

# **Отчёт по лабораторной работе №5**

**Управление системными службами**

Ришард Когенгар

# Содержание

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Цель работы</b>                       | <b>5</b>  |
| <b>2</b> | <b>Ход выполнения</b>                    | <b>6</b>  |
| 2.1      | Управление системными службами . . . . . | 6         |
| 2.2      | Конфликты юнитов systemd . . . . .       | 11        |
| 2.3      | Изолируемые цели systemd . . . . .       | 15        |
| 2.4      | Цель загрузки по умолчанию . . . . .     | 17        |
| 2.5      | Вывод . . . . .                          | 18        |
| <b>3</b> | <b>Контрольные вопросы</b>               | <b>19</b> |

# Список иллюстраций

|      |   |    |
|------|---|----|
| 2.1  | Запуск vsftpd и проверка статуса . . . . .                                | 7  |
| 2.2  | Включение автозапуска vsftpd . . . . .                                    | 7  |
| 2.3  | Отключение автозапуска vsftpd . . . . .                                   | 8  |
| 2.4  | Проверка символических ссылок и повторное включение автозапуска . . . . . | 9  |
| 2.5  | Список зависимостей юнита vsftpd . . . . .                                | 10 |
| 2.6  | Обратные зависимости юнита vsftpd . . . . .                               | 10 |
| 2.7  | Установка пакетов iptables . . . . .                                      | 11 |
| 2.8  | Статус firewalld и iptables . . . . .                                     | 12 |
| 2.9  | Содержимое firewalld.service . . . . .                                    | 13 |
| 2.10 | Содержимое iptables.service . . . . .                                     | 14 |
| 2.11 | Маскирование службы iptables . . . . .                                    | 15 |
| 2.12 | Список изолируемых целей . . . . .  | 16 |
| 2.13 | Переход в rescue.target . . . . .   | 16 |
| 2.14 | Определение цели по умолчанию . . . . .                                   | 17 |
| 2.15 | Возврат к graphical.target . . . . .                                      | 17 |

## **Список таблиц**

# 1 Цель работы

Получить навыки управления системными службами операционной системы посредством systemd.

## 2 Ход выполнения

### 2.1 Управление системными службами

1. Для выполнения операций управления системными службами получены полномочия администратора.

После выполнения входа под пользователем root стало возможным устанавливать пакеты и изменять состояние юнитов systemd.

2. Выполнена проверка состояния службы Very Secure FTP.

На данном этапе служба отсутствовала в системе либо находилась в неактивном состоянии, так как соответствующий пакет не был установлен.

3. Установлен пакет службы Very Secure FTP.

Установка завершилась успешно, после чего сервис стал доступен для управления средствами systemd.

4. Выполнен запуск службы Very Secure FTP.

После запуска демона инициирована проверка его состояния для подтверждения корректной работы.

5. Повторная проверка статуса службы показала, что сервис находится в состоянии active (running).

При этом юнит имеет состояние disabled, что означает отсутствие автозапуска при перезагрузке операционной системы.

```

Installed:
  vsftpd-3.0.5-10.el10_1.1.x86_64

Complete!
root@rishardkogengar:~# systemctl start vsftpd
root@rishardkogengar:~# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset: disabled)
   Active: active (running) since Mon 2026-01-19 14:00:25 MSK; 6s ago
  Invocation: 47da422905e74819880559ac0eb89411
    Process: 9589 ExecStart=/usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf (code=exited, status=0/SUCCESS)
   Main PID: 9590 (vsftpd)
      Tasks: 1 (limit: 23027)
     Memory: 860K (peak: 1.3M)
        CPU: 3ms
    CGroup: /system.slice/vsftpd.service
            └─9590 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Jan 19 14:00:25 rishardkogengar.localdomain systemd[1]: Starting vsftpd.service - Vsftpd ftp daemon...
Jan 19 14:00:25 rishardkogengar.localdomain systemd[1]: Started vsftpd.service - Vsftpd ftp daemon.
root@rishardkogengar:~#

```

Рис. 2.1: Запуск vsftpd и проверка статуса

- Служба Very Secure FTP добавлена в автозапуск при загрузке операционной системы.

Повторная проверка состояния подтвердила изменение статуса юнита на enabled, что свидетельствует об успешном включении автозапуска.

```

root@rishardkogengar:~#
root@rishardkogengar:~# systemctl enable vsftpd
Created symlink '/etc/systemd/system/multi-user.target.wants/vsftpd.service' → '/usr/lib/systemd/system/vsftpd.service'.
root@rishardkogengar:~# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: disabled)
   Active: active (running) since Mon 2026-01-19 14:00:25 MSK; 56s ago
  Invocation: 47da422905e74819880559ac0eb89411
    Main PID: 9590 (vsftpd)
      Tasks: 1 (limit: 23027)
     Memory: 860K (peak: 1.3M)
        CPU: 3ms
    CGroup: /system.slice/vsftpd.service
            └─9590 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Jan 19 14:00:25 rishardkogengar.localdomain systemd[1]: Starting vsftpd.service - Vsftpd ftp daemon...
Jan 19 14:00:25 rishardkogengar.localdomain systemd[1]: Started vsftpd.service - Vsftpd ftp daemon.
root@rishardkogengar:~#

```

Рис. 2.2: Включение автозапуска vsftpd

- Служба удалена из автозапуска.

После выполнения данной операции юнит снова перешёл в состояние disabled, при этом служба продолжила работу до её явной остановки.

```

root@rishardkogengar:~# systemctl disable vsftpd
Removed '/etc/systemd/system/multi-user.target.wants/vsftpd.service'.
root@rishardkogengar:~# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset: disabled)
   Active: active (running) since Mon 2026-01-19 14:00:25 MSK; 2min 55s ago
 Invocation: 47da422905e74819880559ac0eb89411
    Main PID: 9590 (vsftpd)
      Tasks: 1 (limit: 23027)
     Memory: 860K (peak: 1.3M)
        CPU: 3ms
    CGroup: /system.slice/vsftpd.service
            └─9590 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Jan 19 14:00:25 rishardkogengar.localdomain systemd[1]: Starting vsftpd.service - Vsftpd ftp daemon...
Jan 19 14:00:25 rishardkogengar.localdomain systemd[1]: Started vsftpd.service - Vsftpd ftp daemon.
root@rishardkogengar:~# █

```

Рис. 2.3: Отключение автозапуска vsftpd

8. Выведен список символических ссылок, отвечающих за запуск сервисов в target multi-user.

В результате установлено, что ссылка на vsftpd.service отсутствует, что соответствует отключённому автозапуску службы.

9. Служба Very Secure FTP повторно добавлена в автозапуск.

При повторном выводе каталога multi-user.target.wants зафиксировано появление символической ссылки на vsftpd.service, указывающей на файл юнита в каталоге /usr/lib/systemd/system.

Дополнительная проверка статуса подтвердила состояние enabled.



```

root@rishardkogengar:~#
root@rishardkogengar:~# ls /etc/systemd/system/multi-user.target.wants/
atd.service cups.path mcelog.service rsyslog.service vboxadd-service.service
auditd.service cups.service mdmonitor.service smartd.service vmtoolsd.service
audit-rules.service firewall.service ModemManager.service sshd.service
avahi-daemon.service irqbalance.service NetworkManager.service sssd.service
chronyd.service kdump.service remote-cryptsetup.target tuned.service
crond.service libstoragemgmt.service remote-fs.target vboxadd.service
root@rishardkogengar:~# systemctl enable vsftpd
Created symlink '/etc/systemd/system/multi-user.target.wants/vsftpd.service' → '/usr/lib/systemd/system/vsftpd.servic
e'.
root@rishardkogengar:~# ls /etc/systemd/system/multi-user.target.wants/
atd.service cups.path mcelog.service rsyslog.service vboxadd-service.service
auditd.service cups.service mdmonitor.service smartd.service vmtoolsd.service
audit-rules.service firewall.service ModemManager.service sshd.service vsftpd.service
avahi-daemon.service irqbalance.service NetworkManager.service sssd.service
chronyd.service kdump.service remote-cryptsetup.target tuned.service
crond.service libstoragemgmt.service remote-fs.target vboxadd.service
root@rishardkogengar:~# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: disabled)
   Active: active (running) since Mon 2026-01-19 14:00:25 MSK; 3min 52s ago
  Invocation: 47da422905e74819880559ac0eb89411
    Main PID: 9590 (vsftpd)
      Tasks: 1 (limit: 23027)
     Memory: 860K (peak: 1.3M)
        CPU: 3ms
    CGroup: /system.slice/vsftpd.service
            └─9590 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Jan 19 14:00:25 rishardkogengar.localdomain systemd[1]: Starting vsftpd.service - Vsftpd ftp daemon...
Jan 19 14:00:25 rishardkogengar.localdomain systemd[1]: Started vsftpd.service - Vsftpd ftp daemon.
root@rishardkogengar:~#

```

Рис. 2.4: Проверка символических ссылок и повторное включение автозапуска

#### 10. Выведен список зависимостей юнита vsftpd.

В отображённой иерархии показаны базовые системные цели и службы, необходимые для корректного функционирования FTP-сервиса в процессе загрузки системы.

```

root@rishardkogengar:~# systemctl list-dependencies vsftpd
vsftpd.service
● └─system.slice
● └─sysinit.target
● └─dev-hugepages.mount
● └─dev-mqueue.mount
● └─dracut-shutdown.service
○ └─fips-crypto-policy-overlay.service
○ └─iscsi-onboot.service
○ └─iscsi-starter.service
● └─kmod-static-nodes.service
○ └─ldconfig.service
● └─lvm2-lvmpolld.socket
● └─lvm2-monitor.service
○ └─multipathd.service
● └─plymouth-read-write.service
● └─plymouth-start.service
● └─proc-sys-fs-binfmt_misc.automount
○ └─selinux-autorelabel-mark.service
● └─svs-fs-fuse-connections.mount

```

Рис. 2.5: Список зависимостей юнита vsftpd

11. Выведен список юнитов, которые зависят от службы vsftpd.

В результате отображены цели multi-user.target и graphical.target, что указывает на возможность использования службы в соответствующих режимах работы системы.

```

root@rishardkogengar:~#
root@rishardkogengar:~# systemctl list-dependencies vsftpd --reverse
vsftpd.service
● └─multi-user.target
● └─graphical.target
root@rishardkogengar:~#

```

Рис. 2.6: Обратные зависимости юнита vsftpd

## 2.2 Конфликты юнитов systemd

1. Для выполнения административных операций получены полномочия администратора.

После перехода в режим суперпользователя выполнена установка пакетов, связанных с подсистемой iptables.

2. Установлены пакеты iptables и связанные с ними компоненты.

В процессе установки подтверждено наличие базовых библиотек, а также установка дополнительных пакетов, обеспечивающих работу сервисов и утилит iptables.

```
root@rishardkogengar:~# dnf -y install iptables\*
Last metadata expiration check: 0:03:15 ago on Mon 19 Jan 2026 02:03:20 PM MSK.
Package iptables-libs-1.8.11-11.el10.x86_64 is already installed.
Package iptables-nft-1.8.11-11.el10.x86_64 is already installed.
Dependencies resolved.
=====
Package                                Architecture      Version           Repository        Size
=====
Installing:
iptables-devel                         x86_64            1.8.11-11.el10    appstream         17 k
iptables-nft-services                 noarch            1.8.11-11.el10    appstream         24 k
iptables-utils                         x86_64            1.8.11-11.el10    appstream         42 k
=====
Transaction Summary
=====
Install 3 Packages

Total download size: 83 k
Installed size: 142 k
Downloading Packages:
(1/3): iptables-devel-1.8.11-11.el10.x86_64.rpm           55 kB/s | 17 kB      00:00
(2/3): iptables-nft-services-1.8.11-11.el10.noarch.rpm    60 kB/s | 24 kB      00:00
(3/3): iptables-utils-1.8.11-11.el10.x86_64.rpm          104 kB/s | 42 kB      00:00
-----
Total                                                    96 kB/s | 83 kB      00:00
Running transaction check
Transaction check succeeded.
```

Рис. 2.7: Установка пакетов iptables

3. Проверен статус служб firewalld и iptables.

Служба firewalld находится в состоянии active (running) и включена в автозапуск.

Служба iptables отображается как inactive (dead) и disabled, что указывает на её неактивное состояние и отсутствие автозапуска.

```

root@rishardkogengar:~# systemctl status firewalld.service
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; preset: enabled)
   Active: active (running) since Mon 2026-01-19 12:42:04 MSK; 1h 24min ago
 Invocation: fc3f232e12804fe79da4cf34dfc8a67d
    Docs: man:firewalld(1)
   Main PID: 1161 (firewalld)
      Tasks: 2 (limit: 23027)
  Memory: 50.8M (peak: 72.4M)
     CPU: 245ms
   CGroup: /system.slice/firewalld.service
           └─1161 /usr/bin/python3 -sP /usr/sbin/firewalld --nofork --nopid

Jan 19 12:42:03 rishardkogengar.localdomain systemd[1]: Starting firewalld.service - firewalld - dynamic firewall da
Jan 19 12:42:04 rishardkogengar.localdomain systemd[1]: Started firewalld.service - firewalld - dynamic firewall dae
root@rishardkogengar:~#
root@rishardkogengar:~# systemctl status iptables.service
○ iptables.service - IPv4 firewall with iptables
   Loaded: loaded (/usr/lib/systemd/system/iptables.service; disabled; preset: disabled)
   Active: inactive (dead)
root@rishardkogengar:~#
root@rishardkogengar:~# systemctl start firewalld
root@rishardkogengar:~# systemctl start iptables
root@rishardkogengar:~#

```

Рис. 2.8: Статус firewalld и iptables

4. Выполнена попытка одновременного запуска служб firewalld и iptables.

В результате наблюдается конфликт: при активной службе firewalld служба iptables не может быть запущена либо автоматически деактивируется, что подтверждает невозможность их параллельной работы.

5. Проанализирован файл юнита firewalld.

В конфигурации юнита явно указана директива Conflicts, включающая iptables.service, ip6tables.service, ebtables.service и ipset.service.

Это означает, что при запуске firewalld указанные сервисы не могут быть активны одновременно.

```

root@rishardkogengar:~# cat /usr/lib/systemd/system/firewalld.service
[Unit]
Description=firewalld - dynamic firewall daemon
Before=network-pre.target
Wants=network-pre.target
After=dbus.service
After=polkit.service
Conflicts=iptables.service ip6tables.service ebtables.service ipset.service
Documentation=man:firewalld(1)

[Service]
EnvironmentFile=-/etc/sysconfig/firewalld
ExecStart=/usr/sbin/firewalld --nofork --nopid $FIREWALLD_ARGS
ExecStartPost=/usr/bin/firewall-cmd --state
# don't fail ExecStartPost on RUNNING_BUT_FAILED
SuccessExitStatus=251
ExecReload=/bin/kill -HUP $MAINPID
StandardOutput=null
StandardError=null
Type=dbus
BusName=org.fedoraproject.FirewallD1
KillMode=mixed
DevicePolicy=closed
KeyringMode=private
LockPersonality=yes
MemoryDenyWriteExecute=yes
PrivateDevices=yes

```

Рис. 2.9: Содержимое firewalld.service

## 6. Проанализирован файл юнита iptables.

В описании юнита указано, что служба предназначена для настройки IPv4-фильтрации пакетов и должна запускаться до сетевых целей.

Также зафиксировано, что данный сервис относится к типу oneshot и предназначен для инициализации правил, а не для параллельной работы с firewalld.

```
root@rishardkogengar: ~  
root@rishardkogengar:~# cat /usr/lib/systemd/system/iptables.service  
[Unit]  
Description=IPv4 firewall with iptables  
AssertPathExists=/etc/sysconfig/iptables  
Before=network-pre.target  
Wants=network-pre.target  
  
[Service]  
Type=oneshot  
RemainAfterExit=yes  
ExecStart=/usr/libexec/iptables/iptables.init start  
ExecReload=/usr/libexec/iptables/iptables.init reload  
ExecStop=/usr/libexec/iptables/iptables.init stop  
Environment=BOOTUP=serial  
Environment=CONSOLETYPE=serial  
  
[Install]  
WantedBy=multi-user.target  
root@rishardkogengar:~#
```

Рис. 2.10: Содержимое iptables.service

7. Служба iptables выгружена из системы, после чего выполнен запуск службы firewalld.

Это гарантирует, что в системе активен только один механизм управления межсетевым экраном.

8. Выполнена маскировка службы iptables.

В результате была создана символическая ссылка на `/dev/null` в каталоге `/etc/systemd/system`, что полностью блокирует возможность запуска данного сервиса и предотвращает его случайную активацию.

```
root@rishardkogengar:~#  
root@rishardkogengar:~# systemctl stop iptables.service  
root@rishardkogengar:~# systemctl start firewalld.service  
root@rishardkogengar:~# systemctl mask iptables.service  
Created symlink '/etc/systemd/system/iptables.service' → '/dev/null'.  
root@rishardkogengar:~# systemctl start iptables  
Failed to start iptables.service: Unit iptables.service is masked.  
root@rishardkogengar:~# systemctl enable iptables  
Failed to enable unit: Unit /etc/systemd/system/iptables.service is masked  
root@rishardkogengar:~# █
```

Рис. 2.11: Маскирование службы iptables

9. Выполнена попытка запуска и добавления iptables в автозапуск.

Система выдала сообщение об ошибке, указывающее, что юнит замаскирован и не может быть запущен или включён в автозагрузку, что подтверждает корректность выполненной маскировки.

## 2.3 Изолируемые цели systemd

10. Выполнен переход в каталог systemd и получен список целей, которые могут быть изолированы.

Анализ файлов целей показал наличие параметра `AllowIsolate = yes`, что подтверждает возможность их изоляции и использования в качестве цели по умолчанию.

```

root@rishardkogengar: ~
root@rishardkogengar:~# cd /usr/lib/systemd/system
root@rishardkogengar:/usr/lib/systemd/system# grep Isolate *.target
ctrl-alt-del.target:AllowIsolate=yes
default.target:AllowIsolate=yes
emergency.target:AllowIsolate=yes
exit.target:AllowIsolate=yes
graphical.target:AllowIsolate=yes
halt.target:AllowIsolate=yes
initrd-switch-root.target:AllowIsolate=yes
initrd.target:AllowIsolate=yes
kexec.target:AllowIsolate=yes
multi-user.target:AllowIsolate=yes
poweroff.target:AllowIsolate=yes
reboot.target:AllowIsolate=yes
rescue.target:AllowIsolate=yes
runlevel0.target:AllowIsolate=yes
runlevel1.target:AllowIsolate=yes
runlevel2.target:AllowIsolate=yes
runlevel3.target:AllowIsolate=yes
runlevel4.target:AllowIsolate=yes
runlevel5.target:AllowIsolate=yes
runlevel6.target:AllowIsolate=yes
soft-reboot.target:AllowIsolate=yes
system-update.target:AllowIsolate=yes
root@rishardkogengar:/usr/lib/systemd/system#

```

Рис. 2.12: Список изолируемых целей

11. Операционная система переведена в режим восстановления путём изоляции цели `rescue.target`.

При переходе в данный режим выполнен вход под пользователем `root` для обслуживания системы.

```

You are in rescue mode. After logging in, type "journalctl -xb" to view
system logs, "systemctl reboot" to reboot, or "exit"
to continue bootup.
Give root password for maintenance
(or press Control-D to continue):
root@rishardkogengar:~# systemctl isolate reboot.target

```

Рис. 2.13: Переход в `rescue.target`

12. Выполнена изоляция цели `reboot.target`, в результате чего система была



корректно перезапущена с использованием механизмов systemd.

## 2.4 Цель загрузки по умолчанию

13. После загрузки системы определена цель, установленная по умолчанию.

Зафиксировано, что в качестве стандартной цели используется graphical.target.

```
rishard@rishardkogengar:~$ sudo -i
[sudo] password for rishard:
root@rishardkogengar:~# systemctl get-default
graphical.target
root@rishardkogengar:~# systemctl set-default multi-user.target
Removed '/etc/systemd/system/default.target'.
Created symlink '/etc/systemd/system/default.target' -> '/usr/lib/systemd/system/multi-user.target'.
root@rishardkogengar:~#
```

Рис. 2.14: Определение цели по умолчанию

14. Цель загрузки по умолчанию изменена на multi-user.target.

После перезагрузки системы подтверждено, что система стартует в текстовом режиме без графической оболочки.

15. Выполнена повторная смена цели загрузки по умолчанию на graphical.target.

После очередной перезагрузки система успешно загрузилась в графическом режиме, что подтверждает корректность управления целями systemd.

```
Rocky Linux 10.1 (Red Quartz)
Kernel 6.12.0-124.27.1.el10_1.x86_64 on x86_64

Web console: https://rishardkogengar.localdomain:9090/ or https://10.0.2.15:9090/

rishardkogengar login: root
Password:
Last login: Sun Jan 18 13:16:33 on pts/0
root@rishardkogengar:~# systemctl get-default
multi-user.target
root@rishardkogengar:~# systemctl set-default graphical.target
Removed '/etc/systemd/system/default.target'.
Created symlink '/etc/systemd/system/default.target' -> '/usr/lib/systemd/system/graphical.target'.
root@rishardkogengar:~#
```

Рис. 2.15: Возврат к graphical.target

## 2.5 Вывод

В ходе выполнения лабораторной работы были освоены основные приёмы управления системными службами и целями в системе `systemd`. Изучены механизмы запуска, остановки, добавления и удаления сервисов из автозапуска, а также способы анализа их текущего состояния и зависимостей. На примере служб `iptables` и `firewalld` продемонстрирован механизм разрешения конфликтов между юнитами и применение маскировки сервисов для предотвращения их запуска. Дополнительно рассмотрены изолируемые цели `systemd` и выполнена настройка цели загрузки по умолчанию, что позволило на практике закрепить навыки администрирования режимов работы операционной системы.

## 3 Контрольные вопросы

### 1. Что такое юнит (unit)? Приведите примеры.

Юнит (unit) — это базовый объект управления в системе systemd, который описывает ресурс или действие, контролируемое системой инициализации. Каждый юнит представлен конфигурационным файлом с расширением, определяющим его тип, и содержит параметры запуска, зависимости и условия активации.

Основные примеры юнитов:

`service` — сервисы и демоны (например, `vsftpd.service`, `firewalld.service`);  
`target` — цели, объединяющие группы юнитов (например, `multi-user.target`, `graphical.target`);  
`mount` — точки монтирования файловых систем;  
`socket` — сокеты для активации сервисов;  
`timer` — задания, запускаемые по расписанию;  
`device` — устройства, управляемые systemd.

### 2. Какая команда позволяет убедиться, что цель больше не входит в список автоматического запуска при загрузке системы?

Для проверки цели, установленной по умолчанию, используется команда `systemctl get-default`.

Если цель была удалена из автозапуска, вывод команды покажет другую активную цель.

Дополнительно можно проверить отсутствие символической ссылки `default.target`, указывающей на данную цель, в каталоге `/etc/systemd/system/`.

### 3. Какую команду необходимо использовать для отображения всех сервис-

### **ных юнитов, которые в настоящее время загружены?**

Для отображения всех загруженных сервисных юнитов используется команда:  
`systemctl list-units --type=service.`

Она выводит список сервисов, которые в данный момент активны, неактивны или находятся в переходном состоянии.

### **4. Как создать потребность (wants) в сервисе?**

Потребность (wants) в сервисе создаётся путём добавления соответствующей зависимости в конфигурации юнита или автоматически при включении сервиса в автозапуск.

На практике это выполняется командой:  
`systemctl enable <service>,`

которая создаёт символическую ссылку в каталоге цели (например, `multi-user.target.wants`), указывающую на файл юнита сервиса.

### **5. Как переключить текущее состояние системы на цель восстановления (rescue target)?**

Для немедленного перехода системы в режим восстановления используется команда:  
`systemctl isolate rescue.target.`

В результате будут остановлены все не относящиеся к данной цели сервисы, и система перейдёт в минимальный режим работы с доступом администратора.

### **6. Поясните причину получения сообщения о том, что цель не может быть изолирована.**

Сообщение о невозможности изоляции цели возникает в случае, если данная цель не содержит параметр `AllowIsolate=yes` в своём юнит-файле.

Это означает, что цель не предназначена для изоляции и не может быть использована для переключения текущего состояния системы, так как её активация может привести к некорректной работе или нарушению целостности системы.

### **7. Как узнать, какие юниты зависят от службы перед её отключением?**

Для определения всех юнитов, зависящих от конкретной службы, используется

команда:

```
systemctl list-dependencies <service> --reverse.
```

Она выводит список сервисов и целей, для которых данный юнит является зависимостью, что позволяет оценить последствия его отключения.