ОПИСАНИЕ КОНТРОЛЬНОГО ПРИМЕРА

Оглавление

1. Общие сведения....3
2. Разграничение прав доступа в СУБД....5
3. Состав контрольного примера....6
4. Варианты использования контрольного примера....9
4.1. Стенд на базе виртуальной машины....9
4.1.1. Параметры конфигурации ЕПП....9
4.1.2. Конфигурации контрольного примера....10
4.2. Самостоятельное развертывание контрольного примера....11
4.2.1. Доменная структура на базе ALD....12
4.2.2. Доменная структура на базе FreeIPA....23
5. Запуск контрольного примера....37

1. Общие сведения

Контрольный пример разработан в целях демонстрации общих принципов построения и функционирования прикладного программного обеспечения (ПО) в трехзвенной клиент-серверной архитектуре информационной системы (ИС) в среде «Astra Linux Special Edition» (далее ОС СН).

В рассматриваемом примере трехзвенная клиент-серверная архитектура содержит программные компоненты, функционирующие под управлением ОС СН:

– клиентскую часть на основе веб-браузера Firefox в качестве компонента представления данных;

– веб-сервер (сервер приложений) под управлением веб-сервера Apache2 в качестве прикладного компонента, реализующего бизнес-логику ИС;

– сервер базы данных под управлением СУБД PostgreSQL в качестве компонента управления ресурсами.

Все базовые технологии контрольного примера (веб-браузер Firefox, веб-сервер Apache2, СУБД PostgreSQL 11) являются средствами защищенной обработки информации из состава ОС СН.

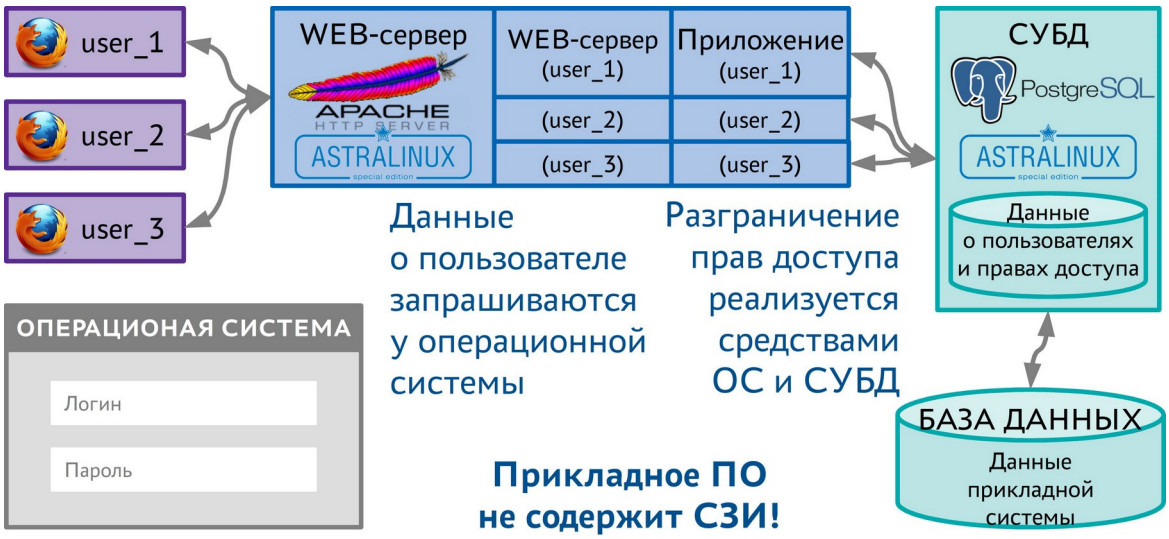

Рисунок 1.1 - Схема построения прикладного ПО в трехзвенной клиент-серверной архитектуре ИС с использованием интегрированных механизмов безопасности ОС СН

Взаимодействие компонентов ИС (рисунок 1.1) происходит с использованием встроенных в ОС СН механизмов разграничения доступа. Указанные компоненты

функционируют в составе многопользовательской информационной системы на базе локальной или глобальной вычислительной сети с единым пространством пользователей (ЕПП). ЕПП обеспечивает сквозную аутентификацию пользователей по сети и централизованное хранение информации об учетных записях пользователей и группах.

Каждому пользователю после входа в систему присваивается контекст безопасности включающий:

- Контекст дискреционного управления доступом (учетная запись, участие в группах);
- Контекст мандатного управления доступом (метка безопасности), состоящий из классификационной метки и метки целостности.

Далее все запущенные пользователем процессы наследуют этот контекст.

При выполнении запроса на обработку ресурсов базы данных (БД), происходит обращение к серверу приложений. Обращение осуществляется из веб-браузера, который наследует контекст безопасности запустившего его пользователя, и между браузером и веб-сервером устанавливается защищенное сетевое соединение.

При сетевом взаимодействии передача контекста безопасности между клиентской и серверной частью обуславливается:

– передачей учетных данных в процессе аутентификации клиента сервером по протоколу Kerberos. В процессе аутентификации клиента сервер получает от контроллера домена безопасности (ЕПП) информацию об параметрах учетной записи пользователя, то есть все параметры контекста работы пользователя, за исключением его мандатного контекста;

– передачей мандатного контекста с использованием сетевого стека ОС СН. ОС обеспечивает автоматическое добавление информации о классификационной метке, включающей иерархический уровень конфиденциальности и неиерархические категории конфиденциальности процесса (веб-браузера), инициировавшего соединение, в исходящие IP-пакеты согласно ГОСТ Р 58256-2018. На серверной стороне происходит извлечение из соединения классификационной метки. При этом метка целостности, также входящая в мандатный контекст, не используется.

Полученная классификационная метка объединяется с полученными в результате аутентификации параметрами учетной записи пользователя, образуя тем самым контекст безопасности, соответствующий контексту безопасности запущенного пользователем процесса (например, веб-браузера), инициировавшего соединение.

Получив контекст безопасности, веб-сервер создаёт в этом контексте дочерний процесс для дальнейшей обработки информации. Данный процесс обладает тем же набором прав, привилегий и мандатных атрибутов, что и учётная запись пользователя, запросившего доступ.

Прикладное ПО из состава сервера приложений формирует средствами языка манипулирования данными запрос к СУБД. Обращение прикладного ПО к СУБД инициирует установку защищённого сетевого соединения. На основании полученного из соединения мандатного контекста безопасности, а также полученных у контроллера домена в процессе аутентификации параметрах пользователя, в контексте которого запущен процесс, пытающийся подключиться к базе данных, СУБД осуществляет проверку прав доступа к запрашиваемым ресурсам. Таким образом, обработка базы данных в дальнейшем выполняется в соответствии с контекстом безопасности пользователя, инициировавшего первоначальный запрос.

Подробное описание взаимодействия компонентов трехзвенной клиент-серверной архитектуры ИС в среде ОС СН можно найти документе «Рекомендации по построению прикладного программного обеспечения для интеграции с механизмами идентификации, аутентификации, авторизации и разграничения доступа ОС СН» (https://nas01.astralinux.ru/sharing/n62RYHgK7).

2. Разграничение прав доступа в СУБД

Контрольный пример демонстрирует разграничение прав доступа в СУБД PostgreSQL. СУБД использует механизмы безопасности ОС СН для получения контекста безопасности пользователя ОС, запросившего доступ к данным БД. Проверка доступа осуществляется на основе контекста и разрешённых для этого контекста операций.

С каждым типом объектов БД ассоциируется определенный набор типов доступа (возможных операций). Для каждого объекта явно задается список разрешен-

ных для каждого из поименованных субъектов БД (пользователей, групп или ролей) типов доступа (ACL). И в дальнейшем при разборе запроса к БД осуществляется проверка возможности предоставления доступа субъекта к объекту типа, соответствующего запросу.

В дополнение к стандартной системе прав, управляемой командой GRANT, для ограничения набора данных, выдаваемых пользователю, можно применять входящую в PostgreSQL систему фильтрации строк (POLICY) под названием ROW LEVEL SECURITY (RLS). Такая политика защиты строк ограничивает для пользователей наборы строк, которые могут быть возвращены обычными запросами или добавлены, изменены и удалены командами, изменяющими данные. Подробное описание стандартной системы прав СУБД PostgreSQL и системы фильтрации строк можно найти по ссылке https://www.postgresql.org/docs/11/ddl-priv.html в разделах 5.6 и 5.7 соответственно.

Также в СУБД PostgreSQL реализован механизм мандатного управления доступом. Разграничение доступа к защищаемым ресурсам БД осуществляется на основе классификационных меток, содержащих иерархические уровни конфиденциальности и неиерархические категории конфиденциальности.. СУБД PostgreSQL извлекает классификационную метку пользователя из контекста входящего соединения и для назначения, хранения и модификации меток использует механизмы ОС СН. Проверка мандатных прав доступа к объектам БД осуществляется одновременно с проверкой дискреционных прав доступа к ним.

3. Состав контрольного примера

Контрольный пример включает в свой состав следующие файлы:

– contrprimer.py - CGI-сценарий на языке python3 для обработки запросов к веб-серверу и взаимодействия с СУБД;

– contrprimer.html - веб-страница на языках html и javascript для предоставления графического интерфейса пользователя и взаимодействия с веб-сервером;

– contrprimer_0.sql, contrprimer_1.sql, contrprimer_2.sql, contrprimer_3.sql - сценарии на языке PL/pgSQL. Сценарии предлагают различные варианты демонстрации разграничения доступа к данным БД. Генерация базы данных каким-

либо из этих сценариев позволяет экспериментировать с различными комбинациями механизмов разграничения доступа.

Далее приведены описания предлагаемых вариантов разграничения доступа.

1) *contrprimer_0.sql*

Данный вариант демонстрирует только мандатное разграничение доступа к базе данных. Дискреционное разграничение доступа и политика защиты на уровне строк (записей) не применяется.

2) *contrprimer_1.sql*

В данном варианте помимо мандатного разграничения доступа используется политика защиты на уровне строк (RLS).

Этот вариант демонстрирует такую политику, при которой изменение прав доступа к записям осуществляется в зависимости от времени создания записи и текущего времени выполнения запросов.

При создании записи в поле *insert_date* записывается время создания записи.

При выполнении запроса на чтение, изменение или удаление записи происходит сравнение значения секунд времени создания записи со значением секунд в текущем времени выполнения запроса. Доступ к записи предоставляется только к записям, созданным в первой или второй половине минуты соответственно текущему времени запроса.

3) *contrprimer_2.sql*

В этот вариант помимо мандатного разграничения доступа и политики защиты на уровне записей (RLS) включено разграничение доступа к полям определенных столбцов таблицы с использованием стандартной системы прав, управляемой командой GRANT.

Пользователи, имеющие права роли «Администратор_БД», имеют доступ на чтение ко всем записям, а также могут создавать, удалять и классифицировать записи.

Пользователи, имеющие права роли «Пользователь», могут изменять поля столбцов *update_user, update_date* и имеют доступ на чтение только к записям в соответствии со следующей иерархической классификацией подразделений:

Организация

Департамент_1

Отдел_11

Отдел_12

Департамент_2

Отдел_21

Отдел_22

По умолчанию участником роли доступа «Администратор_БД» является роль входа user3, участниками роли доступа «Пользователь» являются user0, user1, user2. При этом user0 является участником роли «Отдел_12», user1 является участником роли «Департамент_1», user2 является участником роли «Организация».

Включая или исключая с помощью средств администрирования БД пользователей из ролей «Пользователь» или «Администратор_БД», а также включая или исключая их из ролей в иерархической классификации подразделений, можно управлять их доступом к данным.

4) *contrprimer_3.sql*

В данном варианте кроме мандатного разграничения доступа применяется следующая комбинация политики защиты на уровне записей (RLS) и дискреционного разграничения доступа (стандартной системы прав, управляемой командой GRANT).

При создании каждой записи с помощью триггера выполняется:

- создается роль с именем, равным уникальному идентификатору (*id*) записи;

- пользователю, создавшему запись, даются права этой созданной роли.

В результате каждый пользователь имеет доступ только к созданным им записям.

Пользователи, включенные в роль «Администратор_БД» (по умолчанию user3) имеют доступ ко всем записям.

Включая с помощью средств администрирования пользователей в те или иные роли можно изменять их права доступа к записям БД.

Файлы контрольного примера доступны для скачивания по ссылке:

https://nas01.astralinux.ru/sharing/uMupYU3h2.

4. Варианты использования контрольного примера

Файлы контрольного примера (contrprimer_0.sql, contrprimer_1.sql, contrprimer_2.sql, contrprimer_3.sql, contrprimer.html, contrprimer.py) должны быть расположены на компьютере, выполняющем роль веб-сервера.

В системе должна быть обеспечена сквозная аутентификация и авторизация пользователей в браузере и базе данных путем организации совместной работы соответствующих сервисов (веб-сервера Apache2, СУБД PostrgeSQL) с ЕПП.

Для изучения контрольного примера можно применить два подхода:

– использовать подготовленный стенд на базе виртуальной машины с уже настроенными сервисами (п. 4.1);

– развернуть стенд трехзвенной клиент-серверной архитектуры, выполнив настройки необходимых сервисов самостоятельно (п. 4.2).

4.1. Стенд на базе виртуальных машин

Виртуальная машина представлена в формате OVA и предназначена для использования средствами эмуляции аппаратного обеспечения на основе VirtualBOX или QEMU («Руководство администратора. Часть 1. РУСБ.10015-01 95 01-1», п. 9.1 «Средства виртуализации»).

Для преобразования формата OVA к формату qcow2 (формат qcow2 используется в QEMU) необходимо выполнить следующие команды:

tar -xvf original.ova

qemu-img convert -O qcow2 original.vmdk original.qcow2

Архив с файлами виртуальных машин с настроенным контрольным примером (Client (Smolensk-1.7).ova, Domain server (Smolensk-1.7).ova) доступен для скачивания по ссылке: https://nas01.astralinux.ru/sharing/Rf2Bd5pT6

Виртуальная машина представляет собой локальную трехзвенную информационную систему (ИС), компоненты которой расположены на одном хосте. В качестве

ЕПП используется служба ALD, а для совместной работы с ЕПП веб-сервера Apache2 и СУБД PostgreSQL в ОС СН виртуальной машины уже произведены необходимые настройки.

4.1.1. Параметры конфигурации ЕПП

Перед использованием контрольного примера необходимо выполнить проверку конфигураций сервисов, обеспечивающих работу ЕПП. Для этого предлагается осуществить вход на рабочий стол ОС СН локальным администратором. Аутентификационные данные локального администратора, а также сведения о заведенных в системе доменных пользователях представлены в таблице 4.1.

Таблица 4.1 - Сведения о заведенных в системе пользователях

№	Имя пользователя	Пароль	Описание	Диапазон мандатных уровней
1.	administrator	12345678	Локальный администратор	{0,3}
2.	admin/admin	12345678	Корневой администратор домена	
3.	user0	12345678	Доменный пользователь	{0,0}
4.	user1	12345678	Доменный пользователь	{0,1}
5.	user2	12345678	Доменный пользователь	{0,2}
6.	user3	12345678	Доменный пользователь	{0,3}

Сведения о конфигурации сети и параметры ALD представлены в таблице 4.2.

Таблица 4.2 - Параметры конфигурации ЕПП

№	Описание	Параметр
1.	Имя домена	.astra.ntc
2.	Полное имя серверного компьютера ALD (FQDN)	q.astra.ntc
3.	IP4.ADDRESS	192.168.111.11/24

4.1.2. Конфигурации контрольного примера

Файлы контрольного примера (*contrprimer_0.sql, contrprimer_1.sql, contrprimer_2.sql, contrprimer_3.sql, contrprimer.html, contrprimer.py)* располагаются в каталоге */var/www/html*.

В файле *contrprimer.py* строка подключения к базе данных содержит имя хоста, на котором располагается база данных:

DB_CONNECT = 'dbname=contrprimer host=p.astra.ntc port=5432'

В СУБД PostgreSQL созданы одноименные *доменным* пользователи, а также база данных *contrprimer*.

Демонстрация различных комбинаций механизмов разграничения доступа осуществляется с использованием одного из четырех вариантов, представленных в файлах сценариев *contrprimer_0.sql, contrprimer_1.sql, contrprimer_2.sql, contrprimer_3.sql*.

Для использования sql-сценария следует перейти в каталог */var/www/html:*

cd /var/www/html

и выполнить команду:

sudo psql -h localhost -U postgres -f contrprimer_0.sql

с указанием имени соответствующего сценария в качестве последнего параметра, выбранного в соответствии с одним из четырех вариантов демонстрации разграничения доступа. Описание вариантов генерации базы данных с использованием различных скриптов представлено в п. 3.

Следует отметить, что выполнение какого-либо скрипта генерации БД приведет к удалению предыдущей базы данных.

Запуск контрольного примера осуществляется по инструкции, приведенной в п. 5.

4.2. Самостоятельное развертывание контрольного примера

Ниже представлены сведения, необходимые для самостоятельного развертывания контрольного примера в трехзвенной клиент-серверной архитектуре с использованием веб-сервера Apache2 и СУБД PostrgeSQL из состава ОС СН (Astra Linux Special Edition v1.6.)

Для использования контрольного примера необходимо произвести:

– подготовку инфраструктуры к развертыванию системы;

– установку и настройку ЕПП;

– организацию работы пользователей в домене;

– установку и настройку комплекса программ гипертекстовой обработки данных для совместной работы с ЕПП;

– установку и настройку сервера управления базами данных PostgreSQL для совместной работы с ЕПП;

– создание и настройку БД для контрольного примера.

В качестве ЕПП в системе предлагается использовать либо службу ALD, либо службу FreeIPA. Ниже представлены сценарии развертывания контрольного примера для каждой из этих служб.

В ходе последующих действий по развертыванию контрольного примера потребуется установка пакета адаптации PostgreSQL для языка программирования Python3 (python3-psycopg2). Пакет размещен на установочном диске ОС СН а также доступен в базовом репозитории ОС СН.

4.2.1. Доменная структура на базе ALD

4.2.1.1. Подготовка инфраструктуры

В случае использования службы ALD компоненты трехзвенной информационной системы предлагается расположить на одном хосте. Параметры домена ALD представлены в таблице 4.3.

Таблица 4.3 - Параметры конфигурации домена ALD

№	Описание	Параметр
1.	Имя домена	.astra.ntc
2.	Короткое имя компьютера	q
3.	Полностью определенное доменное имя (FQDN)	q.astra.ntc
4.	IP4.ADDRESS	192.168.111.11/24

Необходимо выполнить настройку сетевого подключения - назначить доступному сетевому адаптеру статический IP-адрес.

Для просмотра доступных соединений необходимо выполнить команду:

sudo nmcli connection show

Для просмотра текущих настроек сетевого адаптера необходимо выполнить команду с указанием значения переменной "DBUS-PATH" (полученной по результатам выполнения предыдущей команды, например, *path 1*):

sudo nmcli connection show path 1

Для задания постоянного IP-адреса для проводного соединения выполнить команду:

sudo nmcli connection modify path 1 ipv4.method manual ipv4.addr "192.168.111.11/24"

После выполненных действий перезапустить сетевое соединение:

sudo nmcli con down path 1

sudo nmcli con up path 1

Для нормального функционирования ЕПП необходимо выполнение следующих условий:

1) разрешение имен контроллера домена должно быть настроено таким образом, чтобы имя системы разрешалось, в первую очередь, как полное имя (например, *q.astra.ntc*). В то же время утилита *hostnamectl* должна возвращать короткое имя компьютера, например, *q*.

Для задания имени хоста (определённого в */etc/hostname*) можно воспользоваться командой:

sudo hostnamectl set-hostname q

2) должно быть уставлено соответствие между IP-адресом и именем компьютера в файле */etc/hosts*. Содержание файла */etc/hosts* для сервера ALD необходимо привести к виду:

Пример

127.0.0.1 localhost

192.168.111.11 q.astra.ntc q

4.2.1.2. Установка и настройка службы ALD

Установка службы ALD может осуществляться как при начальной установке ОС путем выбора соответствующих пунктов в программе установки, так и в ручном режиме уже в работающей системе.

В случае установки сервера ALD в ручном режиме возможно получения следующей ошибки установки:

Пример

insserv: Service nfs-common has to be enabled to start

service nfs-kernel-server insserv: exiting now!

update-rc.d: error: insserv rejected the script header

В таком случае для успешной установки пакетов сервера ALD необходимо вручную включить необходимую службу:

sudo systemctl enable nfs-common

Установку службы ALD надо выполнять с использованием пакетов a*ld-server-common* (установка сервера домена ALD) и *smolensk-security-ald* (установка пакета администрирования БД ALD).

После установки пакетов на сервере ALD (*q.astra.ntc*) – контроллере домена - в конфигурационном файле */etc/ald/ald.conf* необходимо изменить значения следующих параметров:

Пример

DOMAIN= .astra.ntc

SERVER= q.astra.ntc

SERVER_ON=1

CLIENT_ON=1

После внесенных изменений в конфигурационном файле на сервере ALD для инициализации новой базы ALD необходимо выполнить команду:

sudo ald-init init

Данную команду необходимо выполнить только при первоначальной настройке службы. В случае изменения конфигурационного файла */etc/ald/ald.conf* на сервере и для того, чтобы изменения вступили в силу, выполняется команда *ald-init commit-config*.

На этапе выполнения команды задается главный пароль к базе данных Kerberos и пароль администратора Astra Linux Directory. Пароль должен содержать не менее восьми буквенно-цифровых знаков.

В случае успеха в конце инициализации появится сообщение:

Пример

Сервер ALD активен

Клиент ALD включен

Astra Linux Directory сервер успешно инициализирован.

Проверку активности сервера ALD можно выполнить с использованием команды:

sudo ald-init status

Сообщение «Сервер ALD активен. Клиент ALD включен» говорит о том, что сервер Astra Linux Directory сконфигурирован и инициализирован.

4.2.1.3. Управление учетными записями в домене ALD

В рассматриваемом примере предлагается создать учетные записи пользователей в соответствии с таблицей 4.4. Пароли задаются в соответствии с уставленной политикой безопасности в домене.

Таблица 4.4 — Учетные данные доменных пользователей

№	**Имя пользователя**	**Диапазон иерархических уровней конфиденциальности**
1.	user0	{0,0}
2.	user1	{0,1}
3.	user2	{0,2}
4.	user3	{0,3}

Создание доменных пользователей можно производить с помощью графической утилиты «Управление политикой безопасности» (*fly-admin-smc*), либо с командной строки с помощью команды *ald-admin user-add <имя пользователя>*. Например, для создания пользователя user0 команда будет выглядеть так:

ald-admin user-add user0

После создания четырех пользователей (*user0, user1, user2, user3*) необходимо явно задать им мандатные атрибуты в соответствии с указанными в таблице 4.4 значениями. Назначение мандатных атрибутов можно произвести командами:

ald-admin user-mac <имя пользователя> --min-lev-int=<минимально разрешенный иерархический уровень конфиденциальности>

для установки минимального уровня, и

ald-admin user-mac <имя пользователя> --max-lev-int=<максимально разрешенный иерархический уровень конфиденциальности>

для установки максимального разрешенного мандатного уровня.

Иерархические уровни конфиденциальности задаются десятичными числами. При задании неиерархических категорий конфиденциальности (опции —min-cat-hex и --max-cat-hex) используются шестнадцатеричные значения.

Для пользователя *user0* выполнять назначение нулевого иерархического уровня конфиденциальности не обязательно (минимально разрешенный уровень «0» назначен ему по умолчанию), а, например, для пользователя *user3* необходимо выполнить назначение максимального уровня (с учетом установленного по умолчанию минимального нулевого уровня):

ald-admin user-mac user3 --max-lev-int=3

После создания новой учетной записи список разрешенных для входа компьютеров пуст - пользователь не имеет права входа в систему. Необходимо явно указать компьютер, с которого ему будет разрешен вход. Сделать это можно с помощью графической утилиты «Управление политикой безопасности», перейдя по пути «Домен ALD → Пользователи → <Имя Пользователя> → Привилегии домена → Компьютеры» (рисунок 4.1), нажав на кнопку «+», и выбрав из появившегося списка необходимый компьютер. После выполненных действий необходимо сохранить изменения.

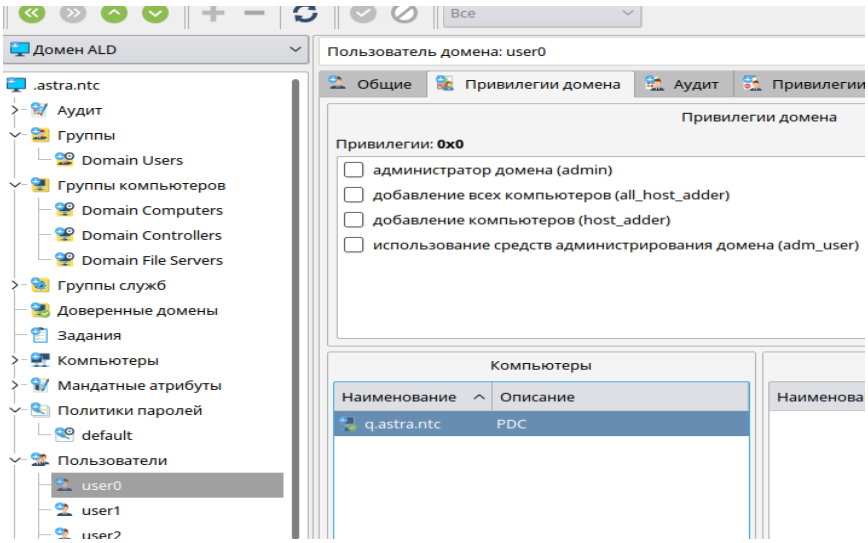

Рисунок 4.1 - Управление политикой безопасности. Привилегии домена пользователя домена: user0

Рекомендуется осуществить проверку входа в систему всех созданных пользователей под различными мандатными уровнями.

4.2.1.4. Установка веб-сервера Apache2 в домене ALD

Перед началом установки веб-сервера Apache2 рекомендуется дополнительно ознакомиться с разделом 10 - «Защищенный комплекс программ гипертекстовой обработки данных» документа «Руководство администратора. Часть 1. РУСБ.10015-01 95 01-1».

В случае, если в системе не предустановлены пакеты веб-сервера Apache2, необходимо произвести их установку, например, с использованием команды:

sudo apt-get install apache2

Установить пакет модуля веб-сервера Apache2 - *auth_kerb* из пакета *libapache2-mod-auth-kerb* для аутентификации через Kerberos (если не установлен):

sudo apt-get install libapache2-mod-auth-kerb

Необходимо активировать модули *headers* и *cgi*:

sudo a2enmod headers

sudo a2enmod cgi

В конфигурационном файле */etc/apache2/apache2.conf* добавить строку *AstraMode on*.

В каталоге */etc/apache2/sites-available* должен находиться файл с настройками виртуальных хостов. Необходимо привести этот конфигурационный файл (в данном случае *000-default.conf)* к следующему виду:

Пример

```
<VirtualHost *:80>
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html

        <Directory /var/www/html>
        AuthType Kerberos
        KrbAuthRealms ASTRA.NTC
        KrbServiceName HTTP/q.astra.ntc
```

Krb5Keytab /etc/apache2/keytab

KrbMethodNegotiate on

KrbMethodK5Passwd off

KrbSaveCredentials on

require valid-user

RequestHeader set MYMACLABEL "%m:%c"

AddHandler cgi-script .py

Options +ExecCGI

</Directory>

AddDefaultCharset UTF-8

SetEnv PYTHONIOENCODING utf8

ErrorLog ${APACHE_LOG_DIR}/error.log

CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>

Разрешить к исполнению файл виртуальных хостов можно командой (по умолчанию разрешен):

sudo a2ensite 000-default

Для корневого каталога виртуального хоста (по умолчанию */var/www/html*) и всех его родительских каталогов необходимо назначить мандатные атрибуты не меньше максимальных атрибутов объектов, к которым будет разграничиваться доступ:

sudo pdpl-file 3:0:0:ccnr /var/www/

sudo pdpl-file 3:0:0:ccnr /var/www/html/

После выполненных действий необходимо перезапустить веб-сервер Apache2, выполнив команду:

sudo systemctl restart apache2

Чтобы настроить автоматический запуск веб-сервера Apache2 при загрузке операционной системы, необходимо выполнить:

sudo systemctl enable apache2

Браузер Mozilla Firefox, в котором будут работать пользователи, необходимо настроить на поддержку аутентификации *negotiate*. Для этого необходимо для каждого доменного пользователя зайти под всеми доступными мандатными уровнями в рабочие сессии, в каждой из них открыть настройки браузера Mozilla Firefox, набрав в адресной строке *about:config*, согласиться с предупреждением о безопасности, найти через поисковую строку параметры *network.negotiate-auth.trusted-uris* и *network.negotiate-auth.delegation-uris* и указать в качестве их значений — *http://*.

4.2.1.5. Настройка Apache2 для совместной работы в домене ALD

Сквозная аутентификация и авторизация пользователей в браузере обеспечивается настройкой веб-сервера и браузера для работы в ЕПП. Перед началом настройки рекомендуется дополнительно ознакомиться инструкцией, представленной в документе «Руководство администратора. Часть 1. РУСБ.10015-01 95 01-1», п. 10.4 «Настройка для работы в ЕПП»

Для обеспечения сквозной авторизации необходимо создать на сервере ALD с помощью утилиты администрирования ALD принципала, соответствующего настраиваемому веб-серверу Apache2. Принципал создается с автоматически сгенерированным случайным ключом. Команда:

sudo ald-admin service-add HTTP/q.astra.ntc.

Ввести созданного «Принципала» в группу сервисов *mac*, используя следующую команду:

sudo ald-admin sgroup-svc-add HTTP/q.astra.ntc --sgroup=mac.

Создать файл ключа Kerberos для веб-сервера Apache2 с помощью утилиты администрирования ALD *ald-client*, используя следующую команду:

sudo ald-client update-svc-keytab HTTP/q.astra.ntc --ktfile="/etc/apache2/keytab".

Полученный файл должен быть доступен веб-серверу Apache2 по пути, указанному в конфигурационном параметре *Krb5Keytab* (*/etc/apache2/keytab*). Права доступа к этому файлу должны позволять читать его пользователю, от имени которого

работает веб-сервер Apache2. Для этого необходимо сменить владельца файла */etc/apache2/keytab* на пользователя *www-data*, выполнив следующую команду:

sudo chown www-data /etc/apache2/keytab.

Установить на файл */etc/apache2/keytab* права доступа для чтения остальным пользователям командой:

sudo chmod 644 /etc/apache2/keytab.

Перезапустить веб-сервер Apache2, выполнив команду:

sudo systemctl restart apache2

Для проверки правильности выполненных настроек следует зайти доменным пользователем в сессию под любым мандатным уровнем, открыть браузер Mozilla Firefox, набрать в адресной строке *q.astra.ntc*. В результате должна отобразиться страница *index.html* с надписью *«It works!»*.

4.2.1.6. Установка СУБД PostgreSQL в домене ALD

Установка СУБД PostgreSQL производится путем установки пакетов*:*

sudo apt install postgresql-astra

Для адаптации PostgreSQL для языка программирования Python3, необходимо установить пакет python3-psycopg2 (устанавливается с диска со средствами разработчика):

sudo apt install python3-psycopg2

Настройка сервера СУБД осуществляется установкой параметров в конфигурационном файле *postgresql.conf* В дополнение к файлу *postgresql.conf* в PostgreSQL используется еще два конфигурационных файла, которые контролируют аутентификацию клиента (*pg_hba.conf* - определяет конфигурационный файл для аутентификации по узлам, и *pg_ident.conf* - определяет конфигурационный файл для аутентификации по методу). По умолчанию все эти три файла находятся в каталоге данных кластера БД или в соответствующем кластеру конфигурационном каталоге, например, */etc/postgresql/11/main*. За расположение указанных файлов отвечают конфигурационные параметры.

Для обеспечения сквозной аутентификации пользователей ЕПП в СУБД необходимо в файле */etc/postgresql/11/main/pg_hba.conf* для внешних соединений в каче-

стве метода аутентификации указать gss и провести соответствующую настройку СУБД PostgreSQL для работы с ЕПП.

Файл конфигурации *pg_hba.conf* отредактировать следующим образом:

Пример

 # TYPE DATABASE USER CIDR-ADDRESS METHOD

local all postgres peer

host all postgres 127.0.0.1/32 trust

host all all 192.168.111.0/24 gss include_realm=0

где *192.168.111.0*/24 – ЛВС домена, 24 – маска подсети

В конфигурационном файле сервера СУБД PostgreSQL */etc/postgreslq/11/main/postgresql.conf* для приведенных ниже параметров задать соответствующие значения:

Пример

ac_ignore_socket_maclabel = false

listen listen_addresses = ''*

krb_server_keyfile = '/etc/postgresql-common/krb5.keytab'

4.2.1.7. Настройка СУБД PostgreSQL для совместной работы в домене ALD

Перед началом настройки СУБД для совместной работы с ALD рекомендуется дополнительно ознакомиться с п. 1.4.2 «Использование сквозной аутентификации в ЕПП» документа «Руководство администратора. Часть 2. РУСБ.10015-01 95 01-2»

Для обеспечения совместной работы сервера СУБД PostgreSQL с ALD необходимо:

1) создать в БД ALD с помощью утилиты администрирования ALD принципала, соответствующего устанавливаемому серверу PostgreSQL. Принципал создается с автоматически сгенерированным случайным ключом:

ald-admin service-add postgres/q.astra.ntc

2) ввести созданного принципала в группу сервисов *mac*, используя следующую команду:

ald-admin sgroup-svc-add postgres/q.astra.ntc --sgroup=mac

3) создать файл ключа Kerberos для сервера СУБД PostgreSQL с помощью утилиты администрирования ALD *ald-client*, используя следующую команду (пример приведен для кластера БД по умолчанию):

ald-client update-svc-keytab postgres/q.astra.ntc --ktfile="/etc/postgresql-common/krb5.keytab"

Полученный файл должен быть доступен серверу СУБД PostgreSQL по пути, указанному в конфигурационном параметре *krb_server_keyfile* (в данном случае – */etc/postgresql-common/krb5.keytab*). Права доступа к этому файлу должны позволять читать его пользователю, от имени которого работает сервер СУБД PostgreSQL (как правило, владельцем файла назначается пользователь *postgres*);

4) сменить владельца файла *krb5.keytab,* полученного на предыдущем шаге, на пользователя *posgtres*, выполнив следующую команду:

sudo chown postgres /etc/postgresql-common/krb5.keytab

По итогам выполненных действий перезагрузить сервис:

sudo systemctl restart postgresql

4.2.1.8. Контрольный пример

Файлы контрольного примера (contrprimer_0.sql, contrprimer_1.sql, contrprimer_2.sql, contrprimer_3.sql, contrprimer.html, contrprimer.py) необходимо расположить в каталоге */var/www/html*.

От имени пользователя *postgres* в СУБД PostgreSQL необходимо создать одноименных *доменным* пользователей, а также базу данных *contrprimer*:

Пример

sudo su postgres

psql

create user user0;

create user user1;

create user user2;

create user user3;

create database contrprimer;

\q

exit

Создание ролей для подключения, правил разграничения доступа в базе данных осуществляется выполнением sql-сценария контрольного примера. Для его использования следует перейти в каталог */var/www/html:*

cd /var/www/html

и выполнить команду:

sudo psql -h localhost -U postgres -f contrprimer_0.sql

с указанием имени сценария в качестве последнего параметра (contrprimer_0.sql, contrprimer_1.sql, contrprimer_2.sql, contrprimer_3.sql), выбранного в соответствии с одним из четырех вариантов демонстрации разграничения доступа. Описание вариантов генерации базы данных с использованием различных сценариев представлено в п. 3.

В файле *contrprimer.py* строка подключения к базе данных должна содержать корректное имя хоста, на котором располагается база данных (для данной инфраструктуры домена — q.astra.ntc):

DB_CONNECT = 'dbname=contrprimer host= q.astra.ntc port=5432'

Запуск контрольного примера осуществляется по инструкции, приведенной в п. 5.

4.2.2. Доменная структура на базе FreeIPA

4.2.2.1. Подготовка инфраструктуры

В случае использования в качестве ЕПП службы FreeIPA компоненты трехзвенной информационной системы необходимо разнести на два хоста следующим образом. Серверную часть службы FreeIPA (контроллер домена) предлагается расположить на хосте с IP-адресом 192.168.111.11, клиентскую часть — на хосте с IP-адресом 192.168.111.12. Клиентская часть FreeIPA настраивается на используемый FreeIPA домен. На клиентской части FreeIPA устанавливаются компоненты веб-сервера и сервера базы данных и настраивается их взаимодействие с FreeIPA.

Инфраструктура домена FreeIPA представлена в таблице 4.5.

Таблица 4.5 - Параметры конфигурации домена FreeIPA

№	Описание	Параметр	
1.	Имя домена	.astra.ntc	
		Контроллер домена	**Клиент**
2.	Короткое имя компьютера	q	p
3.	Полностью определенное доменное имя (FQDN)	q.astra.ntc	p.astra.ntc
4.	IP4.ADDRESS	192.168.111.11/24	192.168.111.12/24

4.2.2.2. Подготовка к установке серверной части FreeIPA

Перед установкой службы FreeIPA необходимо убедиться, что установка сервиса FreeIPA производится на чистой ОС, на которой ранее не были установлены другие сервисы ЕПП.

При подготовке к установке серверной части FreeIPA следует:

1) На компьютере, предназначенном на роль контроллера домена, необходимо выполнить настройку сетевого подключения - назначить доступному сетевому адаптеру статический IP-адрес в соответствии с таблицей 4.5.

2) Разрешение имен контроллера домена должно быть настроено таким образом, чтобы имя системы разрешалось, в первую очередь, как полное имя (например, *q.astra.ntc*). В */etc/hostname* должно содержаться FQDN (*q.astra.ntc*). Для добавления имени хоста можно выполнить команду:

sudo hostnamectl set-hostname q.astra.ntc

3) Файл */etc/hosts* не должен использоваться в качестве базы данных доменных имен других хостов. Так как запрос к этому файлу имеет приоритет перед обращением к DNS. В нем рекомендуется только запись "самого себя", <ip-адрес> + имя в формате FQDN + короткое имя. Таким образом, содержание файла */etc/hosts* должно быть следующего вида:

Пример

127.0.0.1 localhost

192.168.111.11 q.astra.ntc q

4.2.2.3. Установка серверной части FreeIPA

Программные компоненты FreeIPA входят в состав ОС и могут быть установлены из терминала Fly или из графического менеджера пакетов.

Установить необходимые пакеты серверной части из командной строки можно командой:

для установки инструмента командной строки:

sudo apt install astra-freeipa-server

для установки графического инструмента:

sudo apt install fly-admin-freeipa-server

В ходе установки будет выдано несколько предупреждений, с которыми нужно согласиться, нажав *"OK"*.

Для запуска службы FreeIPA необходимо произвести инициализацию контроллера домена FreeIPA (запуск службы FreeIPA) с помощью инструмента командной строки путём выполнения команды:

sudo astra-freeipa-server -d astra.ntc -n q -o

После выполнения команды будет определён адрес компьютера и будут выведены на экран все исходные данные. Для подтверждения данных ввести *y* и нажать <*Enter*>. После подтверждения появится запрос на установку пароля администратора домена. Указанный пароль будет использоваться для входа в веб-интерфейс FreeIPA и при работе с инструментом командной строки.

После успешного завершения инициализации на экран будут выведены сообщения о перезапуске системных служб, а также данные контроллера домена и ссылка на веб-интерфейс.

Пример

Обнаружен настроенный домен astra.ntc

WEB: https://q.astra.ntc

После завершения процедур запуска рекомендуется осуществить перезагрузку ОС. Для дальнейшего администрирования ОС рекомендуется использовать учетную запись *admin* и пароль, заданный при запуске FreeIPA.

Для входа в веб-интерфейс можно в браузере Mozilla Firefox перейти по ссылке, предоставленной по завершению инициализации контроллера домена (*https://q.astra.ntc*). Для входа в веб-интерфейс используется имя учетной записи admin и пароль, заданный при запуске FreeIPA. При первой попытке подключения на экране браузера, появится сообщение о том, что соединение не защищено. В такой ситуации следует нажать кнопку «Дополнительно», в открывшемся окне нажать кнопку «Добавить исключение», в открывшейся экранной форме нажать кнопку «Подтвердить исключение безопасности».

4.2.2.4. Управление учетными записями в домене FreeIPA

Для создания доменных пользователей необходимо зайти в систему под высоким уровнем целостности администратором *admin*, созданным в ходе инициализации сервера FreeIPA.

Создание доменных пользователей можно произвести с использованием графического веб-интерфеса, доступного по ссылке, указанной после инициализации сервера FreeIPA, или с командной строки с помощью команды *ipa user-add* (если указать параметр *«--password»*, то будет запрошен пароль; если указать *« --random»*, то пароль будет сгенерирован автоматически; в параметрах --first и --last указываются имя и фамилия пользователя):

ipa user-add user0 --first=11 --last=11 --password

Необходимо создать четыре доменных пользователя (*user0, user1, user2, user3*) и задать им мандатные атрибуты в соответствии с указанными в таблице 4.4 значениями (п.4.2.1.3 «Управление учетными записями в домене ALD»). Пароли задаются в соответствии с уставленной политикой безопасности в домене.

Установку мандатных атрибутов и уровня целостности доменным пользователям можно осуществить через веб-интерфейс FreeIPA, перейдя по вкладкам меню «Профиль → Пользователи→ Активные пользователи». Далее следует войти в свойства пользователя и отредактировать его параметры учетной записи в разделе «Мандатные атрибуты» для полей *Min MAC* и *Max MAC* (рисунок 4.2).

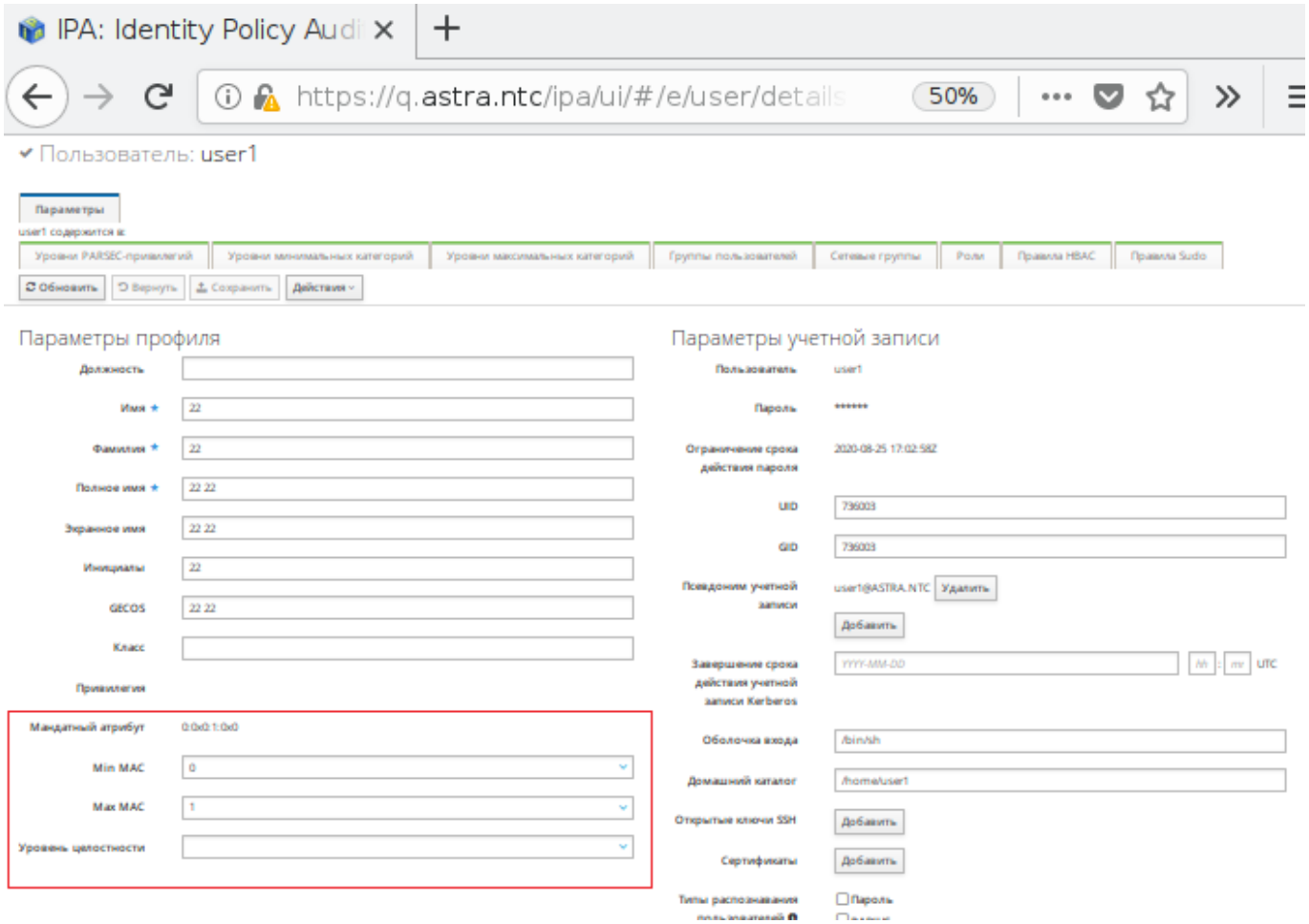

Рисунок 4.2 – Веб-интерфейс FreeIPA. Параметры учетной записи user1. Установка мандатных атрибутов

4.2.2.5. Подготовка к установке клиентской части FreeIPA

При подготовке к установке клиентской части FreeIPA следует:

1) На компьютере, предназначенном на роль клиента, необходимо выполнить настройку сетевого подключения - назначить доступному сетевому адаптеру статический IP-адрес в соответствии с таблицей 4.5.

2) Разрешение имен должно быть настроено таким образом, чтобы имя системы разрешалось, в первую очередь, как полное имя (например, *p.astra.ntc*). Файл */etc/hostname* должен содержать FQDN (*p.astra.ntc*). Для добавления имени хоста можно выполнить команду:

sudo hostnamectl set-hostname p.astra.ntc

3) Файл */etc/hosts* не должен использоваться в качестве базы данных доменных имен других хостов, так как запрос к этому файлу имеет приоритет перед обращени-

ем к DNS. В нем рекомендуется только запись "самого себя", <ip-адрес> + имя в формате FQDN + короткое имя.

Содержание файла */etc/hosts* должно быть следующего вида:

Пример

127.0.0.1 localhost

192.168.111.12 p.astra.ntc p

4) Клиентский компьютер и сервер FreeIPA должны видеть друг друга по сети. Для проверки использовать команду *ping*:

ping q.astra.ntc

4.2.2.6. Установка клиентской части FreeIPA

Программные компоненты клиентской части FreeIPA входят в состав ОС и могут быть установлены из терминала Fly или из графического менеджера пакетов.

Перед установкой службы FreeIPA необходимо убедиться, что клиентский компьютер не является клиентом другого домена (в частности, домена ALD).

Установить необходимые пакеты клиентской части из командной строки можно командой:

sudo apt install astra-freeipa-client

или командой (с использованием графического инструмента):

sudo apt install fly-admin-freeipa-client

В ходе установки в случае появления окна предупреждений - нажать "ОК".

Ввод клиентского компьютера в домен осуществляется командой:

sudo astra-freeipa-client

В качестве имени домена будет автоматически использовано доменное имя сервера DNS.

После выполнения команды будут выведены на экран все исходные данные. Для подтверждения данных ввести *y* и нажать <*Enter*>. После подтверждения появится запрос пароля администратора домена.

В случае успешного введения в домен клиента на экран будет выведено сообщение «Завершено».

После выполненных действий требуется перезагрузка системы. Ее можно осуществить командой:

sudo systemctl reboot

Рекомендуется осуществить проверку входа в систему на клиенте всех созданных доменных пользователей под доступными им мандатными уровнями.

4.2.2.7. Установка веб-сервера Apache2 в домене FreeIPA

Перед началом установки веб-сервера рекомендуется дополнительно ознакомиться с разделом 10 - «Защищенный комплекс программ гипертекстовой обработки данных» документа «Руководства администратора. Часть 1. РУСБ.10015-01 95 01-1».

На клиентской части FreeIPA (*p.astra.ntc*) необходимо зайти под учетной записью администратора FreeIPA (*admin*) и произвести установку пакетов веб-сервера Apache2:

sudo apt-get install apache2

Установить пакет модуля веб-сервера Apache2 - *auth_kerb* из пакета *libapache2-mod-auth-kerb* для аутентификации через Kerberos:

sudo apt-get install libapache2-mod-auth-kerb

Необходимо активировать модули *headers* и *cgi*:

sudo a2enmod headers

sudo a2enmod cgi

В конфигурационном файле */etc/apache2/apache2.conf* добавить строку *AstraMode on.*

В каталоге */etc/apache2/sites-available* должен находиться файл с настройками виртуальных хостов. Необходимо привести этот конфигурационный файл (в данном случае *000-default.conf)* к следующему виду:

Пример

*<VirtualHost *:80>*

ServerAdmin webmaster@localhost

DocumentRoot /var/www/html

<Directory /var/www/html>

AuthType Kerberos

KrbAuthRealms ASTRA.NTC

KrbServiceName HTTP/p.astra.ntc

Krb5Keytab /etc/apache2/http.keytab

KrbMethodNegotiate on

KrbMethodK5Passwd off

KrbSaveCredentials on

require valid-user

RequestHeader set MYMACLABEL "%m:%c"

AddHandler cgi-script .py

Options +ExecCGI

</Directory>

AddDefaultCharset UTF-8

SetEnv PYTHONIOENCODING utf8

ErrorLog ${APACHE_LOG_DIR}/error.log

CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>

Разрешить к исполнению файл виртуальных хостов можно командой (по умолчанию разрешен):

sudo a2ensite 000-default

Для корневого каталога виртуального хоста (по умолчанию */var/www/html*) и всех его родительских каталогов необходимо назначить мандатные атрибуты не меньше максимальных атрибутов объектов, к которым будет разграничиваться доступ:

sudo pdpl-file 3:0:0:ccnr /var/www/

sudo pdpl-file 3:0:0:ccnr /var/www/html/

После выполненных действий необходимо перезапустить веб-сервер Apache2, выполнив команду:

sudo systemctl restart apache2

Чтобы настроить автоматический запуск веб-сервера Apache2 при загрузке операционной системы, необходимо выполнить:

sudo systemctl enable apache2

В файле конфигрурации /etc/sssd/sssd.conf в параметр allowed_uids необходимо добавить пользователя www-data.

Браузер Mozilla Firefox, в котором будут работать пользователи, необходимо настроить на поддержку аутентификации *negotiate*. Для этого необходимо для каждого доменного пользователя зайти на клиентской части FreeIPA под всеми доступными уровнями в рабочие сессии, в каждой из них открыть настройки браузера Mozilla Firefox, набрав в адресной строке *about:config*, согласиться с предупреждением о безопасности, найти через поисковую строку параметр *network.negotiate-auth.trusted-uris* и *network.negotiate-auth.delegation-uris* и указать в качестве значения — *http://*.

4.2.2.8. Настройка веб-сервера Apache2 для совместной работы в домене FreeIPA

Сквозная аутентификация и авторизация пользователей в браузере обеспечивается настройкой веб-сервера и браузера для работы в ЕПП. Перед началом настройки веб-сервера рекомендуется дополнительно ознакомиться с инструкцией, представленной в документе «Руководство администратора. Часть 1. РУСБ.10015-01 95 01-1» п. 10.3 «Настройка авторизации».

Для обеспечения сквозной авторизации необходимо на серверной части FreeIPA (*q.astra.ntc*) зайти в систему под учетной записью администратора FreeIPA (*admin*) и создать с помощью утилиты администрирования FreeIPA принципала, соответствующего настраиваемому веб-серверу Apache2. Принципал создается с автоматически сгенерированным случайным ключом:

ipa service-add HTTP/p.astra.ntc

После этого на клиентской части FreeIPA (*p.astra.ntc*) зайти в систему под учетной записью администратора FreeIPA (*admin*) и создать файл ключа Kerberos для веб-сервера Apache2 с помощью утилиты администрирования FreeIPA:

sudo ipa-getkeytab -p HTTP/p.astra.ntc -k /etc/apache2/http.keytab

Полученный файл ключа должен быть доступен веб-серверу Apache2 по пути, указанному в параметре *Krb5Keytab* конфигурационного файла */etc/apache2/sites-available/000-default.conf (*т.е. */etc/apache2/http.keytab)*. Права доступа к ключевому файлу должны позволять читать его пользователю, от имени которого работает веб-сервер Apache2. Таким образом, на клиентской части FreeIPA необходимо сменить владельца файла */etc/apache2/http.keytab* на пользователя *www-data*, выполнив следующую команду:

sudo chown www-data /etc/apache2/http.keytab

и предоставить права на чтение файла */etc/apache2/keytab* остальным пользователям, выполнив команду:

sudo chmod 644 /etc/apache2/http.keytab

После выполненных действий необходимо перезапустить веб-сервер Apache2, выполнив команду:

sudo systemctl restart apache2

и перезагрузить систему клиентской части FreeIPA:

sudo systemctl reboot

4.2.2.9. Установка СУБД PostgreSQL в домене FreeIPA

На клиентской части FreeIPA (*p.astra.ntc*) необходимо зайти под учетной записью администратора FreeIPA (*admin*) и произвести установку пакетов СУБД PostgreSQL*:*

sudo apt install postgresql-astra

Для адаптации PostgreSQL для языка программирования Python3, необходимо также установить пакет *python3-psycopg2* (устанавливается с диска со средствами разработчика):

sudo apt install python3-psycopg2

Настройка сервера СУБД осуществляется установкой параметров в конфигурационном файле *postgresql.conf* В дополнение к файлу *postgresql.conf* в PostgreSQL используется еще два конфигурационных файла, которые контролируют аутентификацию клиента (*pg_hba.conf* - определяет конфигурационный файл для аутентифика-

ции по узлам, и *pg_ident.conf* - определяет конфигурационный файл для аутентификации по методу). По умолчанию все эти три файла находятся в каталоге данных кластера БД или в соответствующем кластеру конфигурационном каталоге, например, */etc/postgresql/11/main*. За расположение указанных файлов отвечают конфигурационные параметры.

Для обеспечения сквозной аутентификации пользователей ЕПП на клиентской части FreeIPA, где расположена СУБД, необходимо в файле */etc/postgresql/11/main/pg_hba.conf* для внешних соединений в качестве метода аутентификации указать *gss* и провести соответствующую настройку СУБД PostgreSQL для работы с ЕПП.

Файл конфигурации *pg_hba.conf* отредактировать следующим образом:

Пример

 # TYPE DATABASE USER CIDR-ADDRESS METHOD

local all postgres peer

host all postgres 127.0.0.1/32 trust

host all all 192.168.111.0/24 gss include_realm=0 krb_realm=ASTRA.NTC

где 192.168.111.0/24 – ЛВС домена, 24 – маска подсети

В конфигурационном файле сервера СУБД PostgreSQL */etc/postgreslq/11/main/postgresql.conf* для приведенных ниже параметров задать соответствующие значения:

Пример

ac_ignore_socket_maclabel = false

listen listen_addresses = ''*

krb_server_keyfile = '/etc/postgresql-common/krb5.keytab'

В файле конфигрурации /etc/sssd/sssd.conf в параметр allowed_uids необходимо добавить пользователя postgres.

4.2.2.10. Настройка СУБД PostgreSQL для совместной работы в домене FreeIPA

Перед началом настройки СУБД для совместной работы с FreeIPA рекомендуется дополнительно ознакомиться с п. 1.4.2 «Использование сквозной аутентифика-

ции в ЕПП» документа «Руководство администратора. Часть 2. РУСБ.10015-01 95 01-1».

Для обеспечения совместной работы сервера СУБД PostgreSQL с FreeIPA необходимо, чтобы сервер СУБД PostgreSQL функционировал как сервис Kerberos.

Для этого необходимо:

1) На серверной части FreeIPA (*q.astra.ntc*) зайти в систему под учетной записью администратора FreeIPA (*admin*) и создать в БД FreeIPA с помощью утилиты администрирования FreeIPA принципала, соответствующего устанавливаемому серверу PostgreSQL. Принципал создается с автоматически сгенерированным случайным ключом;

ipa service-add postgres/p.astra.ntc

2) На клиентской части FreeIPA (*p.astra.ntc*) необходимо зайти под учетной записью администратора FreeIPA (*admin*) и создать файл ключа Kerberos для сервера СУБД PostgreSQL с помощью утилиты администрирования FreeIPA *ipa service-add:*

sudo ipa-getkeytab --principal=postgres/p.astra.ntc --keytab=/etc/postgresql-common/krb5.keytab

Полученный файл должен быть доступен серверу СУБД PostgreSQL по пути, указанному в конфигурационном параметре *krb_server_keyfile* конфигурационного файла */etc/postgreslq//main/postgresql.conf* (в данном случае /etc/*postgresql-common/krb5.keytab*). Пользователю, от имени которого работает сервер СУБД PostgreSQL (по умолчанию *postgres*), должны быть предоставлены права на чтение данного файла;

3) назначить владельцем файла *krb5.keytab* пользователя *posgtres*, выполнив команду:

sudo chown postgres /etc/postgresql-common/krb5.keytab

По итогам выполненных действий перезагрузить сервис:

sudo systemctl restart postgresql

и перезагрузить систему клиентской части FreeIPA:

sudo systemctl reboot

4.2.2.11. Контрольный пример

Файлы контрольного примера должны располагаться на компьютере, выполняющем роль веб-сервера.

Для этого на клиентской части FreeIPA (*p.astra.ntc*) в каталоге */var/www/html* необходимо разместить файлы контрольного примера: contrprimer_0.sql, contrprimer_1.sql, contrprimer_2.sql, contrprimer_3.sql, contrprimer.html, contrprimer.py.

В файле *contrprimer.py* строка подключения к базе данных должна содержать корректное имя хоста, на котором располагается база данных. Необходимо отредактировать файл *contrprimer.py,* установив для параметра *DB_CONNECT* имя хоста - *p.astra.ntc*:

DB_CONNECT = 'dbname=contrprimer host=p.astra.ntc port=5432'

От имени пользователя *postgres* для СУБД PostgreSQL необходимо создать одноименных *доменным* пользователей, а также базу данных *contrprimer*:

Пример

sudo su postgres

psql

create user user0;

create user user1;

create user user2;

create user user3;

create database contrprimer;

\q

exit

Создание ролей для подключения, разграничение доступа в базе данных осуществляется выполнением sql-сценария контрольного примера. Для его использования следует перейти в каталог */var/www/html:*

cd /var/www/html

и выполнить команду:

sudo psql -h localhost -U postgres -f contrprimer_0.sql

с указанием имени сценария в качестве последнего параметра (contrprimer_0.sql, contrprimer_1.sql, contrprimer_2.sql, contrprimer_3.sql), выбранного в соответствии с одним из четырех вариантов демонстрации разграничения доступа. Описание вариантов генерации базы данных с использованием различных сценариев представлено в п. 3.

Запуск контрольного примера осуществляется на хосте, где расположен веб-сервер Apache2 (в данном случае, *p.astra.ntc*) по инструкции, приведенной в п. 5.

5. Запуск контрольного примера

Запуск контрольного примера осуществляется путем открытия веб-страницы контрольного примера, которая предоставляет графический интерфейс пользователя и обеспечивает взаимодействие с веб-сервером. Для этого необходимо зайти в систему одним из доменных пользователей через графическую оболочку и запустить браузер Mozilla Firefox. В адресной строке браузера необходимо указать адрес расположения файла *contrpimer.html* контрольного примера (рисунок 5.1). Например, для веб-сервера Apache2, расположенного на хосте q.astra.ntc, адрес будет следующий

http://q.astra.ntc/contrpimer.html

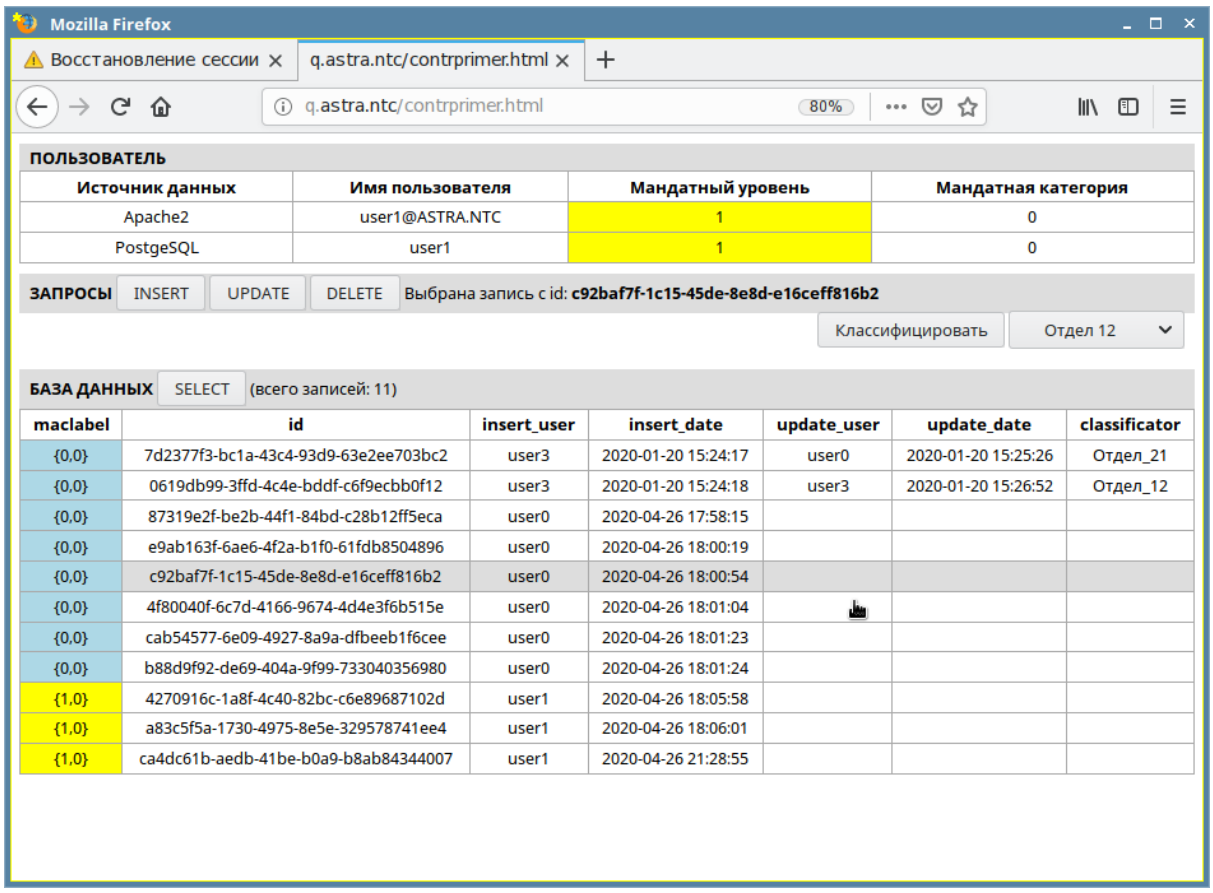

Рисунок 5.1 - Окно вывода контрольного примера

При успешной авторизации доступа пользователя к ресурсам управляемых узлов в таблице «Пользователь» будут отображены параметры реального контекста работы пользователя, полученные из служб контрольного примера (Apache, PostgreSQL): имя пользователя, осуществившего вход в домен (например, user1@ASTRA.NTC, user1), его мандатный уровень и мандатная категория. Ниже будет отображено содержимое базы данных контрольного примера. Для выполнения тестовых операций над данными предлагается использовать кнопки запросов.

При входе в систему другими пользователями домена результат вывода контекста пользователя и содержимого базы данных будет различным и соответствовать параметрам вошедшего с систему пользователя и его правам на доступ к данным, хранящимся в БД.