

Лабораторная работа №5. Управление системными службами

Дисциплина: Администрирование операционных систем

Жукова Арина Александровна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Управление сервисами	7
3.2	Конфликты юнитов	10
3.3	Изолируемые цели	12
3.4	Цель по умолчанию	13
4	Контрольные вопросы	16
5	Выводы	17
	Список литературы	18

Список иллюстраций

3.1	Проверка статуса, установка службы	7
3.2	Проверка статуса	7
3.3	Добавление службы в автозапуск	8
3.4	Удаление службы из автозапуска	8
3.5	Вывод символических ссылок	8
3.6	Повторный запуск службы, вывод ссылок, проверка статуса	9
3.7	Вывод зависимостей юнита	10
3.8	Вывод зависимых юнитов	10
3.9	Установка iptables	11
3.10	Проверка статусов	11
3.11	Работа команды cat	11
3.12	Выгрузка службы, блокировка запуска	12
3.13	Попытка запуск службы	12
3.14	Попытка добавления службы в автозапуск	12
3.15	Список целей для изоляции	13
3.16	Переключение операционной системы в режим восстановления .	13
3.17	Перезапуск операционной системы	13
3.18	Вывод цели установленной по умолчанию	14
3.19	Установки цели по умолчанию	14
3.20	Текстовый режим	14
3.21	Графический режим	15

Список таблиц

1 Цель работы

Получить навыки управления системными службами операционной системы посредством systemd.

2 Задание

1. Выполните основные операции по запуску (останову), определению статуса, добавлению (удалению) в автозапуск и пр. службы Very Secure FTP (раздел 5.4.1).
2. Продемонстрируйте навыки по разрешению конфликтов юнитов для служб firewalld и iptables (раздел 5.4.2).
3. Продемонстрируйте навыки работы с изолированными целями (разделы 5.4.3, 5.4.4).

3 Выполнение лабораторной работы

3.1 Управление сервисами

1. Получаем полномочия администратора. Проверяем статус службы Very Secure FTP при помощи `systemctl status vsftpd`. Вывод команды показывает, юнит `vsftpd` не установлен. Устанавливаем службу Very Secure FTP при помощи команды `dnf -y install vsftpd` (рис. 3.1).

```
[aazhukoval@aazhukoval ~]$ su -
Пароль:
[root@aazhukoval ~]# systemctl status vsftpd
Unit vsftpd.service could not be found.
[root@aazhukoval ~]# dnf -y install vsftpd
Rocky Linux 9 - BaseOS                                10 kB/s | 4.1 kB    00:00
Rocky Linux 9 - BaseOS                                2.9 MB/s | 2.3 MB    00:00
Rocky Linux 9 - AppStream                              9.3 kB/s | 4.5 kB    00:00
Rocky Linux 9 - AppStream                              7.3 MB/s | 8.0 MB    00:01
Rocky Linux 9 - Extras                                7.6 kB/s | 2.9 kB    00:00
Зависимости разрешены.
```

Рис. 3.1: Проверка статуса, установка службы

2. Запускаем службу Very Secure FTP (`systemctl start vsftpd`). Проверяем статус службы Very Secure FTP (рис. 3.2).

```
[root@aazhukoval ~]# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset: disabled)
   Active: active (running) since Thu 2024-10-03 10:40:27 MSK; 40s ago
     Process: 3277 ExecStart=/usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf (code=exited, status=0/SUCCESS)
    Main PID: 3278 (vsftpd)
       Tasks: 1 (limit: 23036)
      Memory: 720.0K
         CPU: 6ms
    CGroup: /system.slice/vsftpd.service
            └─3278 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

окт 03 10:40:27 aazhukoval.localdomain systemd[1]: Starting Vsftpd ftp daemon...
окт 03 10:40:27 aazhukoval.localdomain systemd[1]: Started Vsftpd ftp daemon.
```

Рис. 3.2: Проверка статуса

Вывод показывает, что служба в настоящее время работает, но не будет активирована при перезапуске операционной системы.

3. Добавляем службу Very Secure FTP в автозапуск при загрузке операционной системы, используя команду `systemctl enable`, проверяем статус службы (рис. 3.3).

```
[root@aazhukoval ~]# systemctl enable vsftpd
Created symlink /etc/systemd/system/multi-user.target.wants/vsftpd.service → /usr/lib/systemd/system/vsftpd.service.
[root@aazhukoval ~]# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: disabled)
   Active: active (running) since Thu 2024-10-03 10:40:27 MSK; 2min 42s ago
     Main PID: 3278 (vsftpd)
       Tasks: 1 (limit: 23036)
      Memory: 720.0K
         CPU: 6ms
    CGroup: /system.slice/vsftpd.service
            └─3278 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

окт 03 10:40:27 aazhukoval.localdomain systemd[1]: Starting Vsftpd ftp daemon...
окт 03 10:40:27 aazhukoval.localdomain systemd[1]: Started Vsftpd ftp daemon.
```

Рис. 3.3: Добавление службы в автозапуск

Удаляем службу из автозапуска, используя команду `systemctl disable`, и снова проверяем её статус (рис. 3.4).

```
[root@aazhukoval ~]# systemctl disable vsftpd
Removed "/etc/systemd/system/multi-user.target.wants/vsftpd.service".
[root@aazhukoval ~]# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset: disabled)
   Active: active (running) since Thu 2024-10-03 10:40:27 MSK; 4min 8s ago
     Main PID: 3278 (vsftpd)
       Tasks: 1 (limit: 23036)
      Memory: 720.0K
         CPU: 6ms
    CGroup: /system.slice/vsftpd.service
            └─3278 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

окт 03 10:40:27 aazhukoval.localdomain systemd[1]: Starting Vsftpd ftp daemon...
окт 03 10:40:27 aazhukoval.localdomain systemd[1]: Started Vsftpd ftp daemon.
```

Рис. 3.4: Удаление службы из автозапуска

4. Выводим на экран символические ссылки, ответственные за запуск различных сервисов (рис. 3.5).

```
[root@aazhukoval ~]# ls /etc/systemd/system/multi-user.target.wants
atd.service      cups.path        libstoragemgmt.service  remote-fs.target  tuned.service
auditd.service   cups.service     mcelog.service          rsyslog.service   vboxadd.service
avahi-daemon.service  firewallld.service  mdmonitor.service       smartd.service    vboxadd-service.service
chronyd.service    irqbalance.service  ModemManager.service    sshd.service      vmtoolsd.service
crond.service      kdump.service     NetworkManager.service  sssd.service
```

Рис. 3.5: Вывод символических ссылок

Ссылки на `vsftpd.service` не существует.

5. Снова добавляем службу Very Secure FTP в автозапуск, выводим на экран символические ссылки, ответственные за запуск сервисов, проверяем статус службы (рис. 3.6).

```
[root@aazhukoval ~]# systemctl enable vsftpd
Created symlink /etc/systemd/system/multi-user.target.wants/vsftpd.service → /usr/lib/systemd/system/vsftpd.service.
[root@aazhukoval ~]# ls /etc/systemd/system/multi-user.target.wants
atd.service      cups.path        libstoragemgmt.service  remote-fs.target  tuned.service
auditd.service   cups.service     mcelog.service          rsyslog.service   vboxadd.service
avahi-daemon.service  firewallld.service  mdmonitor.service       smartd.service    vboxadd-service.service
chronyd.service    irqbalance.service  ModemManager.service    sshd.service       vmttoolsd.service
crond.service      kdump.service      NetworkManager.service  sssd.service       vsftpd.service
```

```
[root@aazhukoval ~]# systemctl status vsftpd
* vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: disabled)
   Active: active (running) since Thu 2024-10-03 10:40:27 MSK; 6min ago
     Main PID: 3278 (vsftpd)
       Tasks: 1 (limit: 23036)
      Memory: 720.0K
         CPU: 6ms
        CGroup: /system.slice/vsftpd.service
                └─3278 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

окт 03 10:40:27 aazhukoval.localdomain systemd[1]: Starting Vsftpd ftp daemon...
```

Рис. 3.6: Повторный запуск службы, вывод ссылок, проверка статуса

Вывод команды показывает, что создана символическая ссылка для файла `/usr/lib/systemd/system/vsftpd.service` в каталоге `/etc/systemd/system/multi-user.target.wants`. Для файла юнита состояние изменено с `disabled` на `enabled`.

6. Выводим на экран список зависимостей юнита `systemctl list-dependencies vsftpd` (рис. 3.7).

```
[root@aazhukoval ~]# systemctl list-dependencies vsftpd
vsftpd.service
├─system.slice
├─sysinit.target
├─dev-hugepages.mount
├─dev-mqueue.mount
├─dracut-shutdown.service
├─iscsi-onboot.service
├─iscsi-starter.service
├─kmod-static-nodes.service
├─ldconfig.service
├─lvm2-lvmpolld.socket
├─lvm2-monitor.service
├─multipathd.service
├─nis-domainname.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-boot-update.service
├─systemd-firstboot.service
├─systemd-hwdb-update.service
├─systemd-journal-catalog-update.service
├─systemd-journal-flush.service
lines 1-30...skipping...
```

Рис. 3.7: Вывод зависимостей юнита

7. Выводим на экран список юнитов, которые зависят от данного юнита
`systemctl list-dependencies vsftpd --reverse` (рис. 3.8).

```
[root@aazhukoval ~]# systemctl list-dependencies vsftpd --reverse
vsftpd.service
├─multi-user.target
├─graphical.target
[root@aazhukoval ~]#
```

Рис. 3.8: Вывод зависимых юнитов

3.2 Конфликты юнитов

1. Получаем полномочия администратора. Устанавливаем iptables (рис. 3.9).

```
[root@aazhukoval ~]# dnf -y install iptables*
Последняя проверка окончания срока действия метаданных: 0:09:23 назад, Чт 03 окт 2024 10:39:46.
Пакет iptables-libs-1.8.10-4.el9_4.x86_64 уже установлен.
Пакет iptables-nft-1.8.10-4.el9_4.x86_64 уже установлен.
Зависимости разрешены.
=====
Пакет                Архитектура  Версия      Репозиторий    Размер
=====
Установка:
iptables-devel        x86_64      1.8.10-4.el9_4  appstream      14 k
iptables-nft-services noarch     1.8.10-4.el9_4  appstream      17 k
iptables-utils        x86_64      1.8.10-4.el9_4  baseos         40 k
=====
Результат транзакции
=====
Установка 3 Пакета
```

Рис. 3.9: Установка iptables

2. Проверяем статус firewalld и iptables (рис. 3.10).

```
[root@aazhukoval ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; preset: enabled)
   Active: active (running) since Thu 2024-10-03 10:35:28 MSK; 14min ago
     Docs: man:firewalld(1)
   Main PID: 785 (firewalld)
    Tasks: 2 (limit: 23036)
   Memory: 43.2M
      CPU: 884ms
   CGroup: /system.slice/firewalld.service
           └─785 /usr/bin/python3 -s /usr/sbin/firewalld --nofork --nopid

окт 03 10:35:27 aazhukoval.localdomain systemd[1]: Starting firewalld - dynamic firewall daemon...
окт 03 10:35:28 aazhukoval.localdomain systemd[1]: Started firewalld - dynamic firewall daemon.
[root@aazhukoval ~]# systemctl status iptables
○ iptables.service - IPv4 firewall with iptables
   Loaded: loaded (/usr/lib/systemd/system/iptables.service; disabled; preset: disabled)
   Active: inactive (dead)
```

Рис. 3.10: Проверка статусов

3. Пытаемся запустить firewalld и iptables и видим, что при запуске одной службы вторая деактивируется или не запускается, вводим cat /usr/lib/systemd/system/firewalld.service (рис. 3.11).

```
[root@aazhukoval ~]# 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 nftables.service
Documentation=man:firewalld(1)

[Service]
EnvironmentFile=/etc/sysconfig/firewalld
ExecStart=/usr/sbin/firewalld --nofork --nopid $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

[Install]
WantedBy=multi-user.target
Alias=dbus-org.fedoraproject.FirewallD1.service
[root@aazhukoval ~]#
```

Рис. 3.11: Работа команды cat

4. Выгружаем службу iptables, загружаем службу firewalld, заблокируем запуск iptables, проверяем создание символической ссылки на /dev/null для /etc/systemd/system/iptables.service (рис. 3.12).

```
[root@aaazhukoval ~]# systemctl stop iptables
[root@aaazhukoval ~]# systemctl start firewalld
[root@aaazhukoval ~]# systemctl mask iptables
Created symlink /etc/systemd/system/iptables.service → /dev/null.
[root@aaazhukoval ~]# ls /etc/systemd/system/iptables.service
/etc/systemd/system/iptables.service
```

Рис. 3.12: Выгрузка службы, блокировка запуска

5. Пытаемся запустить iptables (рис. 3.13).

```
[root@aaazhukoval ~]# ls /etc/systemd/system/iptables.service
/etc/systemd/system/iptables.service
[root@aaazhukoval ~]# systemctl start iptables
Failed to start iptables.service: Unit iptables.service is masked.
[root@aaazhukoval ~]#
```

Рис. 3.13: Попытка запуск службы

Появляется сообщение об ошибке, указывающее, что служба замаскирована и по этой причине не может быть запущена.

6. Пытаемся добавить iptables в автозапуск (рис. 3.14).

```
[root@aaazhukoval ~]# systemctl enable iptables
Failed to enable unit: Unit file /etc/systemd/system/iptables.service is masked.
[root@aaazhukoval ~]#
```

Рис. 3.14: Попытка добавления службы в автозапуск

Сервис неактивен, а статус загрузки замаскированный.

3.3 Изолируемые цели

1. Получаем полномочия администратора. Переходим в каталог systemd и находим список всех целей, которые можно изолировать (рис. 3.15).

```
[root@aazhukoval ~]# cd /usr/lib/systemd/system
[root@aazhukoval 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
system-update.target:AllowIsolate=yes
[root@aazhukoval system]#
```

Рис. 3.15: Список целей для изоляции

2. Переключаем операционную систему в режим восстановления (рис. 3.16).

```
[root@aazhukoval system]# systemctl isolate rescue.target
```

Рис. 3.16: Переключение операционной системы в режим восстановления

При этом необходимо ввести пароль root на консоли сервера для входа в систему.

3. Перезапускаем операционную систему (рис. 3.17).

```
You are in rescue mode. After logging in, type "journalctl -xb" to view
system logs, "systemctl reboot" to reboot, "systemctl default" or "exit"
to boot into default mode.
*** ***** root
(*** ***** Control-D *** *****):
[root@aazhukoval ~]# systemctl isolate reboot.target_
```

Рис. 3.17: Перезапуск операционной системы

3.4 Цель по умолчанию

1. Получаем полномочия администратора. Выводим на экран цель, установленную по умолчанию (рис. 3.18).

```
[aazhukoval@aazhukoval ~]$ su -
Пароль:
[root@aazhukoval ~]# systemctl get-default
graphical.target
[root@aazhukoval ~]# systemctl set-default
```

Рис. 3.18: Вывод цели установленной по умолчанию

2. Для установки цели по умолчанию используется команда `systemctl set-default`. Ставим по умолчанию запуск в текстовом режиме и перезагружаем машину (рис. 3.19).

```
[root@aazhukoval ~]# 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@aazhukoval ~]# reboot
```

Рис. 3.19: Установки цели по умолчанию

Система загрузилась в текстовом режиме. Получите полномочия администратора. Для запуска по умолчанию графического режима вводим `systemctl set-default graphical.target`. Вновь перезагружаем систему командой `reboot` (рис. 3.20).

```
Rocky Linux 9.4 (Blue Onyx)
Kernel 5.14.0-427.35.1.el9_4.x86_64 on an x86_64

Activate the web console with: systemctl enable --now cockpit.socket

aazhukoval login: root
Password:
Last login: Thu Oct 3 11:07:56 on pts/0
[root@aazhukoval ~]# 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@aazhukoval ~]# reboot
```

Рис. 3.20: Текстовый режим

Проверяем, что система загрузилась в графическом режиме (рис. 3.21).

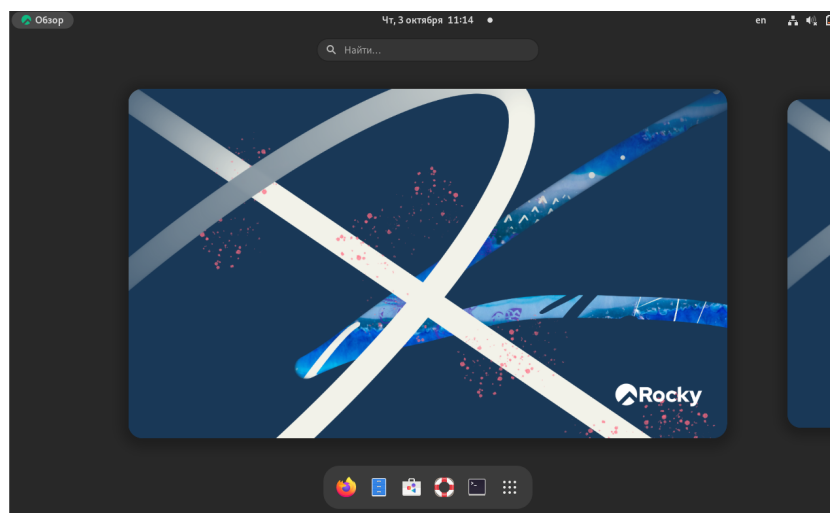


Рис. 3.21: Графический режим

4 Контрольные вопросы

1. Юнит (unit) - описание единицы программного обеспечения, которую systemd может управлять (сервис, устройство, точка монтирования).
Примеры: vsftpd.service, network-online.target, swap.swap.
2. Проверка автоматического запуска: `systemctl disable [имя_цели]`.
3. Список загруженных сервисов: `systemctl --type=service`.
4. Создание потребности: `systemctl enable [имя_сервиса]`.
5. Переключение на rescue.target: `systemctl isolate rescue.target`.
6. Причина невозможности изоляции: Цель, вероятно, не предназначена для изоляции (нет AllowIsolate=yes) или уже находится в изоляции.
7. Проверка зависимостей: `systemctl list-dependencies [имя_сервиса] --reverse`.

5 Выводы

Мы получили навыки управления системными службами операционной системы посредством systemd.

Список литературы

1. Поттеринг Л. Systemd для администраторов: цикл статей. — 2010. — URL: <http://wiki.opennet.ru/Systemd>.
2. Neil N. J. Learning CentOS: A Beginners Guide to Learning Linux. — CreateSpace Independent Publishing Platform, 2016.
3. Systemd. — 2022. — URL: <https://wiki.archlinux.org/title/Systemd>.