

Отчёт по лабораторной работе №5

Управление системными службами

Шаханеоядж Хаоладар

Содержание

1 Цель работы	5
2 Выполнение	6
2.1 Управление сервисом vsftpd	6
2.2 Конфликты юнитов: firewalld и iptables	10
2.3 Изолируемые цели	14
2.4 Цель по умолчанию	16
3 Контрольные вопросы	17
4 Заключение	19

Список иллюстраций

2.1 Попытка проверки статуса до установки	6
2.2 Установка пакета vsftpd	7
2.3 Запуск и проверка работы vsftpd	7
2.4 Изменение состояния автозапуска	8
2.5 Символические ссылки для автозапуска сервисов	8
2.6 Повторная проверка статуса vsftpd	9
2.7 Анализ зависимостей vsftpd	10
2.8 Установка iptables	11
2.9 Статус firewalld и iptables	11
2.10 Запуск firewalld и iptables	12
2.11 Маскирование iptables	14
2.12 Список изолируемых целей	15
2.13 Переход в rescue.target	15
2.14 Установка multi-user.target по умолчанию	16
2.15 Возврат к graphical.target по умолчанию	16

Список таблиц

1 Цель работы

Получить навыки управления системными службами операционной системы посредством systemd.

2 Выполнение

2.1 Управление сервисом vsftpd

- Для начала была проверена доступность службы **vsftpd** с помощью команды

systemctl status vsftpd

Система вернула сообщение о том, что сервис отсутствует.

```
root@haoladar:~# systemctl status vsftpd
Unit vsftpd.service could not be found.
root@haoladar:~# dnf -y install vsftpd
Last metadata expiration check: 0:20:56 ago on Mon 22 Sep 2025 07:14:09 PM MSK.
Dependencies resolved.
=====
 Package           Architecture      Version       Repository      Size
=====
 Installing:
  vsftpd          x86_64          3.0.5-9.el10   appstream      170 k
Transaction Summary
=====
 Install 1 Package

Total download size: 170 k
Installed size: 348 k
Downloading Packages:
vsftpd-3.0.5-9.el10.x86_64.rpm          266 kB/s | 170 kB     00:00
-----
Total                                         168 kB/s | 170 kB     00:01
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing          :                                1/1
  Installing         : vsftpd-3.0.5-9.el10.x86_64    1/1
  Running scriptlet: vsftpd-3.0.5-9.el10.x86_64    1/1
-----
Installed:
  vsftpd-3.0.5-9.el10.x86_64
```

Рис. 2.1: Попытка проверки статуса до установки

- Далее выполнена установка службы Very Secure FTP:

dnf -y install vsftpd

В процессе загрузки и установки пакет **vsftpd** был успешно добавлен в

систему.

```
root@haoladar:~# systemctl start vsftpd
root@haoladar:~# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset: disabled)
   Active: active (running) since Mon 2025-09-22 19:35:51 MSK; 4s ago
     Invocation: 6af83d96910e4830bfaad65378a92bc3
      Process: 7217 ExecStart=/usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf (code=exited, status=0/SUCCESS)
     Main PID: 7218 (vsftpd)
        Tasks: 1 (limit: 24779)
       Memory: 756K (peak: 1.2M)
          CPU: 3ms
         CGroup: /system.slice/vsftpd.service
                   └─7218 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Sep 22 19:35:51 haoladar.localdomain systemd[1]: Starting vsftpd.service - Vsftpd ftp daemon...
Sep 22 19:35:51 haoladar.localdomain systemd[1]: Started vsftpd.service - Vsftpd ftp daemon.
root@haoladar:~#
```

Рис. 2.2: Установка пакета vsftpd

3. После установки служба была запущена командой

systemctl start vsftpd

Проверка состояния показала, что сервис работает, но при перезапуске ОС автоматически не запускается.

```
root@haoladar:~# systemctl enable vsftpd
Created symlink '/etc/systemd/system/multi-user.target.wants/vsftpd.service' → '/usr/lib/systemd/system/vsftpd.service'.
root@haoladar:~# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: disabled)
   Active: active (running) since Mon 2025-09-22 19:35:51 MSK; 1min 14s ago
     Invocation: 6af83d96910e4830bfaad65378a92bc3
      Main PID: 7218 (vsftpd)
        Tasks: 1 (limit: 24779)
       Memory: 756K (peak: 1.2M)
          CPU: 3ms
         CGroup: /system.slice/vsftpd.service
                   └─7218 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Sep 22 19:35:51 haoladar.localdomain systemd[1]: Starting vsftpd.service - Vsftpd ftp daemon...
Sep 22 19:35:51 haoladar.localdomain systemd[1]: Started vsftpd.service - Vsftpd ftp daemon.
root@haoladar:~#
```

Рис. 2.3: Запуск и проверка работы vsftpd

4. Чтобы добавить сервис в автозагрузку, была использована команда

systemctl enable vsftpd

Теперь статус изменился на **enabled**. Для проверки была выполнена команда **systemctl disable vsftpd**, которая убрала службу из автозагрузки.

```

root@haoladar:~# systemctl disable vsftpd
Removed '/etc/systemd/system/multi-user.target.wants/vsftpd.service'.
root@haoladar:~# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset: disabled)
     Active: active (running) since Mon 2025-09-22 19:35:51 MSK; 1min 51s ago
   Invocation: 6af83d96910e4830bfaad65378a92bc3
      Main PID: 7218 (vsftpd)
        Tasks: 1 (limit: 24779)
       Memory: 756K (peak: 1.2M)
         CPU: 3ms
      CGroup: /system.slice/vsftpd.service
              └─7218 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Sep 22 19:35:51 haoladar.localdomain systemd[1]: Starting vsftpd.service - Vsftpd ftp daemon...
Sep 22 19:35:51 haoladar.localdomain systemd[1]: Started vsftpd.service - Vsftpd ftp daemon.
root@haoladar:~#

```

Рис. 2.4: Изменение состояния автозапуска

- Для анализа символьических ссылок, отвечающих за запуск сервисов, был выполнен вывод содержимого каталога

ls /etc/systemd/system/multi-user.target.wants/

После включения vsftpd появилась соответствующая символьная ссылка на юнит /usr/lib/systemd/system/vsftpd.service.

```

root@haoladar:~# ls /etc/systemd/system/multi-user.target.wants/
atd.service          cups.service           ModemManager.service    sssd.service
auditd.service       firewalld.service      NetworkManager.service tuned.service
audit-rules.service  irqbalance.service    remote-cryptsetup.target vboxadd.service
avahi-daemon.service kdump.service         remote-fs.target      vboxadd-service.service
chronyd.service      libstoragemgmt.service rsyslog.service       vmtoolsd.service
crond.service        mcelog.service        smartd.service        vsftpd.service
cups.path            mdmonitor.service     sshd.service
root@haoladar:~# systemctl enable vsftpd
Created symlink '/etc/systemd/system/multi-user.target.wants/vsftpd.service' → '/usr/lib/systemd/system/vsftpd.service'.
root@haoladar:~# ls /etc/systemd/system/multi-user.target.wants/
atd.service          cups.service           ModemManager.service    sssd.service
auditd.service       firewalld.service      NetworkManager.service tuned.service
audit-rules.service  irqbalance.service    remote-cryptsetup.target vboxadd.service
avahi-daemon.service kdump.service         remote-fs.target      vboxadd-service.service
chronyd.service      libstoragemgmt.service rsyslog.service       vmtoolsd.service
crond.service        mcelog.service        smartd.service        vsftpd.service
cups.path            mdmonitor.service     sshd.service
root@haoladar:~#

```

Рис. 2.5: Символические ссылки для автозапуска сервисов

- Повторная проверка статуса показала, что состояние службы изменилось с **disabled** на **enabled**, а сервис продолжает работу.

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

Рис. 2.6: Повторная проверка статуса vsftpd

7. Для анализа зависимостей была выполнена команда

systemctl list-dependencies vsftpd

Также выведен список юнитов, зависящих от данного сервиса, с помощью

systemctl list-dependencies vsftpd –reverse.

```
└─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
○  ├─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
root@haoladar:~# systemctl list-dependencies vsftpd --reverse
vsftpd.service
  └─multi-user.target
    └─graphical.target
root@haoladar:~#
```

Рис. 2.7: Анализ зависимостей vsftpd

2.2 Конфликты юнитов: firewalld и iptables

- Сначала были получены полномочия администратора и выполнена установка пакета **iptables**:

```
dnf -y install iptables*
```

```

Installed:
  iptables-devel-1.8.11-8.el10_0.x86_64           iptables-nft-services-1.8.11-8.el10_0.noarch
  iptables-utils-1.8.11-8.el10_0.x86_64

Complete!
root@haoladar:~# 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 2025-09-22 19:10:02 MSK; 34min ago
     Invocation: 5bf4a53f833841059663d3ade590c90c
   Docs: man:firewalld(1)
   Main PID: 949 (firewalld)
      Tasks: 2 (limit: 24779)
     Memory: 49.3M (peak: 51.3M)
        CPU: 173ms
      CGroup: /system.slice/firewalld.service
              └─949 /usr/bin/python3 -sP /usr/sbin/firewalld --nofork --nopid

Sep 22 19:10:02 haoladar.localdomain systemd[1]: Starting firewalld.service - firewalld - dynamic firewall >
Sep 22 19:10:02 haoladar.localdomain systemd[1]: Started firewalld.service - firewalld - dynamic firewall d>
root@haoladar:~# 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@haoladar:~#

```

Рис. 2.8: Установка iptables

2. Проверка состояния сервисов показала, что **firewalld** активен и включён в автозагрузку, а **iptables** отключён и находится в состоянии inactive:

```

root@haoladar:~# systemctl start firewalld
root@haoladar:~# systemctl start iptables
root@haoladar:~# systemctl status firewalld.service
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; preset: enabled)
   Active: inactive (dead) since Mon 2025-09-22 19:44:49 MSK; 4s ago
     Duration: 34min 46.983s
   Invocation: 5bf4a53f833841059663d3ade590c90c
   Docs: man:firewalld(1)
   Process: 949 ExecStart=/usr/sbin/firewalld --nofork --nopid $FIREWALLD_ARGS (code=exited, status=0/SUCCESS)
   Main PID: 949 (code=exited, status=0/SUCCESS)
      Mem peak: 51.3M
        CPU: 188ms

Sep 22 19:10:02 haoladar.localdomain systemd[1]: Starting firewalld.service - firewalld - dynamic firewall >
Sep 22 19:10:02 haoladar.localdomain systemd[1]: Started firewalld.service - firewalld - dynamic firewall d>
Sep 22 19:44:49 haoladar.localdomain systemd[1]: Stopping firewalld.service - firewalld - dynamic firewall >
Sep 22 19:44:49 haoladar.localdomain systemd[1]: firewalld.service: Deactivated successfully.
Sep 22 19:44:49 haoladar.localdomain systemd[1]: Stopped firewalld.service - firewalld - dynamic firewall d>
root@haoladar:~# systemctl status iptables.service
● iptables.service - IPv4 firewall with iptables
   Loaded: loaded (/usr/lib/systemd/system/iptables.service; disabled; preset: disabled)
   Active: active (exited) since Mon 2025-09-22 19:44:49 MSK; 7s ago
     Invocation: 86272cc4ec9f491489b0fdfe769240c9
   Process: 9101 ExecStart=/usr/libexec/iptables/iptables.init start (code=exited, status=0/SUCCESS)
   Main PID: 9101 (code=exited, status=0/SUCCESS)
      Mem peak: 1.6M
        CPU: 8ms

Sep 22 19:44:49 haoladar.localdomain systemd[1]: Starting iptables.service - IPv4 firewall with iptables...

```

Рис. 2.9: Статус firewalld и iptables

3. Попытка запуска обоих сервисов показала конфликт: при старте **iptables** служба **firewalld** останавливается. Аналогично, запуск **firewalld** деактиви-

пует **iptables**.

```
root@haoladar:~# 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 --nrepid $FIREWALLD_ARGS
ExecReload=/bin/kill -HUP $MAINPID
# suppress to log debug and error output also to /var/log/messages
StandardOutput=null
StandardError=null
Type=dbus
BusName=org.fedoraproject.FirewallD1
KillMode=mixed
DevicePolicy=closed
KeyringMode=private
LockPersonality=yes
MemoryDenyWriteExecute=yes
PrivateDevices=yes
ProtectClock=yes
ProtectControlGroups=yes
ProtectHome=yes
ProtectHostname=yes
ProtectKernelLogs=yes
ProtectKernelModules=no
```

Рис. 2.10: Запуск firewalld и iptables

4. Для анализа конфигурации были просмотрены юнит-файлы:

- В `/usr/lib/systemd/system/firewalld.service` явно указан параметр **Conflicts=iptables.service**, что запрещает одновременную работу двух сервисов.
- В `/usr/lib/systemd/system/iptables.service` конфликты не описаны.

```
root@haoladar:~# 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=CONSOLETEYPE=serial

[Install]
WantedBy=multi-user.target
root@haoladar:~#
root@naoladar:~#
root@haoladar:# systemctl stop iptables.service
root@haoladar:# systemctl start firewalld.service
root@haoladar:# systemctl mask iptables.service
Created symlink '/etc/systemd/system/iptables.service' → '/dev/null'.
root@haoladar:# systemctl start iptables
Failed to start iptables.service: Unit iptables.service is masked.
root@haoladar:# systemctl enable iptables
Failed to enable unit: Unit /etc/systemd/system/iptables.service is masked
root@haoladar:~#
```

5. Для исключения случайного запуска **iptables** сервис был замаскирован командой

systemctl mask iptables.

В результате создана символьная ссылка `/etc/systemd/system/iptables.service → /dev/null`, что делает невозможным его запуск.

```
root@nao.ladar:~#  
root@haoladar:~# cd /usr/lib/systemd/system  
root@haoladar:/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@haoladar:/usr/lib/systemd/system#
```

Рис. 2.11: Маскирование iptables

6. Попытка запуска замаскированного сервиса завершилась ошибкой:

Failed to start iptables.service: Unit iptables.service is masked.

Аналогично, при добавлении в автозагрузку система выдала сообщение о том, что юнит замаскирован.

2.3 Изолируемые цели

1. Получив права администратора, был выполнен переход в каталог `/usr/lib/systemd/system` и поиск целей, которые могут быть изолированы. Для этого использовалась команда

grep Isolate *.target

В списке присутствуют такие цели, как `multi-user.target`, `graphical.target`, `rescue.target`, `reboot.target`, `poweroff.target` и другие, содержащие строку `AllowIsolate=yes`.

```
root@naoladar:~#  
root@naoladar:~# cd /usr/lib/systemd/system  
root@naoladar:/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@naoladar:/usr/lib/systemd/system#
```

Рис. 2.12: Список изолируемых целей

2. Далее система была переведена в режим восстановления командой

systemctl isolate rescue.target

Для входа потребовался пароль суперпользователя.

```
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@haoladar:~# systemctl isolate reboot.target _
```

Рис. 2.13: Переход в rescue.target

3. Перезапуск системы был выполнен с помощью команды

systemctl isolate reboot.target

2.4 Цель по умолчанию

- Для начала был определён текущий режим загрузки:

systemctl get-default

По умолчанию система загружалась в **graphical.target**.

- Для перевода системы в текстовый режим по умолчанию использована команда

systemctl set-default multi-user.target

После перезагрузки ОС загрузилась в консольный режим.

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

Рис. 2.14: Установка multi-user.target по умолчанию

- Чтобы вернуть графический режим по умолчанию, была применена команда

systemctl set-default graphical.target

После перезагрузки система снова загрузилась в графическую оболочку.

```
Rocky Linux 10.0 (Red Quartz)
Kernel 6.12.0-55.12.1.el10_0.x86_64 on x86_64
Web console: https://localhost:9090/
haoladar login: root
Password:
Last login: Mon Sep 22 19:53:10 on pts/0
root@haoladar:~# systemctl get-default
multi-user.target
root@haoladar:~# 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@haoladar:~#
```

Рис. 2.15: Возврат к graphical.target по умолчанию

3 Контрольные вопросы

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

Юнит (unit) – это объект, которым управляет systemd. Он описывает ресурсы или службы системы.

Примеры:

- `sshd.service` – сервис OpenSSH
- `multi-user.target` – цель, соответствующая многопользовательскому режиму
- `home.mount` – юнит монтирования каталога

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

systemctl disable имя_юнита – удаляет юнит из автозагрузки.

Для проверки: **systemctl status имя_юнита** или просмотр каталога `/etc/systemd/system/*.wants/`.

3. Какую команду вы должны использовать для отображения всех сервисных юнитов, которые в настоящее время загружены?

systemctl list-units --type=service

4. Как создать потребность (wants) в сервисе?

systemctl enable имя_юнита – создаёт символьную ссылку в каталоге `*.wants/`, выражая зависимость *wants*.

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

systemctl isolate rescue.target

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

Это происходит, если в юнит-файле цели отсутствует строка **AllowIsolate=yes**.

Такие цели не предназначены для изоляции.

7. Вы хотите отключить службу systemd, но, прежде чем сделать это, вы хотите узнать, какие другие юниты зависят от этой службы. Какую команду вы бы использовали?

systemctl list-dependencies имя_юнита –reverse

4 Заключение

В ходе работы были изучены механизмы управления юнитами `systemd`, включая установку, запуск, автозагрузку, конфликты и изолируемые цели.