помощью теста `PDAC++_test` в ходе выполнения автоматической процедуры тестирования подсистемы безопасности PARSEC (см. 3.1).

Проверяется корректность работы библиотечных функций `libpdac++`: инициализация библиотеки, генерация udev-правил для тестового пользователя, удаление udev-правил для пользователя.

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

```
Running suite(s): PDAC++
100%: Checks: 3, Failures: 0, Errors: 0
/build/parsec-2.5.269+g10e25e0/tests/unit/pdac.cpp:7:P:PDAC++-common:
       test_PDACpp_common:0: Passed
/build/parsec-2.5.269+g10e25e0/tests/unit/pdac.cpp:14:P:PDAC++-udev:
       test_PDACpp_generate_rules:0: Passed
/build/parsec-2.5.269+g10e25e0/tests/unit/pdac.cpp:37:P:PDAC++-udev:
       test_PDACpp_rm_rules:0: Passed
100%: Checks: 3, Failures: 0, Errors: 0
```

13.2. Библиотечные функции libparsec-aud

Проверка библиотечных функций `libparsec-aud` осуществляется с помощью теста `aud_test` в ходе выполнения автоматической процедуры тестирования подсистемы безопасности PARSEC (см. 3.1).

Проверяется корректность работы библиотечных функций `libparsec-aud`: инициализация библиотеки, преобразование текстовых меток правил аудита во внутренний формат и обратно, установка правил аудита для текущего процесса, получение метки правил аудита для текущего процесса, успешность установки метки правил аудита на текущий процесс.

Выполняется три цикла теста. Проверка считается выполненной, если файл отчета `tests.log` содержит следующие строки:

```
Running suite(s): aud
100%: Checks: 3, Failures: 0, Errors: 0
```

13.3. Разрешения на изменение меток безопасности

Проверка разрешений на изменение меток безопасности осуществляется с помощью теста `chlbl` в ходе выполнения автоматической процедуры тестирования подсистемы безопасности PARSEC (см. 3.1).

Проверяется корректность реализации разрешений на изменение меток безопасности объектов с помощью функций `pdp_set_path`, `pdp_set_lpath`, `pdp_set_fd`. Тест проводится для всех возможных комбинаций параметров:

1) объект-файл или сетевой сокет;
2) вызов от имени `root` или обычного пользователя;
3) наличие или отсутствие привилегии `PARSEC_CAP_CHMAC`;
4) наличие или отсутствие привилегии `PARSEC_CAP_SOCK`.

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

```
PARSEC CHLBL TEST: INFO: file (no) SUCCESS
PARSEC CHLBL TEST: INFO: sock (no) SUCCESS
PARSEC CHLBL TEST: INFO: file root (yes) SUCCESS
PARSEC CHLBL TEST: INFO: sock root (no) SUCCESS
PARSEC CHLBL TEST: INFO: file CAP_CHMAC (yes) SUCCESS
PARSEC CHLBL TEST: INFO: sock CAP_CHMAC (no) SUCCESS
PARSEC CHLBL TEST: INFO: file root CAP_CHMAC (yes) SUCCESS
PARSEC CHLBL TEST: INFO: sock root CAP_CHMAC (no) SUCCESS
PARSEC CHLBL TEST: INFO: file CAP_SOCK (no) SUCCESS
PARSEC CHLBL TEST: INFO: sock CAP_SOCK (no) SUCCESS
PARSEC CHLBL TEST: INFO: file root CAP_SOCK (yes) SUCCESS
PARSEC CHLBL TEST: INFO: sock root CAP_SOCK (yes) SUCCESS
PARSEC CHLBL TEST: INFO: file CAP_CHMAC CAP_SOCK (yes) SUCCESS
PARSEC CHLBL TEST: INFO: sock CAP_CHMAC CAP_SOCK (yes) SUCCESS
PARSEC CHLBL TEST: INFO: file root CAP_CHMAC CAP_SOCK (yes) SUCCESS
PARSEC CHLBL TEST: INFO: sock root CAP_CHMAC CAP_SOCK (yes) SUCCESS
PARSEC CHLBL TEST: INFO: result: SUCCESS
```

```
python test passed for /usr/bin/python
python test passed for /usr/bin/python binary
python test passed for /usr/bin/python2
python test passed for /usr/bin/python2 binary
python test passed for /usr/bin/python2.7
python test passed for /usr/bin/python2.7 binary
python test passed for /usr/bin/python3
```

```
python test passed for /usr/bin/python3.5
python test passed for /usr/bin/python3.5m
python test passed for /usr/bin/python3m
perl test passed for /usr/bin/perl
perl test passed for /usr/bin/perl5.24.1
Interpreter lock test:PASS
```

13.4. Подсистема мандатного контроля целостности

Проверка подсистемы мандатного контроля целостности осуществляется в ходе выполнения автоматической процедуры тестирования подсистемы безопасности PARSEC (см. 3.1).

Также проверка может быть выполнена путем запуска исполняемого файла `mictest` и скрипта `mictest.sh`. Исполняемый файл запускает проверку, скрипт выполняет подготовку окружения и агрегацию результатов проверки.

Проверяется корректность функционирования подсистемы мандатного контроля целостности (МКЦ). Выполняется проверка следующих функций:

1) процесс с уровнем МКЦ, не включающим в себя по маске уровень МКЦ файла на ФС, не может произвести в него запись, но может читать;

2) работа привилегий, позволяющих обходить МКЦ, и возможность их несанкционированной установки для низкоцелостных процессов;

3) процесс с уровнем МКЦ, не включающим в себя по маске уровень МКЦ другого процесса, не может посылать ему сигналы;

4) процесс с уровнем МКЦ не может назначить себе через API parsec уровень МКЦ, не включающий в себя по маске уже имеющийся уровень МКЦ;

5) процесс с уровнем МКЦ, не включающим в себя по маске уровень МКЦ другого `systemd` юнита, не может им управлять средствами `systemd`.

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

```
PARSEC MIC TEST: INFO: test files: SUCCESS
PARSEC MIC TEST: INFO: test files (IGNMACINT set) : SUCCESS
PARSEC MIC TEST: INFO: test signals (IGNMACINT set) : SUCCESS
PARSEC MIC TEST: INFO: test signals: SUCCESS
PARSEC MIC TEST: INFO: test ilev change: SUCCESS
PARSEC MIC TEST: INFO: test miccap: SUCCESS
PARSEC MIC TEST: INFO: test systemd level 63: SUCCESS
PARSEC MIC TEST: INFO: test systemd level 31: SUCCESS
PARSEC MIC TEST: INFO: test systemd level 15: SUCCESS
```

```
PARSEC MIC TEST: INFO: test systemd level 7: SUCCESS
PARSEC MIC TEST: INFO: test systemd level 3: SUCCESS
PARSEC MIC TEST: INFO: test systemd level 1: SUCCESS
PARSEC MIC TEST: INFO: test systemd level 0: SUCCESS
```

13.5. Библиотечные функции libparsec-aux

Проверка библиотечных функций `libparsec-aux` осуществляется с помощью теста `pdp_aux` в ходе выполнения автоматической процедуры тестирования подсистемы безопасности PARSEC (см. 3.1).

Проверяется корректность работы библиотечных функций `libparsec-aux`: создание сессионной каталога пользователя с установкой метки безопасности.

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

```
Running suite(s): pdp_aux
/tmp/user/l0i0c0x0t0x0
100%: Checks: 2, Failures: 0, Errors: 0
/build/parsec-2.5.269+g10e25e0/tests/unit/pdp_aux.c:28:P:pdp_make_session_dir:
       test_pdplugm_make_session_dir:0: Passed
/build/parsec-2.5.269+g10e25e0/tests/unit/pdp_aux.c:36:P:pdp_make_session_dir:
       test_pdpml_session_dir:0: Passed
100%: Checks: 2, Failures: 0, Errors: 0
```

13.6. Работа утилиты rsync

Проверка работы утилиты `rsync` с файлами осуществляется с помощью скрипта `rsync.sh` в ходе выполнения автоматической процедуры тестирования подсистемы безопасности PARSEC (см. 3.1).

Проверяется работа утилиты `rsync` с файлами, для которых в расширенных атрибутах (`xattrs`) указаны метки мандатного управления доступом.

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

```
sending incremental file list
./
level/
level/file
```

```
sent 389 bytes  received 48 bytes  874.00 bytes/sec
total size is 346  speedup is 0.79
sending incremental file list
parsec_testdir/
parsec_testdir/level/
parsec_testdir/level/file

sent 420 bytes  received 50 bytes  940.00 bytes/sec
total size is 519  speedup is 1.10
```

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

БД — база данных
КСЗ — комплекс средств защиты
НСД — несанкционированный доступ
ОП — оперативная память
ОС — операционная система специального назначения «Astra Linux Special Edition»
ПРД — правила разграничения доступа
СВТ — средства вычислительной техники
СУБД — система управления базами данных
ФС — файловая система

CCR — Container Clearance Required (атрибут способа доступа к содержимому контейнера в рамках мандатного управления доступом)
DAC — Discretionary Access Control (дискреционное управление доступом)
IP — Internet Protocol (межсетевой протокол)
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 управления передачей данных)
UDP — User Datagram Protocol (протокол пользовательских дейтаграмм)

Лист регистрации изменений

Изм.	Номера листов (страниц) изменен-ных	заменен-ных	новых	аннулиро-ванных	Всего листов (страниц) в документе	Номер документа	Входящий номер сопро-водительного документа и дата	Подпись	Дата
1		Все				РУСБ.14.03-11 от 20.10.11			
2		Все				РУСБ.14.04-13 от 21.03.13			
3		Все				РУСБ.14.05-14 от 19.11.14			
4		Все				РУСБ.14.02-16 от 17.02.16			
5		Все				РУСБ.14.02-18 от 11.04.18			
6		Все				РУСБ.14.03-18 от 19.04.18			
7		Все				РУСБ.08-20 от 18.02.20			
8		Все				РУСБ.06-21 от 01.04.21			
9		Все				РУСБ.10-22 от 03.11.22			
10		Все				РУСБ.14-23 от 26.12.23			
11		Все				РУСБ.02-24 от 25.03.24			