

Лабораторная работа №5

Основы администрирования операционных систем.

Бызова М.О.

30 сентября 2024

Российский университет дружбы народов, Москва, Россия.

Целью данной работы является получение навыков управления системными службами операционной системы посредством systemd.

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).

Выполнение лабораторной работы

Снова проверим статус службы Very Secure FTP: `systemctl status vsftpd`. Вывод команды показывает, что служба в настоящее время работает, но не будет активирована при перезапуске операционной системы (рис. 2).



```
systemctl status vsftpd
vsftpd.service - Very Secure FTP Daemon
Loaded: loaded (/usr/lib/systemd/systemd vsftpd.service; vendor preset: disabled)
Active: active (running) since Sun 2016-11-20 12:04:14 MSK; 1min 1s ago
Main PID: 1000
CGroup: /systemd/systemd/slices/user.slice/user-1000.slice/user@1000.service
└─ 1000 /usr/sbin/vsftpd

vsftpd.service is active (running)
  Main PID: 1000
  CGroup: /systemd/systemd/slices/user.slice/user-1000.slice/user@1000.service
          └─ 1000 /usr/sbin/vsftpd
```

Рис. 2: Проверка статуса службы Very Secure FTP.

Добавим службу Very Secure FTP в автозапуск при загрузке операционной системы, используя команду `systemctl enable vsftpd`. Затем проверим статус службы и удалим службу из автозапуска, используя команду `systemctl disable vsftpd`, и снова проверим её статус (рис. 3, рис. 4).

```

root@kali:~# systemctl status vultp
● systemd-journal-journald.service
   Loaded: loaded (/usr/lib/systemd/system/journal-journald.service; vendor preset: disabled)
   Active: active (running) since Mon 2024-08-20 18:00:39 PM EDT; 1min 46s ago
   Main PID: 4000 (journald)
   Tasks: 3 (limit: 24614)
   Memory: 768.0K
   CPU: 0ms
   CGroup: /system.slice/systemd-journald
           └─ns: /usr/bin/systemd-journald-journald

Sep 20 18:00:39 multihost.localtime systemd[1]: Starting vultpd Tlp daemon...
Sep 20 18:00:39 multihost.localtime systemd[1]: Started vultpd Tlp daemon.
● systemd-journald-journald.service

```

Рис. 3: Добавление службы Very Secure FTP в автозапуск и проверка её статуса.


```
[root@mobhzova ~]# systemctl disable vsftpd
Removed /etc/systemd/system/multi-user.target.wants/vsftpd.service.
[root@mobhzova ~]# systemctl status vsftpd
* vsftpd.service - Vsftpd Ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset: disabled)
   Active: active (running) since Sun 2024-09-29 19:40:19 MSK; 3min 46s ago
     Main PID: 4929 (vsftpd)
       Tasks: 1 (limit: 24674)
        Memory: 768.0K
           CPU: 3ms
      CGroup: /system.slice/vsftpd.service
              └─4929 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Sep 29 19:40:19 mobhzova.localdomain systemd[1]: Starting Vsftpd ftp daemon...
Sep 29 19:40:19 mobhzova.localdomain systemd[1]: Started Vsftpd ftp daemon.
[root@mobhzova ~]#
```

Рис. 4: Удаление службы Very Secure FTP из автозапуска и проверка её статуса.

Далее выведем на экран символические ссылки, ответственные за запуск различных сервисов. Отображается, что ссылка на `vsftpd.service` не существует. Снова добавляем службу Very Secure FTP в автозапуск: `systemctl enable vsftpd` и выводим на экран символические ссылки, ответственные за запуск различных сервисов. Вывод команды показывает, что создана символическая ссылка для файла `/usr/lib/systemd/system/vsftpd.service` в каталоге `/etc/systemd/system/multi-user.target.wants`. Проверяем статус службы Very Secure FTP: `systemctl status vsftpd`. Теперь мы видим, что для файла юнита состояние изменено с `disabled` на `enabled` (рис. 5).

```
root@kali:~# systemctl status vsftpd
vsftpd.service - Very Secure FTP Daemon
Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; vendor preset: enabled)
Active: active (running) since Wed 2024-05-01 12:00:00 MSK; 1min ago
Main PID: 1000
CGroup: /systemd/system/vsftpd.service
└─ 1000 /usr/sbin/vsftpd

root@kali:~# systemctl start vsftpd
root@kali:~# systemctl status vsftpd
vsftpd.service - Very Secure FTP Daemon
Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; vendor preset: enabled)
Active: active (running) since Wed 2024-05-01 12:00:00 MSK; 1min ago
Main PID: 1000
CGroup: /systemd/system/vsftpd.service
└─ 1000 /usr/sbin/vsftpd
```

Рис. 5: Вывод на экран символических ссылок, добавление службы Very Secure FTP в автозапуск, проверка статуса службы.

Выведем на экран список зависимостей юнита: `systemctl list-dependencies vsftpd` и список юнитов, которые зависят от данного юнита: `systemctl list-dependencies vsftpd --reverse` (рис. 6, рис. 7).

```
[root@elbhzova ~]# systemctl list-dependencies vsftpd
vsftpd.service
└─system.slice
   └─systemd.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
      ├─nfs-domainname.service
      ├─plymouth-read-write.service
      ├─plymouth-start.service
      ├─proc-sys-fs-binfmt_misc.automount
      ├─selinux-autorelabel-mark.service
      └─sys-fs-fuse-connections.mount
```

Рис. 6: Список зависимостей юнита.



Рис. 7: Список юнитов, которые зависят от данного юнита.

Получим полномочия администратора su – и установим iptables: `dnf -y install iptables*` (рис. 8).

[illegible]

Рис. 8: Получение полномочий администратора и установка iptables.

Введем `cat /usr/lib/systemd/system/firewalld.service` и опишем настройки конфликтов для этого юнита при наличии, далее введём `cat /usr/lib/systemd/system/iptables.service` и опишем настройки конфликтов для этого юнита (рис. 11).

```
root@mdzhova:~# cat /usr/lib/systemd/system/firewalld.service
[Unit]
Description=Firewalld - dynamic firewall daemon
Before=network-pre.target
Wants=network-pre.target
After=dbus.service
Conflicts=iptables.service iptables.service nftables.service
Documentation=man:firewalld(1)

[Service]
EnvironmentFile=/etc/sysconfig/firewalld
ExecStart=/usr/sbin/firewalld --nofork --nagld SPINALLD_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=mix

[Install]
WantedBy=multi-user.target

root@mdzhova:~# cat /usr/lib/systemd/system/iptables.service
[Unit]
Description=IPTables 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

[Install]
WantedBy=multi-user.target
root@mdzhova:~#
```

Рис. 11: Настройки конфликтов для юнитов.

Выгрузим службу iptables (на всякий случай, чтобы убедиться, что данная служба не загружена в систему): `systemctl stop iptables` и загрузим службу firewalld: `systemctl start firewalld`. Далее заблокируем запуск iptables, введя: `systemctl mask iptables`. Видим, как создана символическая ссылка на `/dev/null` для `/etc/systemd/system/iptables.service`. Поскольку юнит-файлы в `/etc/systemd` имеют приоритет над файлами в `/usr/lib/systemd`, то это сделает невозможным случайный запуск сервиса iptables. Для проверки попробуем запустить iptables: `systemctl start iptables`. После попытки запуска появилось сообщение об ошибке, указывающее, что служба замаскирована и по этой причине не может быть запущена. Теперь попробуем добавить iptables в автозапуск: `systemctl enable iptables`. Показывает, что сервис неактивен, а статус загрузки отображается как замаскированный (рис. 12).

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

Рис. 12: Выгрузка службы iptables, загрузка службы firewalld, блокировка запуска iptables, попытка запуска iptables и добавления iptables в автозапуск.

Изолируемые цели

Получим полномочия администратора su – и перейдём в каталог systemd, найдём список всех целей, которые можно изолировать: `cd /usr/lib/systemd/system grep Isolate *.target`. На следующем шаге переключим операционную систему в режим восстановления: `systemctl isolate rescue.target` (рис. 13).

```
(root@mobihzova ~)# systemctl start iptables
Failed to start iptables.service: Unit iptables.service is masked.
(root@mobihzova ~)# systemctl enable iptables
Failed to enable unit: Unit file /etc/systemd/system/iptables.service is masked.
(root@mobihzova ~)# cd /usr/lib/systemd/system
(root@mobihzova 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@mobihzova system)# systemctl isolate rescue.target
```

Рис. 13: Открытие каталога. Нахождение списка всех целей, которые можно изолировать. Переключение операционной системы в режим восстановления

Как только операционная система переключилась в режим восстановления вводим пароль root. После чего перезапустим операционную систему следующим образом: `systemctl isolate reboot.target` (рис. 14).

A terminal window with a black background and white text. The text reads: "You are in rescue mode. After logging in, type 'journalctl -xb' to view the journal log. 'systemctl reboot' to reboot, 'systemctl default' or 'systemctl isolate reboot.target' to reboot into default mode. Now pass password for rootuser and press Ctrl+D to continue: _".

```
You are in rescue mode. After logging in, type 'journalctl -xb' to view  
the journal log. 'systemctl reboot' to reboot, 'systemctl default' or 'systemctl  
isolate reboot.target' to reboot into default mode.  
Now pass password for rootuser  
and press Ctrl+D to continue: _
```

Рис. 14: Перезапуск операционной системы.

Получим полномочия администратора `su` – и выведем на экран цель, установленную по умолчанию: `systemctl get-default`. Для установки цели по умолчанию используется команда `systemctl set-default`. В нашем случае для запуска по умолчанию текстового режима введём `systemctl set-default multi-user.target`. После чего перезагрузим систему командой `reboot` (рис. 15).

Цель по умолчанию

```
root@kali:~# su -  
Password:  
root@kali:~# systemctl get-default  
multi-user.target  
root@kali:~# systemctl set-default  
Too few arguments.  
root@kali:~# 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@kali:~# reboot
```

Рис. 15: Получение полномочий администратора, установка запуска по умолчанию текстового режима, последующая перезагрузка системы.

Убедимся, что система загрузилась в текстовом режиме, после чего получим полномочия администратора (для начала зайдём в пользователя `tomihzova`, а затем в режим администратора). Для запуска по умолчанию графического режима введём `systemctl set-default graphical.target` и вновь перегрузим систему командой `reboot`. Убедимся, что система загрузилась в графическом режиме (рис. 16, рис. 17).

```
Rocky Linux 7.4 (Blue Onyx)
Kernel 3.10.0-527.11.1.el7.x86_64 on an x86_64

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

root@rocky: ~#
root@rocky: ~# systemctl enable cockpit.socket
Created symlink from /etc/systemd/system/default.target to /usr/lib/systemd/system/cockpit.socket.
root@rocky: ~# systemctl restart cockpit.socket
root@rocky: ~# systemctl restart cockpit.socket
```

Рис. 16: Загрузка системы в текстовом режиме, получение полномочий администратора, установка запуска по умолчанию графического режима и последующая перезагрузка системы.

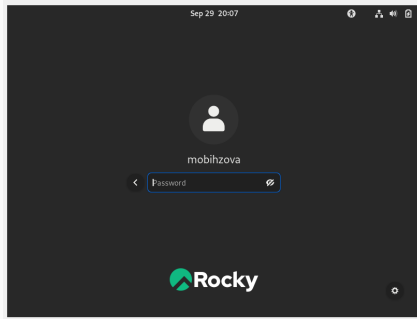


Рис. 17: Загрузка системы в графическом режиме.

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

1. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВ-Петербург, 2010.
2. Колисниченко Д. Н. Самоучитель системного администратора Linux. — СПб. : БХВ-Петербург, 2011. — (Системный администратор).
3. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — (Классика Computer Science).
4. Neil N. J. Learning CentOS: A Beginners Guide to Learning Linux. — CreateSpace Independent Publishing Platform, 2016.
5. Unix и Linux: руководство системного администратора / Э. Немец, Г. Снайдер, Т.Хейн, Б. Уэйли, Д. Макни. — 5-е изд. — СПб. : ООО «Диалектика», 2020.