

Лабораторная работа № 10

Расширенные настройки SMTP-сервера

Митрофанов Тимур Александрович

Содержание

1 Цель работы	5
2 Задание	6
3 Выполнение лабораторной работы	7
4 Выводы	24
Список литературы	25
4.1 —	25

Список иллюстраций

3.1	Переход в режим суперпользователя на сервере	7
3.2	Запуск мониторинга логов почтовой службы	8
3.3	Активация протокола LMTP в основном конфиге Dovecot	9
3.4	Настройка LMTP сокета для взаимодействия с почтовым агентом	9
3.5	Переопределение транспорта доставки в настройках Postfix	10
3.6	Изменение формата имени пользователя для аутентификации	10
3.7	Перезапуск почтовых служб для применения настроек	11
3.8	Подтверждение успешной доставки почты в системных логах	11
3.9	Просмотр полученного тестового письма в почтовом ящике	12
3.10	Конфигурирование службы аутентификации SASL в Dovecot	12
3.11	Связывание Postfix с механизмом аутентификации Dovecot	13
3.12	Настройка политик безопасности и ограничений для получателей	13
3.13	Ограничение списка доверенных сетей в параметрах Postfix	14
3.14	Редактирование параметров SMTP сервиса в файле master.cf	14
3.15	Перезагрузка служб после настройки механизмов безопасности	15
3.16	Установка утилиты telnet на клиентской машине	15
3.17	Генерация строки аутентификации в формате base64	15
3.18	Имитация процесса аутентификации через telnet сессию	16
3.19	Получение подтверждения успешной авторизации от сервера	17
3.20	Копирование сертификатов для настройки шифрования TLS	17
3.21	Конфигурирование параметров TLS шифрования в Postfix	18
3.22	Настройка правил межсетевого экрана для нового порта	19
3.23	Финальный перезапуск Postfix после настройки TLS	19
3.24	Проверка защищенного TLS соединения через openssl	20
3.25	Копирование конфигураций в каталог автоматизации Vagrant	21
3.26	Обновление скрипта автоматической настройки сервера	22
3.27	Редактирование скрипта инициализации клиентской машины	23

Список таблиц

1 Цель работы

Приобретение практических навыков по конфигурированию SMTP-сервера в части настройки аутентификации.

2 Задание

1. Настройте Dovecot для работы с LMTP.
2. Настройте аутентификацию посредством SASL на SMTP-сервере.
3. Настройте работу SMTP-сервера поверх TLS.
4. Скорректируйте скрипт для Vagrant, фиксирующий действия расширенной настройки SMTP-сервера во внутреннем окружении виртуальной машины server

3 Выполнение лабораторной работы

На начальном этапе выполнения работы осуществляется вход в систему и переход в режим суперпользователя. Это необходимо для получения неограниченного доступа к конфигурационным файлам и возможности управления системными службами почтового сервера (рис. 3.1).

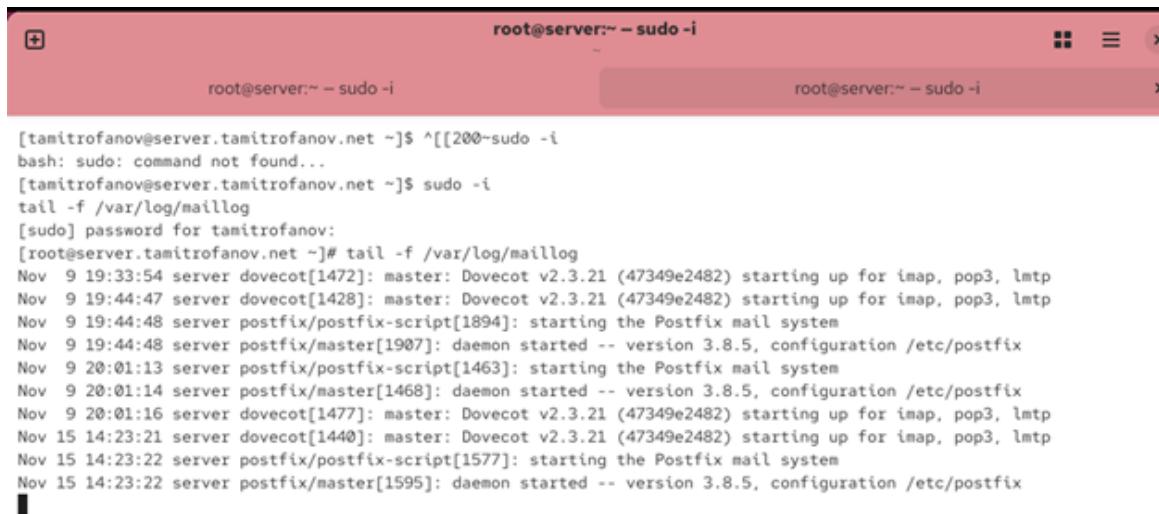
Использование данной команды позволяет избежать проблем с правами доступа при редактировании критически важных объектов в каталоге dovecot и postfix. На экране виден терминал с приглашением командной строки для пользователя root, что подтверждает готовность среды к дальнейшим манипуляциям.



Рисунок 3.1: Переход в режим суперпользователя на сервере

Для оперативного отслеживания изменений в работе почтовой системы запускается мониторинг лог-файла в реальном времени. В окне терминала отображаются последние записи о запуске служб Dovecot и Postfix, что позволяет сразу увидеть возможные ошибки конфигурации (рис. 3.2).

Мониторинг осуществляется через просмотр файла maillog, где фиксируются все события- от инициализации демонов до передачи конкретных сообщений. На данный момент службы запущены в стандартном режиме и готовы к приему команд для настройки протокола локальной доставки.



```
[tmitrofanov@server.tmitrofanov.net ~]$ ^[[200-sudo -i
bash: sudo: command not found...
[tmitrofanov@server.tmitrofanov.net ~]$ sudo -i
tail -f /var/log/maillog
[sudo] password for tmitrofanov:
[root@server.tmitrofanov.net ~]# tail -f /var/log/maillog
Nov  9 19:33:54 server dovecot[1472]: master: Dovecot v2.3.21 (47349e2482) starting up for imap, pop3, lmtp
Nov  9 19:44:47 server dovecot[1428]: master: Dovecot v2.3.21 (47349e2482) starting up for imap, pop3, lmtp
Nov  9 19:44:48 server postfix/postfix-script[1894]: starting the Postfix mail system
Nov  9 19:44:48 server postfix/master[1907]: daemon started -- version 3.8.5, configuration /etc/postfix
Nov  9 20:01:13 server postfix/postfix-script[1463]: starting the Postfix mail system
Nov  9 20:01:14 server postfix/master[1468]: daemon started -- version 3.8.5, configuration /etc/postfix
Nov  9 20:01:16 server dovecot[1477]: master: Dovecot v2.3.21 (47349e2482) starting up for imap, pop3, lmtp
Nov 15 14:23:21 server dovecot[1440]: master: Dovecot v2.3.21 (47349e2482) starting up for imap, pop3, lmtp
Nov 15 14:23:22 server postfix/postfix-script[1577]: starting the Postfix mail system
Nov 15 14:23:22 server postfix/master[1595]: daemon started -- version 3.8.5, configuration /etc/postfix
```

Рисунок 3.2: Запуск мониторинга логов почтовой службы

Следующим шагом в основном конфигурационном файле Dovecot активируется протокол LMTP. Это расширение позволяет организовать более эффективную локальную пересылку почты и применять фильтрацию на стороне сервера в момент доставки (рис. 3.3).

В секции protocols добавляется значение lmtp рядом с уже существующими imap и pop3. Данное действие сообщает серверу о необходимости запуска соответствующего сервиса для взаимодействия с агентом передачи почты в лице Postfix.

```

# Protocols we want to be serving.
#protocols = imap pop3 lmtp submission

protocols = imap pop3 lmtp

# A comma separated list of IPs or hosts where to
# "*" listens in all IPv4 interfaces, ":::" listens
# If you want to specify non-default ports or anyt
# edit conf.d/master.conf.

```

Рисунок 3.3: Активация протокола LMTP в основном конфиге Dovecot

Для обеспечения взаимодействия между Dovecot и Postfix настраивается специальный unix-слушатель. В конфигурационном файле master.conf задаются параметры сокета, через который будет происходить обмен данными между службами (рис. 3.4).

Устанавливаются права доступа 0600, а также назначаются владелец и группа postfix. Это гарантирует безопасность передачи данных, так как доступ к сокету будет иметь только почтовый агент, что предотвращает несанкционированный перехват сообщений внутри системы.

```

}

service lmtp {
    unix_listener /var/spool/postfix/private/dovecot-lmtp {
        group = postfix
        user = postfix
        mode = 0600
    }
}

service imap {
    # Most of the memory goes to mmap'ing files. You may need to increase this
}

```

Рисунок 3.4: Настройка LMTP сокета для взаимодействия с почтовым агентом

После настройки сокета на стороне Dovecot необходимо указать Postfix использовать именно этот транспорт для доставки сообщений в почтовые ящики. С помощью специальной утилиты управления конфигурацией задается параметр транспорта (рис. 3.5).

Команда устанавливает использование lmtp через указанный ранее путь к сокету. Это позволяет отказаться от прямой записи в файлы и перейти к более современному

му и гибкому механизму управления почтовыми очередями на этапе финальной доставки.

```
[root@server.tamitrofanov.net ~]#  
[root@server.tamitrofanov.net ~]# postconf -e 'mailbox_transport = lmtp:unix:private/dovecot-lmtp'  
[root@server.tamitrofanov.net ~]#
```

Рисунок 3.5: Переопределение транспорта доставки в настройках Postfix

Для корректной работы механизмов авторизации в файле настроек аутентификации изменяется формат имени пользователя. Установка специального шаблона позволяет отсекать доменную часть, оставляя только логин (рис. 3.6).

Это изменение важно для унификации доступа, чтобы система воспринимала пользователя tamitrofanov одинаково, независимо от того, указано ли в почтовом клиенте полное имя с доменом или краткое. На скриншоте виден процесс редактирования параметра в текстовом редакторе.

```
# Username formatting before it's looked up from databases. You can use  
# the standard variables here, eg. %Lu would lowercase the username, %n would  
# drop away the domain if it was given, or "%n-AT-%d" would change the '@' into  
# "-AT-". This translation is done after auth_username_translation changes.  
auth_username_format = %Ln
```



```
# If you want to allow master users to log in by specifying the master  
# username within the normal username string (ie. not using SASL mechanism's  
# support for it), you can specify the separator character here. The format  
# is then <username><separator><master username>. UW-IMAP uses "*" as the  
# separator, so that could be a good choice.  
#auth_master_user_separator =
```

Рисунок 3.6: Изменение формата имени пользователя для аутентификации

Чтобы внесенные изменения вступили в силу, производится перезапуск обеих ключевых служб. Сначала перезапускается Postfix для применения новых настроек транспорта, а затем Dovecot для инициализации сервиса LMTP (рис. 3.7).

Использование системного менеджера служб позволяет убедиться, что демоны корректно завершили работу и запустились снова с обновленными параметрами. Отсутствие сообщений об ошибках в консоли косвенно подтверждает правильность правок в текстовых файлах.

```
[root@server.tamitrofanov.net ~]#  
[root@server.tamitrofanov.net ~]#  
[root@server.tamitrofanov.net ~]#  
[root@server.tamitrofanov.net ~]# systemctl restart postfix  
systemctl restart dovecot  
[root@server.tamitrofanov.net ~]# █
```

Рисунок 3.7: Перезапуск почтовых служб для применения настроек

В окне мониторинга логов фиксируется первая успешная доставка сообщения через новый протокол. В записях отчетливо видна цепочка событий- соединение с локальным сокетом, передача данных и итоговый статус о сохранении письма в INBOX (рис. 3.8).

Важным моментом является идентификатор сессии LMTP, который подтверждает, что доставка была выполнена именно через настроенный механизм. Логи также показывают, что сообщение было принято от клиента и успешно обработано сервером без задержек.

```
Nov 15 14:53:13 server dovecot[7783]: master: Dovecot v2.3.21 (47349e2482) starting up for imap, pop3, lmtp  
Nov 15 14:55:00 server postfix/smtpd[8167]: connect from client.tamitrofanov.net[192.168.1.30]  
Nov 15 14:55:00 server postfix/smtpd[8167]: D1E1121BDBAA: client=client.tamitrofanov.net[192.168.1.30]  
Nov 15 14:55:00 server postfix/cleanup[8171]: D1E1121BDBAA: message-id=<20251115145500.A7339412A2E0@client.tamitrofan  
ov.net>  
Nov 15 14:55:00 server postfix/smtpd[8167]: disconnect from client.tamitrofanov.net[192.168.1.30] ehlo=2 starttls=1 m  
ail=1 rcpt=1 data=1 quit=1 commands=7  
Nov 15 14:55:00 server postfix/qmgr[7772]: D1E1121BDBAA: from=<tamitrofanov@client.tamitrofanov.net>, size=587, nrcpt  
=1 (queue active)  
Nov 15 14:55:00 server postfix/local[8173]: D1E1121BDBAA: passing <tamitrofanov@tamitrofanov.net> to transport=lmtp  
Nov 15 14:55:00 server dovecot[7785]: lmtp(8177): Connect from local  
Nov 15 14:55:00 server postfix/lmtp[8174]: D1E1121BDBAA: to=<tamitrofanov@tamitrofanov.net>, relay=server.tamitrofan  
ov.net[private/dovecot-lmtp], delay=0.11, delays=0.03/0.01/0.04/0.03, dsn=2.0.0, status=sent (250 2.0.0 <tamitrofanov@  
tamitrofanov.net> yBsMN0SUGGnxHwAAYT2eVQ Saved)  
Nov 15 14:55:00 server postfix/qmgr[7772]: D1E1121BDBAA: removed  
Nov 15 14:55:00 server dovecot[7785]: lmtp(tamitrofanov)<8177><yBsMN0SUGGnxHwAAYT2eVQ>: msgid=<20251115145500.A733941  
2A2E0@client.tamitrofanov.net>; saved mail to INBOX  
Nov 15 14:55:00 server dovecot[7785]: lmtp(8177): Disconnect from local: Logged out (state=READY)
```

Рисунок 3.8: Подтверждение успешной доставки почты в системных логах

Проверка содержимого почтового ящика через консольную утилиту подтверждает фактическое получение письма. В списке входящих отображается новое сообщение с темой LMTP test, отправленное ранее для проверки работоспособности (рис. 3.9).

На скриншоте виден интерфейс почтового клиента, где письмо помечено как новое. Это является финальным подтверждением того, что связка Postfix и Dovecot через LMTP транспорт работает корректно и почта попадает по назначению в каталог пользователя.

```
>N 2 tamitrofanov@client. 2025-11-15 14:55 21/901 "LMTP test
&
```

Рисунок 3.9: Просмотр полученного тестового письма в почтовом ящике

Переходим к настройке механизмов SASL для обеспечения безопасности. В файле master.conf определяется служба аутентификации, которая позволит Postfix проверять учетные данные пользователей через базу Dovecot (рис. 3.10).

Настраиваются два слушателя- один для внутреннего обмена данными с Postfix, другой для работы с базой данных пользователей. Установка прав 0660 для первого сокета позволяет группам postfix и dovecot безопасно обмениваться информацией о подлинности клиентов.

```
service auth {
    unix_listener /var/spool/postfix/private/auth {
        group = postfix
        user = postfix
        mode = 0660
    }
    unix_listener auth-userdb {
        mode = 0600
        user = dovecot
    }
}

# Postfix smtp-auth
#unix_listener /var/spool/postfix/private/auth {
```

Рисунок 3.10: Конфигурирование службы аутентификации SASL в Dovecot

С помощью команд управления конфигурацией Postfix настраивается на использование механизмов Dovecot для проверки подлинности. Указывается тип аутентификации и путь к ранее созданному сокету в приватном каталоге (рис. 3.11).

Эти параметры связывают две службы в единый механизм обеспечения безопасности. Теперь при попытке отправить почту через сервер, Postfix будет обращаться к Dovecot за подтверждением легитимности пользователя, используя стандартные протоколы взаимодействия.

```
[root@server.tamitrofanov.net ~]#  
[root@server.tamitrofanov.net ~]# postconf -e 'smtpd_sasl_type = dovecot'  
postconf -e 'smtpd_sasl_path = private/auth'  
[root@server.tamitrofanov.net ~]# █
```

Рисунок 3.11: Связывание Postfix с механизмом аутентификации Dovecot

Важнейшим этапом является настройка ограничений для получателей сообщений. Формируется список правил, которые определяют, кому разрешено пересылать почту, а какие запросы должны быть отклонены для борьбы со спамом и несанкционированным использованием релея (рис. 3.12).

В список включаются проверки на существование домена получателя, проверку локальных сетей и обязательное требование аутентификации для внешних клиентов. Порядок следования этих опций критичен для правильной работы фильтрации на входе.

```
[root@server.tamitrofanov.net ~]#  
[root@server.tamitrofanov.net ~]# postconf -e 'smtpd_recipient_restrictions = reject_unknown_recipient_domain, permit  
_mynetworks, reject_non_fqdn_recipient, reject_unauth_destination, reject_unverified_recipient, permit'
```

Рисунок 3.12: Настройка политик безопасности и ограничений для получателей

Ограничиваются список доверенных сетей только локальным интерфейсом сервера. Это стандартная мера безопасности, предотвращающая использование сервера как открытого релея для рассылки сообщений из интернета (рис. 3.13).

Установка параметра `mynetworks` в значение локальной петли гарантирует, что любая попытка отправки почты не с самого сервера потребует обязательного прохождения процедуры аутентификации. На скриншоте зафиксировано выполнение команды обновления настроек.

```
[root@server.tamitrofanov.net ~]#  
[root@server.tamitrofanov.net ~]# postconf -e 'mynetworks = 127.0.0.0/8'  
[root@server.tamitrofanov.net ~]#
```

Рисунок 3.13: Ограничение списка доверенных сетей в параметрах Postfix

В файле описания сервисов Postfix редактируются параметры демона smtp. Добавляются опции, активирующие поддержку SASL и применяющие специфические ограничения для входящих соединений на 25 порту (рис. 3.14).

Внесение изменений непосредственно в конфигурацию сервиса позволяет переопределить глобальные настройки для конкретного порта. Здесь активируется включение проверки подлинности и дублируются правила фильтрации получателей для надежности.

```
#      (узн) (узн) (узн) (узн) (узн)  
# *****  
smtp inet n - - smtpd  
#smtp  inet  n  -   n    -     1    postscreen  
#smtpd  pass  -   -   n    -     smtpd  
#dnsblog unix  -   -   n    -     0    dnsblog  
#tlsproxy unix  -   -   n    -     0    tlsproxy  
# Choose one: enable submission for loopback clients only, or for any client.  
#127.0.0.1:submission inet n -   n    -   -     smtpd  
#submission inet n  -   n    -   -     smtpd  
#  -o syslog_name=postfix/submission  
#  -o smtpd_tls_security_level=encrypt  
#  -o smtpd_sasl_auth_enable=yes  
#  -o smtpd_recipient_restrictions=reject_non_fqdn_recipient,reject_unknown_recipient_domain,permit_sasl_authenticated  
#  -o smtpd_tls_auth_only=yes  
#  -o local_header_rewrite_clients=static:all  
#  -o smtpd_reject_unlisted_recipient=no  
#    Instead of specifying complex smtod <xxx> restrictions here.
```

Рисунок 3.14: Редактирование параметров SMTP сервиса в файле master.cf

После завершения настройки SASL вновь производится перезапуск сервисов для активации новых функций. Теперь сервер готов проверять учетные данные клиентов перед тем, как принять от них сообщение для дальнейшей пересылки (рис. 3.15).

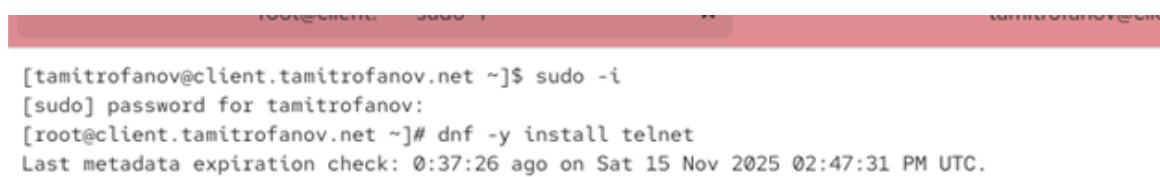
Консоль показывает последовательное выполнение команд рестарта. Важно, чтобы обе службы были перезапущены, так как они теперь зависят друг от друга в вопросах проверки прав доступа пользователей.

```
[root@server.tamitrofanov.net ~]# systemctl restart postfix
[root@server.tamitrofanov.net ~]# systemctl restart dovecot
[root@server.tamitrofanov.net ~]#
```

Рисунок 3.15: Перезагрузка служб после настройки механизмов безопасности

На стороне клиента подготавливается инструментарий для тестирования аутентификации. С помощью пакетного менеджера устанавливается утилита telnet, которая позволяет имитировать работу почтового клиента в ручном режиме (рис. 3.16).

Установка производится из репозиториев операционной системы. Наличие данного инструмента необходимо для проверки того, как сервер отвечает на команды авторизации и выдает ли он правильные заголовки в ответ на приветствие EHLO.



```
[root@client.tamitrofanov.net ~]$ sudo -i
[sudo] password for tamitrofanov:
[root@client.tamitrofanov.net ~]# dnf -y install telnet
Last metadata expiration check: 0:37:26 ago on Sat 15 Nov 2025 02:47:31 PM UTC.
```

Рисунок 3.16: Установка утилиты telnet на клиентской машине

Для прохождения аутентификации по протоколу PLAIN необходимо подготовить строку с логином и паролем в кодировке base64. С помощью командной строки формируется специальный зашифрованный блок данных (рис. 3.17).

Строка содержит имя пользователя и пароль, разделенные нулевыми байтами, что соответствует требованиям протокола. Полученный результат в дальнейшем будет скопирован и вставлен в сессию telnet для подтверждения личности пользователя перед сервером.

```
[root@client.tamitrofanov.net ~]#
[root@client.tamitrofanov.net ~]# printf 'tamitrofanov\x00tamitrofanov\x00123456' | base64
dGFtaXRyb2Zhbm92AHRhbWl0cm9mYW5vdgAxMjM0NTY=
[root@client.tamitrofanov.net ~]#
```

Рисунок 3.17: Генерация строки аутентификации в формате base64

Осуществляется подключение к почтовому серверу через telnet на стандартный 25

порт. В ходе сессии отправляется команда приветствия, после чего сервер сообщает о поддержке метода аутентификации PLAIN (рис. 3.18).

На скриншоте виден диалог- клиент отправляет EHLO, а сервер выдает список своих возможностей. В конце вводится команда AUTH с ранее сгенерированной строкой, что инициирует процесс проверки подлинности на стороне Dovecot.

```
[root@client.tamitrotanov.net ~]# telnet server.tamitrofanov.net 25
Trying 192.168.1.1...
Connected to server.tamitrofanov.net.
Escape character is '^].
220 server.tamitrofanov.net ESMTP Postfix
EHLO test
250-server.tamitrofanov.net
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-AUTH PLAIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-DSN
250-SMTPUTF8
250 CHUNKING
AUTH PLAIN dXNlcgB1c2VyADEyMzQ1Ng==^C^C^C
AUTH PLAIN dGFtaXRyb2Zhbmc92AHRhbWl0cm9mYW5vdgAxMjM0NTY=
-
```

Рисунок 3.18: Имитация процесса аутентификации через telnet сессию

Сервер возвращает статус об успешном прохождении проверки подлинности. Код 235 означает, что введенные данные верны и пользователь теперь имеет право отправлять сообщения через данный сервер (рис. 3.19).

Это важный этап тестирования, доказывающий, что настроенная ранее связка SASL между Postfix и Dovecot работает правильно. Теперь сервер защищен от анонимных рассылок и требует идентификации для работы с почтой.

```
220 server.tamitrofanov.net ESMTP Postfix
AUTH PLAIN dGFtaXRyb2Zhbmc92AHRhbWl0cm9mYW5vdgAxMjM0NTY=
235 2.7.0 Authentication successful
```

Рисунок 3.19: Получение подтверждения успешной авторизации от сервера

Для настройки защищенного соединения TLS производится копирование сертификатов. Временные сертификаты, созданные при установке Dovecot, переносятся в системные каталоги TLS для использования их почтовым сервером Postfix (рис. 3.20).

Копируются как сам публичный сертификат, так и приватный ключ. Это позволяет активировать шифрование трафика, чтобы пароли и содержимое писем не передавались по сети в открытом виде. На скриншоте видны команды копирования файлов.

```
[root@server.tamitrofanov.net ~]# cp /etc/pki/dovecot/certs/dovecot.pem /etc/pki/tls/certs
cp /etc/pki/dovecot/private/dovecot.pem /etc/pki/tls/private
[root@server.tamitrofanov.net ~]#
[root@server.tamitrofanov.net ~]#
[root@server.tamitrofanov.net ~]#
[root@server.tamitrofanov.net ~]#
[root@server.tamitrofanov.net ~]# postconf -e 'smtpd_tls_cert_file=/etc/pki/tls/certs/dovecot.pem'
postconf -e 'smtpd_tls_key_file=/etc/pki/tls/private/dovecot.pem'
postconf -e 'smtpd_tls_session_cache_database = btree:/var/lib/postfix/smtpd_scache'
postconf -e 'smtpd_tls_security_level = may'
postconf -e 'smtp_tls_security_level = may'
[root@server.tamitrofanov.net ~]#
```

Рисунок 3.20: Копирование сертификатов для настройки шифрования TLS

Производится детальная настройка параметров TLS в конфигурации Postfix. Указываются пути к файлам сертификата и ключа, а также настраивается база данных для кэширования сессий шифрования (рис. 3.21).

Устанавливается режим безопасности may, который позволяет серверу предлагать шифрование, но сохранять совместимость с клиентами, которые его не поддерживают. Это обеспечивает баланс между безопасностью и доступностью почтового сервиса.

```

#
# =====
# service type  private unpriv  chroot  wakeup  maxproc command + args
#          (yes)  (yes)  (no)   (never) (100)
# =====
smtp inet n - - smtpd
#smtp  inet  n      -      n      -      1      postscreen
#smtpd  pass  -      -      n      -      -      smtpd
#dnsblog  unix  -      -      n      -      0      dnsblog
#tlsproxy  unix  -      -      n      -      0      tlsproxy
# Choose one: enable submission for loopback clients only, or for any client.
#127.0.0.1:submission  inet n - - - - smtpd
submission  inet n      -      n      -      -      smtpd

-o smtpd_tls_security_level=encrypt
-o smtpd_sasl_auth_enable=yes
-o smtpd_recipient_restrictions=reject_non_fqdn_recipient,reject_unknown_recipient_domain,permit_sasl_authenticated,reject
# -o syslog_name=postfix/submission
# -o smtpd_tls_security_level=encrypt
# -o smtpd_sasl_auth_enable=yes
# -o smtpd_recipient_restrictions=reject_non_fqdn_recipient,reject_unknown_recipient_domain,permit_sasl_authenticated,reject
# -o smtpd_enforce_tls=yes

```

Рисунок 3.21: Конфигурирование параметров TLS шифрования в Postfix

В файле конфигурации сервисов активируется специальный порт 587 для приема почты от клиентов (submission). Этот порт по умолчанию настроен на обязательное использование TLS и аутентификации.

Редактирование master.cf позволяет четко разделить трафик между серверами и трафик от конечных пользователей. Для порта 587 задаются жесткие политики безопасности, включая принудительное шифрование сессии перед началом передачи данных.

Для обеспечения доступности нового сервиса извне настраивается межсетевой экран. В список разрешенных служб добавляется smtp-submission, после чего правила фильтрации перезагружаются

Этот шаг необходим, чтобы сетевые фильтры операционной системы не блокировали входящие соединения на порту 587. На скриншоте виден расширенный список сервисов, в котором теперь присутствует нужная запись, и подтверждение успешного применения настроек.

```
[root@server.tamitrofanov.net ~]# firewall-cmd --get-services
firewall-cmd --add-service=smtp-submission
firewall-cmd --add-service=smtp-submission --permanent
firewall-cmd --reload
0-AD RH-Satellite-6 RH-Satellite-6-capsule afp alvr amanda-client amanda-k5-client amqp amqps anno-1602 anno-1800 apcupsd a
seqnet audit ausweisapp2 bacula bacula-client bareos-director bareos-filedaemon bareos-storage bb bgn bitcoin bitcoin-rpc b
itcoin-testnet bitcoin-testnet-rpc bittorrent-lsd ceph ceph-exporter ceph-mon cfengine checkmk-agent civilization-iv civili
zation-cockpit collectd condor-collector cratedb ctdb dds dds-multicast dds-unicast dhcp dhcpcv6 dhcpcv6-client distcc dns
dns-over-quic dns-over-tls docker-registry docker-swarm dropbox-lansync elasticsearch etcd-client etcd-server factorio fmg
er foreman foreman-proxy freeipa-4 freeipa-ldap freeipa-ldaps freeipa-replication freeipa-trust ftp galera ganglia-client g
anglia-master git gpgsql grafana gre high-availability http http3 https ident imap imaps iperf2 iperf3 ipfs ipp ipp-client ip
sec irc icsci-target isns jenkins kadmin kdeconnect kerberos kibana klogin kpasswd kprop kshell kube-api kube-apiserve
r kube-control-plane kube-control-plane-secure kube-controller-manager kube-controller-manager-secure kube-nodeport-service
s kube-scheduler kube-scheduler-secure kube-worker kubelet kubelet-readonly kubelet-worker ldap ldaps libvirt libvirt-tls l
ightning-network llmnr llmnr-client llmnr-tcp llmnr-udp managesieve matrix mdns memcache minecraft minidlna mndp mongodb mo
sh mountd mpd mqtt mqtt-tls ms-wbt mssql murmur mysql nbd nebula need-for-speed-most-wanted netbios-ns netdata-dashboard nf
s nfs3 nmea-0183 nrpe ntp nut opentelemetry openvpn ovirt-imageio ovirt-storageconsole ovirt-vmconsole plex pmcd pmproxy pm
webapi pmwebapis pop3 pop3s postgresql privoxy prometheus prometheus-node-exporter proxy-dhcp ps2link ps3netsrv ptp pulseau
dio puppetmaster quassel radius radsec rdp redis redis-sentinel rootd rpc-bind rquotad rsh rsyncd rtsp salt-master samba sa
mba-client samba-dc sane settlers-history-collection sip sips slimevr slp smtp smtp-submission smtsp snmp snmpstls snmpstls-t
rap snmptrap spideroak-lansync spotify-sync squid ssdp ssh ssh-custom stlsvr steam-elan-transfer steam-streaming stellaris
stronghold-crusader stun stuns submission supertuxkart svdrp svn syncthing syncthing-gui syncthing-relay synergy syscomlan
syslog syslog-tls telnet tentacle terraria tftp tile38 tinc tor-socks transmission-client turn turns upnp-client vdsm vnc-s
erver vrrp warpinator wbem-http wbem-https wireguard ws-discovery ws-discovery-client ws-discovery-host ws-discovery-tcp ws
-discovery-udp wsdd wsdd-http wsman wsmans xdmcp xmpp-bosh xmpp-client xmpp-local xmpp-server zabbix-agent zabbix-java-gate
way zabbix-server zabbix-trapper zabbix-web-service zero-k zerotier
success
success
success
```

Рисунок 3.22: Настройка правил межсетевого экрана для нового порта

После внесения всех изменений в настройки безопасности и портов производится финальный перезапуск Postfix. Теперь сервер полноценно поддерживает прием почты через защищенный протокол с обязательной проверкой подлинности (рис. 3.22).

Стабильный перезапуск без системных предупреждений свидетельствует о корректности синтаксиса всех добавленных параметров. Сервер готов к финальному тестированию защищенного соединения с использованием специализированных утилит.

```
success
[root@server.tamitrofanov.net ~]# systemctl restart postfix
```

Рисунок 3.23: Финальный перезапуск Postfix после настройки TLS

Для проверки TLS используется утилита openssl, которая позволяет установить защищенное соединение и просмотреть параметры сертификата. В выводе команды отображается информация о шифре, протоколе и успешной авторизации (рис. 3.23).

На скриншоте видна подробная техническая информация о сессии, включая тип

шифрования AES-256. После установления канала выполняется команда аутентификации, которая также завершается успешно, подтверждая полную работоспособность всех систем защиты.

```
Cipher      : TLS_AES_256_GCM_SHA384
Session-ID: 0E7D9ADD25B5AA976539E5387A1E334F3BA22E58C6C455720A11E938A0014C84
Session-ID-ctx:
Resumption PSK: 8F4DEBC8815E0F688E4C81976C5616DFB98C560CD15758FF361A2C9AB4C03FBE67A0610C59E36943D9A4BF0A262D0AFD
PSK identity: None
PSK identity hint: None
SRP username: None
TLS session ticket lifetime hint: 7200 (seconds)
TLS session ticket:
0000 - 70 60 d8 9e ef 2c 0a 08-6d 25 0f ea 3f 28 d5 fc p`....,..m%..?{..
0010 - 65 1a a7 79 98 17 7e 79-14 8a a6 7c 74 46 d2 bc e..y..~y...|tF..
0020 - 5d 74 d9 7f 03 36 e7 b6-70 36 c9 ea 8d 92 06 18 ]t...6..p6.....
0030 - ed 43 06 17 09 2c 5d 36-57 c0 68 c3 52 2f a9 63 .C...,]6W.h.R/.c
0040 - ad 23 4c fd 5a 86 fe fa-d2 72 bc 63 17 c7 43 90 .#L.Z....r.c..C.
0050 - a4 b0 38 5f 85 f9 9a e5-9d ca 64 e4 5f b2 c1 e8 ..8.....d._...
0060 - f4 fc ee c8 ce fc ad e7-33 e2 e1 b8 a3 4e ac 7e .....3...N.~
0070 - a4 34 9a a0 50 3c b4 d4-eb 13 1b eb bd 10 b1 04 .4..P<.....
0080 - da 32 11 93 1e 1a 9b 0f-ac 3f 15 74 44 ae ff d7 .2.....?..tD...
0090 - c6 c2 d4 70 b9 3d e1 15-ae fc 16 3a 3a d3 9b 2b ...p.=....::..+
00a0 - ca a9 fb 0f 48 a5 86 98-e2 93 23 d4 ff 1f 28 db ....H.....#...(..
00b0 - 1f a7 5e 91 82 fa 47 d0-e5 bd 99 35 79 86 1c a4 ..^...G....5y...
00c0 - af c8 ee a9 77 e6 73 10-ce 78 a6 36 af fe 5f ab ....w.s..x.6..._.

Start Time: 1763221123
Timeout   : 7200 (sec)
Verify return code: 18 (self-signed certificate)
Extended master secret: no
Max Early Data: 0
---
read R BLOCK
EHLO test
250-server.tamitrofanov.net
250 PIPELINING
250 SIZE 10240000
250 VRFY
250 ETRN
250 AUTH PLAIN
250 ENHANCEDSTATUSCODES
250 8BITMIME
250 DSN
250 SMTPUTF8
250 CHUNKING
AUTH PLAIN dGFtaXRyb2Zhbmc92AHRhbWl0cm9mYW5vdgAxMjM0NTY=
235 2.7.0 Authentication successful
```

Рисунок 3.24: Проверка защищенного TLS соединения через openssl

Для обеспечения сохранности настроек в среде автоматизации конфигурационные файлы копируются в каталог синхронизации Vagrant. Это позволит автоматически применять эти настройки при развертывании виртуальной машины в будущем (рис. 3.24).

Процесс включает создание необходимых директорий и перенос файлов Dovecot

и Postfix с подтверждением перезаписи старых версий. Таким образом, текущее состояние сервера фиксируется в репозитории проекта для дальнейшего использования.

```
[root@server.tamitrofanov.net ~]# cd /vagrant/provision/server
cp -R /etc/dovecot/dovecot.conf /vagrant/provision/server/mail/etc/dovecot/
cp -R /etc/dovecot/conf.d/10-master.conf /vagrant/provision/server/mail/etc/dovecot/conf.d/
cp -R /etc/dovecot/conf.d/10-auth.conf /vagrant/provision/server/mail/etc/dovecot/conf.d/
mkdir -p /vagrant/provision/server/mail/etc/postfix/
cp -R /etc/postfix/master.cf /vagrant/provision/server/mail/etc/postfix/
cp: overwrite '/vagrant/provision/server/mail/etc/dovecot/dovecot.conf'? y
cp: overwrite '/vagrant/provision/server/mail/etc/dovecot/conf.d/10-auth.conf'? y
[root@server.tamitrofanov.net server]#
[root@server.tamitrofanov.net server]#
[root@server.tamitrofanov.net server]# █
```

Рисунок 3.25: Копирование конфигураций в каталог автоматизации Vagrant

Редактируется скрипт автоматической настройки сервера. В него вносятся все команды управления конфигурацией через postconf и инструкции по управлению службами, которые выполнялись вручную в ходе работы (рис. 3.25).

Это превращает ручные действия в воспроизводимый код. Скрипт дополняется секциями настройки аутентификации, TLS и параметров LMTP, что гарантирует идентичность настроек при каждом новом запуске виртуального окружения.

```
GNU nano 8.1                               /vagrant/provision/server/mail.sh                         Modified
firewall-cmd --add-service imaps --permanent
firewall-cmd --add-service smtp-submission --permanent
firewall-cmd --reload
echo "Start postfix service"
systemctl enable postfix
systemctl start postfix

echo "Configure postfix"
postconf -e 'mydomain = user.net'
postconf -e 'myorigin = $mydomain'
postconf -e 'inet_protocols = ipv4'
postconf -e 'inet_interfaces = all'
postconf -e 'mydestination = $myhostname, localhost.$mydomain, localhost,
$mydomain'*
#postconf -e 'mynetworks = 127.0.0.0/8, 192.168.0.0/16'
echo "Configure postfix for dovecot"
postconf -e 'home_mailbox = Maildir/'
echo "Configure postfix for auth"
postconf -e 'smtpd_sasl_type = dovecot'
postconf -e 'smtpd_sasl_path = private/auth'

postconf -e 'smtpd_recipient_restrictions = reject_unknown_recipient_domain, permit_mynetworks, reject_non_fqdn_recipient,'
postconf -e 'mynetworks = 127.0.0.0/8

echo "Configure postfix for SMTP over TLS"
cp /etc/pki/dovecot/certs/dovecot.pem /etc/pki/tls/certs
cp /etc/pki/dovecot/private/dovecot.pem /etc/pki/tls/private

postconf -e 'smtpd_tls_cert_file=/etc/pki/tls/certs/dovecot.pem'
postconf -e 'smtpd_tls_key_file=/etc/pki/tls/private/dovecot.pem'
postconf -e 'smtpd_tls_session_cache_database = btree:/var/lib/postfix/smtpd_scache'
postconf -e 'smtpd_tls_security_level = may'
postconf -e 'smtp_tls_security_level = may'

postfix set-permissions
restorecon -vR /etc
systemctl stop postfix
systemctl start postfix
systemctl restart dovecot
```



Рисунок 3.26: Обновление скрипта автоматической настройки сервера

Аналогичные изменения вносятся в скрипт настройки клиентской машины. Добавляются команды для установки необходимых пакетов, таких как telnet и почтовый клиент Evolution, которые использовались для проверки (рис 3.27).

Автоматизация установки инструментов тестирования позволяет быстро подготовить рабочее место клиента. На скриншоте виден процесс редактирования bash скрипта в текстовом редакторе nano, завершающий цикл настройки всей инфраструктуры.

```
GNU nano 8.1                                     /vagrant/provisioning.sh
#!/bin/bash
echo "Provisioning script $0"
echo "Install needed packages"
dnf -y install postfix
dnf -y install s-nail
dnf -y install telnet
dnf -y install evolution
echo "Configure postfix"
postconf -e 'inet_protocols = ipv4'
echo "Start postfix service"
systemctl enable postfix
systemctl start postfix
```

Рисунок 3.27: Редактирование скрипта инициализации клиентской машины

4 Выводы

В ходе выполнения лабораторной работы я приобрёл практические навыки по конфигурированию SMTP-сервера в части настройки аутентификации.

Список литературы

4.1 —