

Лабораторная работа №5

Дисциплина: Основы администрирования операционных систем

Жибицкая Евгения Дмитриевна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Ответы на контрольные вопросы	17
4	Выводы	20
	Список литературы	21

Список иллюстраций

2.1	Статус и установка службы Very Secure FTP	6
2.2	Запуск и статус службы Very Secure FTP	7
2.3	Добавление автозапуска	7
2.4	Отключение автозапуска	8
2.5	Символические ссылки	8
2.6	Просмотр статуса	9
2.7	Список зависимостей юнита	9
2.8	Список юнитов, которые зависят от данного юнита	9
2.9	Установка iptables	10
2.10	Статусы firewalld и iptables	11
2.11	Запуск конфликтующих служб	11
2.12	cat /usr/lib/systemd/system/firewalld.service	12
2.13	cat /usr/lib/systemd/system/iptables.service	12
2.14	Работа с iptables и firewalld	13
2.15	Списки целей	13
2.16	Изолируемые цели	14
2.17	Режим восстановления и перезапуск	14
2.18	Цель по умолчанию	15
2.19	Возвращение графического режима	15
2.20	Проверка	16
3.1	Пример	17
3.2	Пример	18
3.3	Пример	19

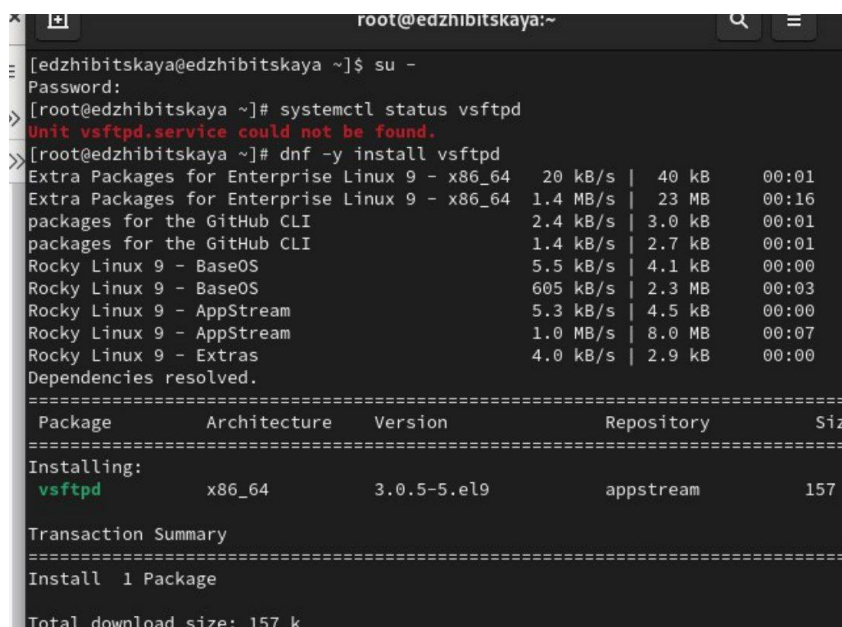
Список таблиц

1 Цель работы

Продолжение изучения ОС Linux. Знакомство и получение навыков управления системными службами операционной системы посредством systemd.

2 Выполнение лабораторной работы

Ознакомимся с различной информацией по теме, затем перейдем в режим суперпользователя, командой `systemctl status vsftpd` проверим статус службы Very Secure FTP. Установим ее(рис. 2.1).



```
root@edzhibitskaya:~  
[edzhibitskaya@edzhibitskaya ~]$ su -  
Password:  
[root@edzhibitskaya ~]# systemctl status vsftpd  
Unit vsftpd.service could not be found.  
[root@edzhibitskaya ~]# dnf -y install vsftpd  
Extra Packages for Enterprise Linux 9 - x86_64 20 kB/s | 40 kB 00:01  
Extra Packages for Enterprise Linux 9 - x86_64 1.4 MB/s | 23 MB 00:16  
packages for the GitHub CLI 2.4 kB/s | 3.0 kB 00:01  
packages for the GitHub CLI 1.4 kB/s | 2.7 kB 00:01  
Rocky Linux 9 - BaseOS 5.5 kB/s | 4.1 kB 00:00  
Rocky Linux 9 - BaseOS 605 kB/s | 2.3 MB 00:03  
Rocky Linux 9 - AppStream 5.3 kB/s | 4.5 kB 00:00  
Rocky Linux 9 - AppStream 1.0 MB/s | 8.0 MB 00:07  
Rocky Linux 9 - Extras 4.0 kB/s | 2.9 kB 00:00  
Dependencies resolved.  
=====
```

Package	Architecture	Version	Repository	Size
Installing:				
vsftpd	x86_64	3.0.5-5.el9	appstream	157

```
=====
```

Transaction Summary	
Install	1 Package

```
=====
```

Total download size: 157 k

Рис. 2.1: Статус и установка службы Very Secure FTP

Запустим службу и также проверим ее статус(рис. 2.2).

```
19 20:38 en
root@edzhibitskaya:~
[root@edzhibitskaya ~]# systemctl start vsftpd
[root@edzhibitskaya ~]# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset: disabled)
   Active: active (running) since Thu 2024-09-19 20:37:43 MSK; 7s ago
     Process: 4064 ExecStart=/usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf (code=exit)
   Main PID: 4065 (vsftpd)
      Tasks: 1 (limit: 12194)
     Memory: 708.0K
        CPU: 7ms
    CGroup: /system.slice/vsftpd.service
            └─4065 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Sep 19 20:37:43 edzhibitskaya systemd[1]: Starting Vsftpd ftp daemon...
Sep 19 20:37:43 edzhibitskaya systemd[1]: Started Vsftpd ftp daemon.
lines 1-13/13 (END)
```

Рис. 2.2: Запуск и статус службы Very Secure FTP

Командой enable добавим службу Very Secure FTP в автозапуск при загрузке ОС, проверим статус(рис. 2.3).

```
[root@edzhibitskaya ~]# systemctl enable vsftpd
Created symlink /etc/systemd/system/multi-user.target.wants/vsftpd.service → /usr/lib/systemd/system/vsftpd.service.
[root@edzhibitskaya ~]# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: disabled)
   Active: active (running) since Thu 2024-09-19 20:37:43 MSK; 4min 59s ago
     Main PID: 4065 (vsftpd)
      Tasks: 1 (limit: 12194)
     Memory: 708.0K
        CPU: 7ms
    CGroup: /system.slice/vsftpd.service
            └─4065 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Sep 19 20:37:43 edzhibitskaya systemd[1]: Starting Vsftpd ftp daemon...
Sep 19 20:37:43 edzhibitskaya systemd[1]: Started Vsftpd ftp daemon.
...skipping...
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: disabled)
   Active: active (running) since Thu 2024-09-19 20:37:43 MSK; 4min 59s ago
     Main PID: 4065 (vsftpd)
      Tasks: 1 (limit: 12194)
     Memory: 708.0K
        CPU: 7ms
    CGroup: /system.slice/vsftpd.service
            └─4065 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Sep 19 20:37:43 edzhibitskaya systemd[1]: Starting Vsftpd ftp daemon...
Sep 19 20:37:43 edzhibitskaya systemd[1]: Started Vsftpd ftp daemon.
~
```

Рис. 2.3: Добавление автозапуска

Удалим автозапуск при загрузке системы, также проверим статус(рис. 2.4).

```
[root@edzhibitskaya ~]# systemctl disable vsftpd
Removed "/etc/systemd/system/multi-user.target.wants/vsftpd.service".
[root@edzhibitskaya ~]# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset:
   Active: active (running) since Thu 2024-09-19 20:37:43 MSK; 6min ago
   Main PID: 4065 (vsftpd)
     Tasks: 1 (limit: 12194)
    Memory: 708.0K
       CPU: 7ms
   CGroup: /system.slice/vsftpd.service
           └─4065 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Sep 19 20:37:43 edzhibitskaya systemd[1]: Starting Vsftpd ftp daemon...
Sep 19 20:37:43 edzhibitskaya systemd[1]: Started Vsftpd ftp daemon.
lines 1-12/12 (END)
```

Рис. 2.4: Отключение автозапуска

Выведем на экран символические ссылки, ответственные за запуск различных сервисов - увидим, что ссылка на `vsftpd.service` не существует. Снова добавим службу Very Secure FTP в автозапуск: `systemctl enable vsftpd` и выведем на экран символические ссылки, ответственные за запуск различных сервисов. Увидим, что создана символическая ссылка для файла `/usr/lib/systemd/system/vsftpd.service` в каталоге `/etc/systemd/system/multi-user.target.wants`. (рис. 2.5).

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

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

Рис. 2.5: Символические ссылки

Опять проверим статус. Увидим, что для файла юнита состояние изменено

(рис. 2.6).

```
[root@edzhbitskaya ~]# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: d
   Active: active (running) since Thu 2024-09-19 20:37:43 MSK; 10min ago
   Main PID: 4065 (vsftpd)
     Tasks: 1 (limit: 12194)
    Memory: 708.0K
       CPU: 7ms
    CGroup: /system.slice/vsftpd.service
           └─4065 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Sep 19 20:37:43 edzhbitskaya systemd[1]: Starting Vsftpd ftp daemon...
Sep 19 20:37:43 edzhbitskaya systemd[1]: Started Vsftpd ftp daemon.
lines 1-12/12 (END)
```

Рис. 2.6: Просмотр статуса

Также, выведем список зависимостей юнита - `systemctl list-dependencies vsftpd` и список юнитов, которые зависят от данного юнита - `systemctl list-dependencies vsftpd --reverse`(рис. 2.7) и (рис. 2.8).

```
>> Sep 19 20:37:43 edzhbitskaya systemd[1]: Starting Vsftpd ftp daemon...
>> Sep 19 20:37:43 edzhbitskaya systemd[1]: Started Vsftpd ftp daemon.
>> [root@edzhbitskaya ~]# 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
```

Рис. 2.7: Список зависимостей юнита

```
├─systemd-repart.service
├─systemd-sysctl.service
├─systemd-sysusers.service
[root@edzhbitskaya ~]# systemctl list-dependencies vsftpd --reverse
vsftpd.service
├─multi-user.target
└─graphical.target
[root@edzhbitskaya ~]#
```

Рис. 2.8: Список юнитов, которые зависят от данного юнита

Перейдем к изучению конфликтов юнитов. Получим необходимые полномочия и установим iptables(рис. 2.9).

```
Total download size: 153 k
Installed size: 314 k
Downloading Packages:
(1/5): iptables-legacy-libs-1.8.10-2.2.el9.x86_ 88 kB/s | 37 kB 00:00
(2/5): iptables-legacy-devel-1.8.10-2.2.el9.x86 27 kB/s | 13 kB 00:00
(3/5): iptables-legacy-1.8.10-2.2.el9.x86_64.rp 94 kB/s | 49 kB 00:00
(4/5): iptables-services-1.8.10-2.2.el9.noarch. 102 kB/s | 15 kB 00:00
(5/5): iptables-utils-1.8.10-4.el9_4.x86_64.rpm 87 kB/s | 40 kB 00:00
-----
Total                                          58 kB/s | 153 kB 00:02
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :                                1/1
  Installing     : iptables-legacy-libs-1.8.10-2.2.el9.x86_64 1/5
  Installing     : iptables-legacy-1.8.10-2.2.el9.x86_64 2/5
  Running scriptlet: iptables-legacy-1.8.10-2.2.el9.x86_64 2/5
  Installing     : iptables-services-1.8.10-2.2.el9.noarch 3/5
  Running scriptlet: iptables-services-1.8.10-2.2.el9.noarch 3/5
  Installing     : iptables-utils-1.8.10-4.el9_4.x86_64 4/5
  Installing     : iptables-legacy-devel-1.8.10-2.2.el9.x86_64 5/5
  Running scriptlet: iptables-legacy-devel-1.8.10-2.2.el9.x86_64 5/5
  Verifying      : iptables-legacy-1.8.10-2.2.el9.x86_64 1/5
  Verifying      : iptables-legacy-devel-1.8.10-2.2.el9.x86_64 2/5
  Verifying      : iptables-legacy-libs-1.8.10-2.2.el9.x86_64 3/5
  Verifying      : iptables-services-1.8.10-2.2.el9.noarch 4/5
  Verifying      : iptables-utils-1.8.10-4.el9_4.x86_64 5/5

Installed:
  iptables-legacy-1.8.10-2.2.el9.x86_64
  iptables-legacy-devel-1.8.10-2.2.el9.x86_64
  iptables-legacy-libs-1.8.10-2.2.el9.x86_64
  iptables-services-1.8.10-2.2.el9.noarch
  iptables-utils-1.8.10-4.el9_4.x86_64
Skipped:
  iptables-libs-1.8.10-4.el9_4.x86_64

Complete!
[root@edzhibitskaya ~]#
```

Рис. 2.9: Установка iptables

После скачивания, проверим статусы firewalld и iptables(рис. 2.10).

```
root@edzhibitskaya ~]# systemctl status firewalld
firewalld.service - firewalld - dynamic firewall daemon
Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; preset>
Active: active (running) since Thu 2024-09-19 20:30:05 MSK; 23min ago
Docs: man:firewalld(1)
Main PID: 771 (firewalld)
Tasks: 2 (limit: 12194)
Memory: 9.0M
CPU: 1.150s
CGroup: /system.slice/firewalld.service
└─771 /usr/bin/python3 -s /usr/sbin/firewalld --nofork --nopid

Sep 19 20:30:04 edzhibitskaya systemd[1]: Starting firewalld - dynamic firewall>
Sep 19 20:30:05 edzhibitskaya systemd[1]: Started firewalld - dynamic firewall>
root@edzhibitskaya ~]# systemctl status iptables
iptables.service - IPv4 firewall with iptables
Loaded: loaded (/usr/lib/systemd/system/iptables.service; disabled; preset>
Active: inactive (dead)
lines 1-3/3 (END)
```

Рис. 2.10: Статусы firewalld и iptables

Попробуем запустить их. Увидим, что сделать одновременно это невозможно - одна из служб дезактивируется при запуске второй(рис. 2.11).

```
[root@edzhibitskaya ~]# systemctl start firewalld
[root@edzhibitskaya ~]# systemctl start iptables
[root@edzhibitskaya ~]# systemctl status iptables
● iptables.service - IPv4 firewall with iptables
   Loaded: loaded (/usr/lib/systemd/system/iptables.service; disabled; preset>
   Active: active (exited) since Thu 2024-09-19 20:56:15 MSK; 1min 17s ago
   Process: 4758 ExecStart=/usr/libexec/iptables/iptables.init start (code=exi>
   Main PID: 4758 (code=exited, status=0/SUCCESS)
      CPU: 39ms

Sep 19 20:56:15 edzhibitskaya systemd[1]: Starting IPv4 firewall with iptables.
Sep 19 20:56:15 edzhibitskaya iptables.init[4758]: iptables: Applying firewall
Sep 19 20:56:15 edzhibitskaya systemd[1]: Finished IPv4 firewall with iptables.
[root@edzhibitskaya ~]# systemctl status iptables
● iptables.service - IPv4 firewall with iptables
   Loaded: loaded (/usr/lib/systemd/system/iptables.service; disabled; preset>
   Active: active (exited) since Thu 2024-09-19 20:56:15 MSK; 1min 34s ago
   Process: 4758 ExecStart=/usr/libexec/iptables/iptables.init start (code=exi>
   Main PID: 4758 (code=exited, status=0/SUCCESS)
      CPU: 39ms

Sep 19 20:56:15 edzhibitskaya systemd[1]: Starting IPv4 firewall with iptables.
Sep 19 20:56:15 edzhibitskaya iptables.init[4758]: iptables: Applying firewall
Sep 19 20:56:15 edzhibitskaya systemd[1]: Finished IPv4 firewall with iptables.
lines 1-10/10 (END)
```

Рис. 2.11: Запуск конфликтующих служб

Используем команды `cat /usr/lib/systemd/system/firewalld.service` и `cat /usr/lib/systemd/system/iptables.service`. При выводе результата видим, что iptables является конфликтом для firewalld(рис. 2.12) и (рис. 2.13).

```
root@edzhbitskaya:~  
[root@edzhbitskaya ~]# 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@edzhbitskaya ~]#
```

Рис. 2.12: cat /usr/lib/systemd/system/firewalld.service

```
[root@edzhbitskaya ~]# 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@edzhbitskaya ~]#
```

Рис. 2.13: cat /usr/lib/systemd/system/iptables.service

Выгрузим службу iptables, запустим firewalld, заблокируем запуск iptables (создана символическая ссылка на /dev/null для /etc/systemd/system/iptables.service) и попробуем запустить. Также попробуем добавить службу в автозапуск(рис. 2.14).

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

Рис. 2.14: Работа с iptables и firewalld

Изучим изолируемые цели. Получим список всех активных целей и просто всех целей(рис. 2.15).

```
[root@edzhibitskaya ~]# systemctl --type=target
```

UNIT	LOAD	ACTIVE	SUB	DESCRIPTION
basic.target	loaded	active	active	Basic System
cryptsetup.target	loaded	active	active	Local Encrypted Volumes
getty.target	loaded	active	active	Login Prompts
graphical.target	loaded	active	active	Graphical Interface
integritysetup.target	loaded	active	active	Local Integrity Protected Volumes
local-fs-pre.target	loaded	active	active	Preparation for Local File Systems
local-fs.target	loaded	active	active	Local File Systems
multi-user.target	loaded	active	active	Multi-User System
network-online.target	loaded	active	active	Network is Online
network-pre.target	loaded	active	active	Preparation for Network
network.target	loaded	active	active	Network
nss-user-lookup.target	loaded	active	active	User and Group Name Lookups
paths.target	loaded	active	active	Path Units
remote-fs.target	loaded	active	active	Remote File Systems
slices.target	loaded	active	active	Slice Units
sockets.target	loaded	active	active	Socket Units
sound.target	loaded	active	active	Sound Card
sshd-keygen.target	loaded	active	active	sshd-keygen.target
swap.target	loaded	active	active	Swaps
sysinit.target	loaded	active	active	System Initialization
timers.target	loaded	active	active	Timer Units
veritysetup.target	loaded	active	active	Local Verity Protected Volumes

```
LOAD = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB = The low-level unit activation state, values depend on unit type.
22 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.
[root@edzhibitskaya ~]# systemctl --type=target --all
```

UNIT
basic.target
blockdev@dev-disk-by\x2duuid-1d21a81d\x2d1079\x2d4692\x2d933a\x2df9b20806c
blockdev@dev-dm\x2d1.target
blockdev@dev-mapper-rl_10\x2droot.target
blockdev@dev-mapper-rl_10\x2dswap.target
blockdev@dev-sda1.target

Рис. 2.15: Списки целей

Далее получим список всех целей, которые можно изолировать, используем

команды `cd /usr/lib/systemd/system grep Isolate *.target`(рис. 2.16).

```
[root@edzhibitskaya ~]# cd /usr/lib/systemd/system
[root@edzhibitskaya 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@edzhibitskaya system]#
```

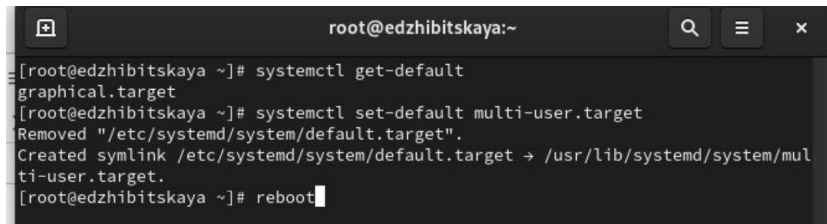
Рис. 2.16: Изолируемые цели

Переключимся в режим восстановления - `systemctl isolate rescue.target` и перезапустим систему(рис. 2.17).

```
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.
Give root password for maintenance
(or press Control-D to continue):
[root@edzhibitskaya ~]# systemctl isolate rescue.target_
```

Рис. 2.17: Режим восстановления и перезапуск

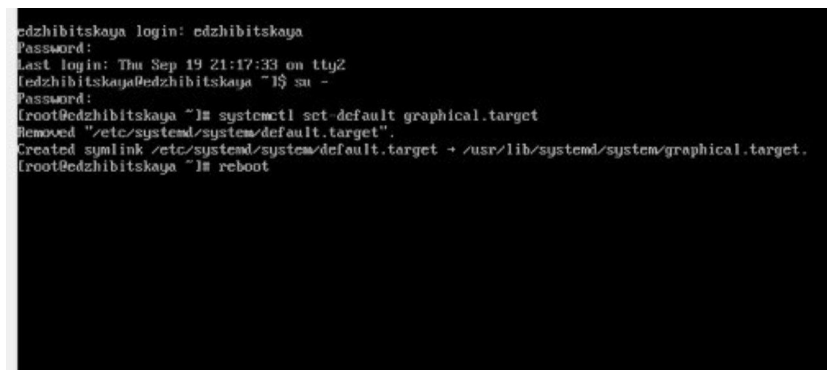
Изучим цели по умолчанию. Выведем установленную по умолчанию цель - `systemctl get-default`. Далее, для запуска по умолчанию текстового режима введем `systemctl set-default multi-user.target`(рис. 2.18).



```
root@edzhbitskaya:~  
[root@edzhbitskaya ~]# systemctl get-default  
graphical.target  
[root@edzhbitskaya ~]# 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@edzhbitskaya ~]# reboot
```

Рис. 2.18: Цель по умолчанию

Перезапустим систему. Получив полномочия, командой `systemctl set-default graphical.target` вернем графический режим(рис. 2.19).



```
edzhbitskaya login: edzhbitskaya  
Password:  
Last login: Thu Sep 19 21:17:33 on tty2  
edzhbitskaya@edzhbitskaya ~]$ su -  
Password:  
[root@edzhbitskaya ~]# 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@edzhbitskaya ~]# reboot
```

Рис. 2.19: Возвращение графического режима

Перезапустим систему и убедимся, что по умолчанию открывается графический режим(рис. 2.20).

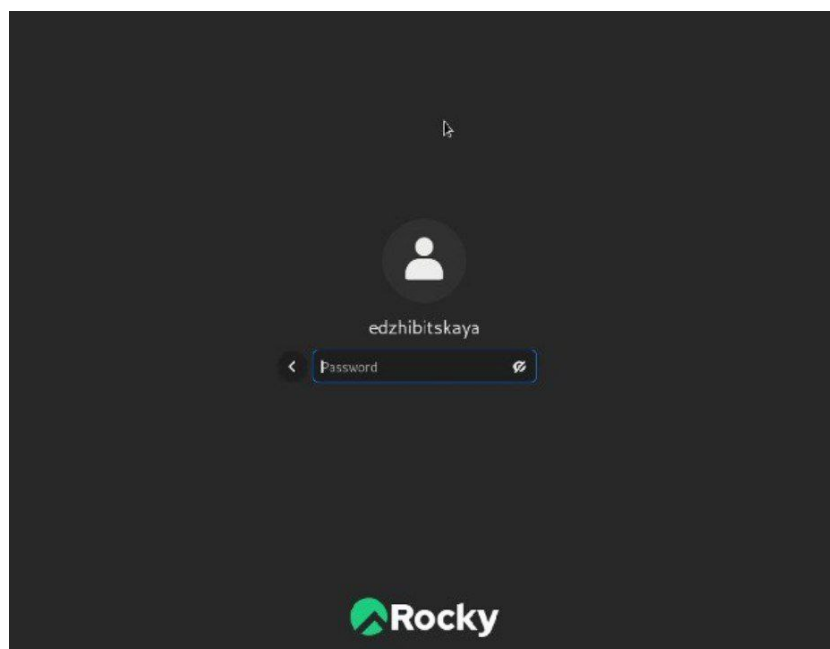


Рис. 2.20: Проверка

3 Ответы на контрольные вопросы

1. Что такое юнит (unit)? Приведите примеры. Юнит systemd — это основной объект управления, используемый для описания различных ресурсов и служб в системе. Каждый юнит имеет определенный тип и может управлять различными компонентами. Примеры юнитов:
 - service: управляет службами (например, http.service).
 - socket: используется для управления сокетами, на которые могут подписываться службы (например, http.socket).
 - target: группа юнитов, которые могут быть запущены вместе (например, multi-user.target).
 - timer: запускает юниты по расписанию (например, apt-daily.timer).
2. Какая команда позволяет вам убедиться, что цель больше не входит в список автоматического запуска при загрузке системы? systemctl disable (рис. 3.1).

```
[root@edzhbitskaya ~]# systemctl disable vsftpd
Removed "/etc/systemd/system/multi-user.target.wants/vsftpd.service".
[root@edzhbitskaya ~]# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset:
   Active: active (running) since Thu 2024-09-19 20:37:43 MSK; 6min ago
   Main PID: 4065 (vsftpd)
     Tasks: 1 (limit: 12194)
    Memory: 708.0K
       CPU: 7ms
    CGroup: /system.slice/vsftpd.service
            └─4065 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Sep 19 20:37:43 edzhbitskaya systemd[1]: Starting Vsftpd ftp daemon...
Sep 19 20:37:43 edzhbitskaya systemd[1]: Started Vsftpd ftp daemon.
lines 1-12/12 (END)
```

Рис. 3.1: Пример

Для проверки статуса можно использовать: systemctl is-enabled

3. Какую команду вы должны использовать для отображения всех сервисных юнитов, которые в настоящее время загружены? Для отображения всех загруженных сервисных юнитов используйте команду: `systemctl list-units -type=service`
4. Как создать потребность (wants) в сервисе? Чтобы создать зависимость типа wants для юнита, необходимо отредактировать конфигурацию юнита (обычно это файл .wants в каталоге, например, /etc/systemd/system/.wants/). В этом каталоге необходимо создать символическую ссылку на нужный юнит: `ln -s /etc/systemd/system/.service /etc/systemd/system/.wants/`
5. Как переключить текущее состояние на цель восстановления (rescue target)? Для переключения на цель восстановления используйте команду: `systemctl isolate rescue.target` (рис. 3.2).

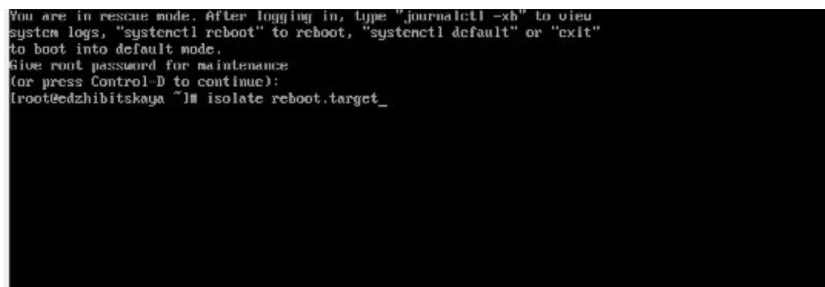


Рис. 3.2: Пример

6. Поясните причину получения сообщения о том, что цель не может быть изолирована. Сообщение о том, что цель не может быть изолирована, может появиться, если в системе есть активные юниты, которые мешают 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? Для того чтобы узнать, какие юниты зависят от определенной службы, используем команду: `systemctl list-dependencies` (рис. 3.3).

```

>> Sep 19 20:37:43 edzhbitskaya systemd[1]: Starting Vsftpd ftp daemon...
>> Sep 19 20:37:43 edzhbitskaya systemd[1]: Started Vsftpd ftp daemon.
>> [root@edzhbitskaya ~]# 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

```

Рис. 3.3: Пример

4 Выводы

В ходе работы было произведено знакомство с системными службами операционной системы. Были получены навыки управления системными службами посредством `systemd`.

Список литературы

ТУИС