

Отчёт по лабораторной работе №5

Управление системными службами

Тукаев Тимур

Содержание

1	Цель работы	5
2	Выполнение	6
2.1	Управление сервисами	6
2.2	Конфликты юнитов (iptables и firewalld)	9
2.3	Изолируемые цели	11
2.4	Цель по умолчанию	13
3	Контрольные вопросы	14
4	Заключение	16

Список иллюстраций

2.1	Проверка статуса vsftpd	6
2.2	Статус после запуска	7
2.3	Enable/Disable vsftpd	7
2.4	Создание ссылки vsftpd	8
2.5	Зависимости vsftpd	8
2.6	Статусы firewalld и iptables	9
2.7	Юнит firewalld	10
2.8	Юнит iptables	10
2.9	Маскирование iptables	11
2.10	Поиск изолируемых целей	12
2.11	Переход в rescue.target	12
2.12	Проверка default target	13
2.13	Установка графической цели и перезагрузка	13

Список таблиц

1 Цель работы

Получить навыки управления системными службами операционной системы посредством systemd.

2 Выполнение

2.1 Управление сервисами

1. Получены полномочия администратора с помощью команды `su -`.
2. Выполнена проверка статуса службы Very Secure FTP (`vsftpd`). Поскольку пакет не был установлен, вывод показал отсутствие юнита в системе.

```
root@titukaev:/home/titukaev#  
root@titukaev:/home/titukaev# systemctl status vsftpd  
Unit vsftpd.service could not be found.  
root@titukaev:/home/titukaev# dnf -y install vsftpd  
Rocky Linux 10 - BaseOS                               14 kB/s | 4.3 kB  00:00  
Rocky Linux 10 - AppStream                             16 kB/s | 4.3 kB  00:00  
Rocky Linux 10 - Extras                               13 kB/s | 3.1 kB  00:00  
Dependencies resolved.  
-----  
Package      Architecture Version      Repository      Size  
-----  
Installing:  vsftpd      x86_64      3.0.5-10.el10 appstream       170 k
```

Рис. 2.1: Проверка статуса `vsftpd`

3. Установлен пакет Very Secure FTP с помощью менеджера пакетов DNF. После обработки зависимостей компонент `vsftpd` был успешно установлен.
4. Выполнен запуск службы Very Secure FTP. Сервис был успешно активирован.
5. Повторная проверка статуса показала, что служба находится в состоянии `active (running)`, однако не включена в автозагрузку.

```

root@titukaev:/home/titukaev#
root@titukaev:/home/titukaev# systemctl start vsftpd
root@titukaev:/home/titukaev# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset: disabled)
   Active: active (running) since Wed 2025-12-10 10:06:10 MSK; 2s ago
  Invocation: 63118dd76c744a3b9f14f62cc2fb8434
    Process: 4544 ExecStart=/usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf (code=exited, status=0/SUCCESS)
   Main PID: 4545 (vsftpd)
      Tasks: 1 (limit: 23136)
     Memory: 860K (peak: 1.2M)
          CPU: 3ms
    CGroup: /system.slice/vsftpd.service
            └─4545 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Dec 10 10:06:10 titukaev.localdomain systemd[1]: Starting vsftpd.service - Vsftpd ftp daemon...
Dec 10 10:06:10 titukaev.localdomain systemd[1]: Started vsftpd.service - Vsftpd ftp daemon.
root@titukaev:/home/titukaev#

```

Рис. 2.2: Статус после запуска

- Служба добавлена в автозапуск с помощью `systemctl enable`. Повторная проверка подтвердила изменение состояния на `enabled`. Затем служба удалена из автозагрузки, и статус вернулся к `disabled`.

```

root@titukaev:/home/titukaev# systemctl enable vsftpd
Created symlink '/etc/systemd/system/multi-user.target.wants/vsftpd.service' → '/usr/lib/systemd/system/vsftpd.service'.
root@titukaev:/home/titukaev# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: disabled)
   Active: active (running) since Wed 2025-12-10 10:06:10 MSK; 37s ago
  Invocation: 63118dd76c744a3b9f14f62cc2fb8434
    Main PID: 4545 (vsftpd)
      Tasks: 1 (limit: 23136)
     Memory: 860K (peak: 1.2M)
          CPU: 3ms
    CGroup: /system.slice/vsftpd.service
            └─4545 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Dec 10 10:06:10 titukaev.localdomain systemd[1]: Starting vsftpd.service - Vsftpd ftp daemon...
Dec 10 10:06:10 titukaev.localdomain systemd[1]: Started vsftpd.service - Vsftpd ftp daemon.
root@titukaev:/home/titukaev# systemctl disable vsftpd
Removed '/etc/systemd/system/multi-user.target.wants/vsftpd.service'.
root@titukaev:/home/titukaev# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset: disabled)
   Active: active (running) since Wed 2025-12-10 10:06:10 MSK; 51s ago
  Invocation: 63118dd76c744a3b9f14f62cc2fb8434
    Main PID: 4545 (vsftpd)
      Tasks: 1 (limit: 23136)
     Memory: 860K (peak: 1.2M)
          CPU: 3ms
    CGroup: /system.slice/vsftpd.service
            └─4545 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Dec 10 10:06:10 titukaev.localdomain systemd[1]: Starting vsftpd.service - Vsftpd ftp daemon...
Dec 10 10:06:10 titukaev.localdomain systemd[1]: Started vsftpd.service - Vsftpd ftp daemon.
root@titukaev:/home/titukaev#

```

Рис. 2.3: Enable/Disable vsftpd

- Выведен список символических ссылок, которые соответствуют юнитам, запускаемым в таргете `multi-user.target`. Было подтверждено отсутствие ссылки на `vsftpd.service`.
- Служба повторно добавлена в автозагрузку. Вывод списка ссылок показал

появление символьной ссылки на файл юнита `vsftpd.service`.

```
root@titikuae:/home/titikuae# root@titikuae# ls /etc/systemd/system/multi-user.target.wants/
atd.service          avahi-daemon.service  cups.service          httpd.service        libstoragemgmt.service  NetworkManager.service  remote-fs.target  sshd.service  vboxadd.service
audit.rules.service cron.service          firewall.service     kdump.service        mclog.service          mdeamon.service        smartd.service   tuned.service  vboxadd.service.service
audit.rules.service cron.service          firewall.service     kdump.service        mclog.service          remote-cryptsetup.target  smartd.service   tuned.service  vmtoolsd.service

root@titikuae:/home/titikuae# systemctl enable vsftpd
Created symlink /etc/systemd/system/multi-user.target.wants/vsftpd.service → /usr/lib/systemd/system/vsftpd.service.

root@titikuae:/home/titikuae# ls /etc/systemd/system/multi-user.target.wants/
atd.service          avahi-daemon.service  cups.service          httpd.service        libstoragemgmt.service  NetworkManager.service  smartd.service  vboxadd.service
audit.rules.service cron.service          cups.service          httpd.service        mclog.service          mdeamon.service        remote-cryptsetup.target  smartd.service  vboxadd.service.service
audit.rules.service cron.service          cups.service          httpd.service        mclog.service          mdeamon.service        remote-fs.target  sshd.service   tuned.service  vmtoolsd.service
avahi-daemon.service cups.service          kdump.service        ModemManager.service  remote-fs.service      tuned.service        vsftpd.service

root@titikuae:/home/titikuae# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: disabled)
   Active: active (running) since Wed 2025-12-10 10:06:10 MSK; 2min 35s ago
 Invocation: 6311b6d6c744b39f14362c2f8b434
    Main PID: 6545 (vsftpd)
      Tasks: 1 (limit: 23136)
     Memory: 860K (peak: 1.2M)
        CPU: 3ms
    CGroup: /system.slice/vsftpd.service
            └─6545 /usr/sbin/vsftpd -etc/vsftpd/vsftpd.conf

Dec 10 10:06:10 titikuae.localdomain systemd[1]: Starting vsftpd.service - Vsftpd ftp daemon...
Dec 10 10:06:10 titikuae.localdomain systemd[1]: Started vsftpd.service - Vsftpd ftp daemon.

root@titikuae:/home/titikuae#
```

Рис. 2.4: Создание ссылки vsftpd

9. Статус службы проверен повторно. Теперь юнит имеет состояние `enabled`, а служба активна.
10. Выведен список зависимостей юнита `vsftpd`. Команда отобразила дерево зависимых служб и таргетов.
11. Выведен список юнитов, которые зависят от `vsftpd`, с использованием опции `--reverse`.

```

● | selinux-audit-label-mk.service
● | sys-fs-fuse-connections.mount
● | sys-kernel-config.mount
● | sys-kernel-debug.mount
● | sys-kernel-tracing.mount
○ | systemd-ask-password-console.path
○ | systemd-binfmt.service
○ | systemd-boot-random-seed.service
○ | systemd-confext.service
○ | systemd-firstboot.service
○ | systemd-hibernate-clear.service
○ | systemd-hwdb-update.service
○ | systemd-journal-catalog-update.service
● | systemd-journal-flush.service
● | systemd-journald.service
○ | systemd-machine-id-commit.service
● | systemd-modules-load.service
● | systemd-network-generator.service
○ | systemd-pcrmachine.service
○ | systemd-pcrphase-sysinit.service
○ | systemd-pcrphase.service
○ | systemd-pstore.service
● | systemd-random-seed.service

root@titukaev:/home/titukaev# systemctl list-dependencies vsftpd --reverse
vsftpd.service
● └─ multi-user.target
● └─ graphical.target

root@titukaev:/home/titukaev#
```

Рис. 2.5: Зависимости vsftpd

2.2 Конфликты юнитов (iptables и firewalld)

1. Получены полномочия администратора и установлен пакет iptables.
2. Проверен статус служб firewalld и iptables. Первая служба оказалась активной, вторая — неактивной.

```
Upgraded:
iptables-libs-1.8.11-11.el10.x86_64
iptables-devel-1.8.11-11.el10.x86_64
iptables-nft-1.8.11-11.el10.x86_64
iptables-nft-services-1.8.11-11.el10.noarch
iptables-utls-1.8.11-11.el10.x86_64
kernel-modules-extra-matched-6.12.0-124.16.1.el10_1.x86_64

Complete!
root@titukaev:/home/titukaev# 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 Wed 2025-12-10 18:02:47 MSK; 9min ago
  Invocation: a97dd4befd404f18b9ec456532d8e29b
     Docs: man:firewalld(1)
    Main PID: 1289 (firewalld)
       Tasks: 2 (limit: 23136)
      Memory: 50M (peak: 72M)
         CPU: 474ms
    CGroup: /system.slice/firewalld.service
            └─1289 /usr/bin/python3 -sP /usr/sbin/firewalld --nofork --nopid

Dec 10 18:02:44 titukaev.localdomain systemd[1]: Starting firewalld.service - firewalld - dynamic firewall daemon...
Dec 10 18:02:47 titukaev.localdomain systemd[1]: Started firewalld.service - firewalld - dynamic firewall daemon.
root@titukaev:/home/titukaev# 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@titukaev:/home/titukaev# systemctl start firewalld
root@titukaev:/home/titukaev# systemctl start iptables
root@titukaev:/home/titukaev#
```

Рис. 2.6: Статусы firewalld и iptables

3. Выполнены попытки запуска обеих служб. Запуск одной служб послужил причиной остановки другой, что указывает на конфликт.
4. Просмотрено содержимое юнит-файла firewalld.service. В параметрах обнаружена директива Conflicts=iptables.service, фиксирующая несовместимость этих сервисов.

```

root@titukaev:/home/titukaev# 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

```

Рис. 2.7: Юнит firewalld

5. Просмотрено содержимое юнит-файла `iptables.service`, что позволило изучить особенности его запуска и конфигурации.

```

root@titukaev:/home/titukaev#
root@titukaev:/home/titukaev# 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@titukaev:/home/titukaev# █

```

Рис. 2.8: Юнит iptables

6. Служба `iptables` была остановлена, после чего без конфликтов запущена служба `firewalld`.

7. Для предотвращения случайного запуска iptables сервис был замаскирован. В каталоге /etc/systemd/system/ появилась символическая ссылка на /dev/null.

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

Рис. 2.9: Маскирование iptables

8. Попытка запуска iptables привела к сообщению об ошибке, что юнит замаскирован и не может быть активирован.
9. Аналогично завершилась попытка добавить iptables в автозапуск: система сообщила, что юнит замаскирован и не может быть включён.

2.3 Изолируемые цели

1. Получены полномочия администратора, выполнен переход в каталог /usr/lib/systemd/system, после чего выполнен поиск всех целей, поддерживающих изоляцию. В вывод попали только те .target-файлы, в которых присутствует строка AllowIsolate=yes.

```

root@titukaev:/home/titukaev#
root@titukaev:/home/titukaev# cd /usr/lib/systemd/system
root@titukaev:/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@titukaev:/usr/lib/systemd/system# systemctl isolate rescue.target

```

Рис. 2.10: Поиск изолируемых целей

- Операционная система была переключена в режим восстановления с помощью изоляции цели `rescue.target`. На консоли появилось сообщение о входе в режим `rescue` и запрос пароля суперпользователя, после чего был предоставлен доступ к окружению восстановления.

```

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@titukaev:~# systemctl isolate reboot.target _

```

Рис. 2.11: Переход в `rescue.target`

- Выполнен перезапуск операционной системы через изоляцию цели `reboot.target`. После выполнения команды система инициировала процесс перезагрузки.

2.4 Цель по умолчанию

1. Получены привилегии администратора. Выведена текущая цель, установленная по умолчанию. В данном случае система загружалась в режиме `graphical.target`.

```
titukaev@titukaev:~$ su
Password:
root@titukaev:/home/titukaev#
root@titukaev:/home/titukaev# systemctl get-default
graphical.target
root@titukaev:/home/titukaev# 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@titukaev:/home/titukaev#
```

Рис. 2.12: Проверка default target

2. По умолчанию был установлен текстовый режим загрузки путём выбора цели `multi-user.target`. После выполнения команды была перезаписана ссылка `default.target`, направленная на выбранный таргет.

После перезагрузки система действительно загрузилась в текстовом режиме.

3. Выполнено переключение обратно в графический режим. Цель по умолчанию изменена на `graphical.target`, что также привело к обновлению символической ссылки `default.target`.

После очередной перезагрузки система запустилась в графическом режиме.

```
Rocky Linux 10.0 (Red Quartz)
Kernel 6.12.0-55.39.1.el10_0.x86_64 on x86_64

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

titukaev login: root
Password:
Last login: Wed Dec 10 10:19:43 on pts/0
root@titukaev:~# systemctl get-default
multi-user.target
root@titukaev:~# 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@titukaev:~# reboot _
```

Рис. 2.13: Установка графической цели и перезагрузка

3 Контрольные вопросы

1. Юнит (unit) — это объект systemd, описывающий ресурс или действие в системе.
Примеры: `service` (сервис), `target` (цель), `mount` (точка монтирования), `timer` (таймер), `socket` (сокет).
2. Чтобы убедиться, что цель больше не включена в автозапуск, используют команду `systemctl disable`, а затем проверяют её состояние командой `systemctl status`.
3. Для отображения всех загруженных сервисных юнитов применяется команда
`systemctl --type=service`.
4. Чтобы создать потребность (wants) в сервисе, используют команду `systemctl enable`, которая создаёт символическую ссылку в каталоге `*.wants/`.
5. Для переключения системы в режим восстановления выполняют команду `systemctl isolate rescue.target`.
6. Сообщение о невозможности изоляции цели появляется, если в её юнит-файле отсутствует параметр `AllowIsolate=yes`, что делает её недоступной для изоляции.
7. Чтобы определить, какие юниты зависят от конкретной службы, используют команду

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

4 Заключение

В ходе лабораторной работы были изучены механизмы управления системными службами и целями в среде `systemd`. На практических примерах выполнены установка, запуск, остановка и настройка автозагрузки сервисов, а также разбор конфликтов между ними. Особое внимание уделено работе с изолируемыми целями, которые позволяют переключать систему в различные режимы, такие как восстановление или перезагрузка. Были рассмотрены зависимости юнитов, порядок их загрузки и способы предотвращения запуска конфликтующих служб посредством маскирования.