

Отчет по лабораторной работе №5

Дисциплина: Основы администрирования операционных систем

Иванов Сергей Владимирович

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
4	Контрольные вопросы	13
5	Выводы	15

Список иллюстраций

3.1	Статус службы Very Secure FTP	6
3.2	Установка и запуск Very Secure	6
3.3	Добавляем в автозапуск	7
3.4	Убираем из автозапуска	7
3.5	Символические ссылки	7
3.6	Выводим символические ссылки	7
3.7	Статус Very Secure	8
3.8	Список зависимостей юнита	8
3.9	Список юнитов которые зависят от данного	8
3.10	Установка iptables	9
3.11	Проверка статусов	9
3.12	Проверка статусов	9
3.13	Запуск служб	9
3.14	Настройки конфликтов	10
3.15	Настройки конфликтов	10
3.16	Настройки запуска служб	10
3.17	Загруженные цели	11
3.18	Список всех целей	11
3.19	Цели которые можно изолировать	11
3.20	Перезапуск	11
3.21	Текстовый и графический режим	12

1 Цель работы

Получить навыки управления системными службами операционной системы посредством systemd.

2 Задание

1. Выполнить основные операции по запуску, определению статуса, добавлению (удалению) в автозапуск службы Very Secure FTP
2. Продемонстрировать навыки по разрешению конфликтов юнитов для служб `firewalld` и `iptables`
3. Продемонстрировать навыки работы с изолированными целями

3 Выполнение лабораторной работы

Получим полномочия администратора. Проверим статус службы Very Secure FTP: `systemctl status vsftpd` (рис. 1).

```
[svivanov1@svivanov1 ~]$ su -
Password:
[root@svivanov1 ~]# systemctl status vsftpd
Unit vsftpd.service could not be found.
[root@svivanov1 ~]# dnf -y install vsftpd
packages for the GitHub CLI                28 kB/s | 3.0 kB    00:00
packages for the GitHub CLI                12 kB/s | 2.7 kB    00:00
Rocky Linux 9 - BaseOS                    12 kB/s | 4.1 kB    00:00
Rocky Linux 9 - BaseOS                    4.3 MB/s | 2.3 MB    00:00
Rocky Linux 9 - AppStream                  13 kB/s | 4.5 kB    00:00
Rocky Linux 9 - AppStream                  11 MB/s | 8.0 MB    00:00
Rocky Linux 9 - Extras                     660 B/s | 2.9 kB    00:04
Dependencies resolved.
=====
Package           Architecture Version           Repository        Size
=====
```

Рис. 3.1: Статус службы Very Secure FTP

Установим службу Very Secure FTP: `dnf -y install vsftpd`. Запустим службу Very Secure FTP: `systemctl start vsftpd`. Проверим статус службы Very Secure FTP: `systemctl status vsftpd` (рис. 2).

```
[root@svivanov1 ~]# systemctl start vsftpd
[root@svivanov1 ~]# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset: >
   Active: active (running) since Sat 2024-10-05 13:20:40 MSK; 6s ago
   Process: 3911 ExecStart=/usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf (code=exit>
   Main PID: 3912 (vsftpd)
     Tasks: 1 (limit: 23032)
    Memory: 716.0K
       CPU: 2ms
   CGroup: /system.slice/vsftpd.service
           └─3912 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf
```

Рис. 3.2: Установка и запуск Very Secure

Добавим службу Very Secure FTP в автозапуск при загрузке операционной системы, используя команду `systemctl enable`. Затем проверим статус службы. (рис. 3).

```
[root@svivanov1 ~]# systemctl enable vsftpd
[root@svivanov1 ~]# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: disab
   Active: active (running) since Sat 2024-10-05 13:20:40 MSK; 3min 33s ago
   Main PID: 3912 (vsftpd)
     Tasks: 1 (limit: 23032)
    Memory: 716.0K
       CPU: 2ms
    CGroup: /system.slice/vsftpd.service
           └─3912 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf
```

Рис. 3.3: Добавляем в автозапуск

Удалим службу из автозапуска, используя команду `systemctl disable`, и снова проверим её статус. (рис. 4).

```
[root@svivanov1 ~]# systemctl disable vsftpd
Removed "/etc/systemd/system/multi-user.target.wants/vsftpd.service".
[root@svivanov1 ~]# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset: disa
   Active: active (running) since Sat 2024-10-05 13:20:40 MSK; 5min ago
   Main PID: 3912 (vsftpd)
     Tasks: 1 (limit: 23032)
    Memory: 716.0K
       CPU: 2ms
    CGroup: /system.slice/vsftpd.service
           └─3912 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf
```

Рис. 3.4: Убираем из автозапуска

Выведем на экран символические ссылки, ответственные за запуск различных сервисов: `ls /etc/systemd/system/multi-user.target.wants` (рис. 5).

```
[root@svivanov1 ~]# 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
```

Рис. 3.5: Символические ссылки

Снова добавим службу Very Secure FTP в автозапуск и выведем на экран символические ссылки, ответственные за запуск различных сервисов. (рис. 6).

```
[root@svivanov1 ~]# systemctl enable vsftpd
Created symlink /etc/systemd/system/multi-user.target.wants/vsftpd.service → /usr/lib/systemd/system/vsftpd.service.
[root@svivanov1 ~]# ls /etc/systemd/system/multi-user.target.wants
atd.service          kdump.service       rsyslog.service
auditd.service       libstoragemgmt.service sssd.service
avahi-daemon.service mcelog.service      tuned.service
chronyd.service      mdmonitor.service   vboxadd.service
crond.service        ModemManager.service vboxadd-service.service
cups.path            NetworkManager.service vmtoolsd.service
cups.service         remote-fs.target     vsftpd.service
firewalld.service    rsyslog.service
irqbalance.service  smartd.service
```

Рис. 3.6: Выводим символические ссылки

Снова проверим статус службы Very Secure FTP: `systemctl status vsftpd`. Теперь мы видим, что для файла юнита состояние изменено с `disabled` на `enabled`. (рис. 7).

```
[root@svivanov1 ~]# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: disab
   Active: active (running) since Sat 2024-10-05 13:20:40 MSK; 9min ago
     Main PID: 3912 (vsftpd)
       Tasks: 1 (limit: 23032)
      Memory: 716.0K
         CPU: 2ms
        CGroup: /system.slice/vsftpd.service
                └─3912 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Oct 05 13:20:40 svivanov1.localdomain systemd[1]: Starting Vsftpd ftp daemon...
Oct 05 13:20:40 svivanov1.localdomain systemd[1]: Started Vsftpd ftp daemon.
lines 1-12/12 (END)
```

Рис. 3.7: Статус Very Secure

Выведем на экран список зависимостей юнита: `systemctl list-dependencies vsftpd` (рис. 8).

```
[root@svivanov1 ~]# 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
└─vsftpd.service
```

Рис. 3.8: Список зависимостей юнита

Выведем на экран список юнитов, которые зависят от данного юнита: `systemctl list-dependencies vsftpd --reverse` (рис. 9).

```
[root@svivanov1 ~]# systemctl list-dependencies vsftpd --reverse
vsftpd.service
├─multi-user.target
└─graphical.target
[root@svivanov1 ~]#
```

Рис. 3.9: Список юнитов которые зависят от данного

Получим полномочия администратора. Установим `iptables`: `dnf -y install iptables*` (рис. 10).


```
[root@svivanov1 ~]# dnf -y install iptables*
Last metadata expiration check: 0:06:16 ago on Sat 05 Oct 2024 01:25:50 PM MSK.
Package iptables-libs-1.8.10-4.el9_4.x86_64 is already installed.
Package iptables-nft-1.8.10-4.el9_4.x86_64 is already installed.
Dependencies resolved.
=====
Package                Arch      Version      Repository      Size
=====
Installing:
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
=====
```

Рис. 3.10: Установка iptables

Проверим статус firewalld и iptables: `systemctl status firewalld`; `systemctl status iptables` (рис. 11). (рис. 12).

```
[root@svivanov1 ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; preset: en>
   Active: active (running) since Sat 2024-10-05 13:14:52 MSK; 17min ago
     Docs: man:firewalld(1)
    Main PID: 837 (firewalld)
      Tasks: 2 (limit: 23032)
     Memory: 44.4M
        CPU: 955ms
```

Рис. 3.11: Проверка статусов

```
[root@svivanov1 ~]# systemctl status iptables
○ iptables.service - IPv4 firewall with iptables
   Loaded: loaded (/usr/lib/systemd/system/iptables.service; disabled; preset: di>
   Active: inactive (dead)
[root@svivanov1 ~]#
```

Рис. 3.12: Проверка статусов

Попробуем запустить firewalld и iptables: `systemctl start firewalld`; `systemctl start iptables`. Видим, что при запуске одной службы вторая деактивируется или не запускается. (рис. 13)

```
[root@svivanov1 ~]# systemctl start firewalld
[root@svivanov1 ~]# systemctl start iptables
```

Рис. 3.13: Запуск служб

Введем `cat /usr/lib/systemd/system/firewalld.service` и опишем настройки конфликтов для этого юнита при наличии. (рис. 14).

```
[root@svivanov1 ~]# 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)
```

Рис. 3.14: Настройки конфликтов

Введем `cat /usr/lib/systemd/system/iptables.service` и опишем настройки конфликтов для этого юнита. (рис. 15).

```
[root@svivanov1 ~]# 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
```

Рис. 3.15: Настройки конфликтов

Выгрузим службу iptables: `systemctl stop iptables` и загрузим службу firewalld `systemctl start firewalld`. Заблокируем запуск iptables, введя: `systemctl mask iptables`. Попробуем запустить iptables: `systemctl start iptables`. Появилось сообщение об ошибке, указывающее, что служба замаскирована и по этой причине не может быть запущена. Попробуем добавить iptables в автозапуск: `systemctl enable iptables`. Сервис будет неактивен, а статус загрузки отобразится как замаскированный. (рис. 16).

```
[root@svivanov1 ~]# systemctl stop iptables
[root@svivanov1 ~]# systemctl start firewalld
[root@svivanov1 ~]# systemctl mask iptables
Created symlink /etc/systemd/system/iptables.service -> /dev/null.
[root@svivanov1 ~]# ls /etc/systemd/system/iptables.service
/etc/systemd/system/iptables.service
[root@svivanov1 ~]# systemctl start iptables
Failed to start iptables.service: Unit iptables.service is masked.
[root@svivanov1 ~]# systemctl enable iptables
Failed to enable unit: Unit file /etc/systemd/system/iptables.service is masked.
[root@svivanov1 ~]#
```

Рис. 3.16: Настройки запуска служб

Чтобы получить список всех активных загруженных целей, вводим: `systemctl -type=target` (рис. 17)

```
[root@svivanov1 ~]# 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

Рис. 3.17: Загруженные цели

Чтобы получить список всех целей, введем: `systemctl --type=target --all` (рис. 18)

```
[root@svivanov1 ~]# cd /usr/lib/systemd/system
[root@svivanov1 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
```

Рис. 3.18: Список всех целей

Перейдем в каталог `systemd` и найдем список всех целей, которые можно изолировать: `cd /usr/lib/systemd/system; grep Isolate *.target` (рис. 19)

```
system-update.target:AllowIsolate=yes
[root@svivanov1 system]# systemctl isolate rescue.target
```

Рис. 3.19: Цели которые можно изолировать

Переключим операционную систему в режим восстановления: `systemctl isolate rescue.target` и перезапустим операционную систему следующим образом: `systemctl isolate reboot.target` (рис. 20)

```
[svivanov1@svivanov1 ~]$ su -
Password:
[root@svivanov1 ~]# systemctl get-default
graphical.target
[root@svivanov1 ~]# systemctl set-default
Too few arguments.
[root@svivanov1 ~]# 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@svivanov1 ~]#
```

Рис. 3.20: Перезапуск

Перезагрузим систему командой `reboot`. Убедимся, что система загрузилась в текстовом режиме. Для запуска по умолчанию графического режима введем

systemctl set-default graphical.target. Вновь перегрузим систему командой reboot. Убедимся, что система загрузилась в графическом режиме. (рис. 21)

```
[svivanov1@svivanov1 ~]$ systemctl set-default graphical.target
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-unit-files ====
Authentication is required to manage system service or unit files.
Multiple identities can be used for authentication:
 1. svivanov1
 2. alice
Choose identity to authenticate as (1-2): 1
Password:
==== AUTHENTICATION COMPLETE ====
Removed "/etc/systemd/system/default.target".
Created symlink /etc/systemd/system/default.target → /usr/lib/systemd/system/graphical.target.
[svivanov1@svivanov1 ~]$ reboot_
```

Рис. 3.21: Текстовый и графический режим

4 Контрольные вопросы

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

Unit – объект, которым может управлять система.

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

`systemctl is-enable “имя_юнита”` (пример: `systemctl is-enable vsftpd.service`).

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

`system list-units`.

4. Как создать потребность (wants) в сервисе?

Нужно внести всю необходимую информацию в переменную “Wants”, которая находится в файле `имя_сервиса.service`.

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

`systemctl set-default rescue.target`.

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

Изолируя цель, мы запускаем эту цель со всеми её зависимостями. Не все цели могут быть изолированы (в случае, если цель является неотъемлемой частью `system`).

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` (пример: `systemctl list-dependencies firewalld.service --reverse`).

5 Выводы

В ходе выполнения лабораторной работы были получены навыки управления с системными службами операционной системы посредством systemd.