

Лабораторная работа 9

1132232887

Накова Амина Михайловна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	11
	Список литературы	12

Список иллюстраций

2.1 шаг 1	6
2.2 шаг 2	7
2.3 шаг 3	7
2.4 шаг 4	7
2.5 шаг 5	8
2.6 шаг 6	8
2.7 шаг 7	8
2.8 шаг 8	9
2.9 шаг 9	9

Список таблиц

1 Цель работы

Целью данной работы является получение навыков работы с планировщиками событий cron и at.

2 Выполнение лабораторной работы

Планирование задач с помощью cron: Мониторинг журнала системных событий в реальном времени: Запустим терминал и получим полномочия администратора: `su -`. Просмотрим статус демона `crond`: `systemctl status crond -l` и содержимое файла конфигурации `/etc/crontab`: `cat /etc/crontab`

```
SELinux status:                enabled
SELinuxfs mount:              /sys/fs/selinux
SELinux root directory:       /etc/selinux
Loaded policy name:            targeted
Current mode:                  enforcing
Mode from config file:         enforcing
Policy MLS status:             enabled
Policy deny_unknown status:    allowed
Memory protection checking:    actual (secure)
Max kernel policy version:     33

Process contexts:
Current context:               unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
Init context:                  system_u:system_r:init_t:s0
/usr/sbin/sshd                 system_u:system_r:sshd_t:s0-s0:c0.c1023

File contexts:
Controlling terminal:         unconfined_u:object_r:user_devpts_t:s0
/etc/passwd                   system_u:object_r:passwd_file_t:s0
/etc/shadow                   system_u:object_r:shadow_t:s0
/bin/bash                     system_u:object_r:shell_exec_t:s0
/bin/login                    system_u:object_r:login_exec_t:s0
/bin/sh                       system_u:object_r:bin_t:s0 -> system_u:object_r:shell_exec_t:s0
xec_t:s0
/sbin/agetty                  system_u:object_r:getty_exec_t:s0
/sbin/init                    system_u:object_r:bin_t:s0 -> system_u:object_r:init_t:s0
```

Рис. 2.1: шаг 1

Теперь посмотрим список заданий в расписании: `crontab -l`. Ничего не отобразилось, так как расписание ещё не задано. Далее откроем файл расписания на редактирование: `crontab -e`

```

getenforce

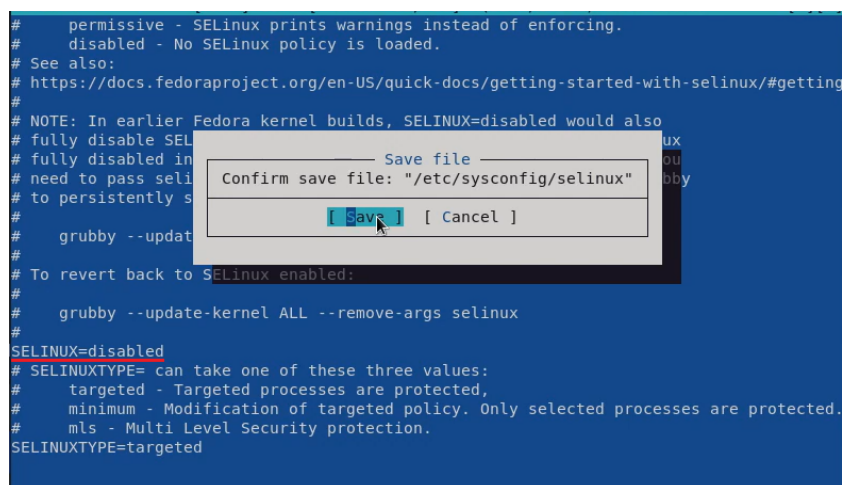
setenforce 0
getenforce

mcedit /etc/sysconfig/selinux

```

Рис. 2.2: шаг 2

Предыдущая команда запустила интерфейс редактора (по умолчанию используется vi). Добавим следующую строку в файл расписания (запись сообщения в системный журнал), используя Ins для перехода в vi в режим ввода: `/1 * * * logger This message is written from root cron`. Закроем сеанс редактирования vi и сохраним изменения, используя команду vi: Esc : wq



```

# permissive - SELinux prints warnings instead of enforcing.
# disabled - No SELinux policy is loaded.
# See also:
# https://docs.fedoraproject.org/en-US/quick-docs/getting-started-with-selinux/#getting
#
# NOTE: In earlier Fedora kernel builds, SELINUX=disabled would also
# fully disable SELinux.
# fully disabled in
# need to pass seli
# to persistently s
#
# grubby --updat
#
# To revert back to SELinux enabled:
#
# grubby --update-kernel ALL --remove-args selinux
#
SELINUX=disabled
# SELINUXTYPE= can take one of these three values:
# targeted - Targeted processes are protected,
# minimum - Modification of targeted policy. Only selected processes are protected.
# mls - Multi Level Security protection.
SELINUXTYPE=targeted

```

Рис. 2.3: шаг 3

Просмотрим список заданий в расписании: `crontab -l` (в расписании появилась запись о запланированном событии). Не выключая систему, через некоторое время (2–3 минуты) посмотрим журнал системных событий: `grep written /var/log/messages`

```

reboot

```

Рис. 2.4: шаг 4

Вернёмся в текстовый редактор vi и изменим запись в расписании crontab на следующую: 0/1 * 1-5 logger This message is written from root cron



Рис. 2.5: шаг 5

Теперь посмотрим список заданий в расписании: crontab -l

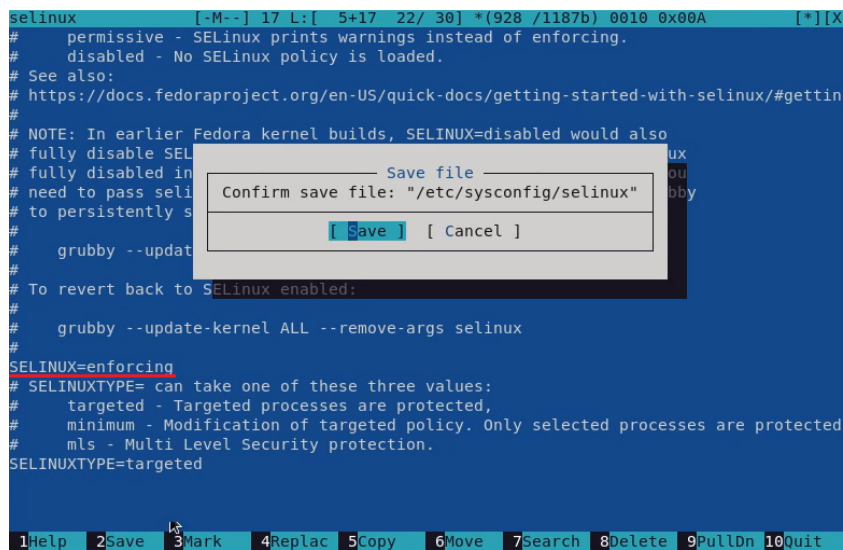


Рис. 2.6: шаг 6

Перейдём в каталог /etc/cron.hourly и создадим в нём файл сценария с именем eachhour : cd /etc/cron.hourly touch eachhour Далее откроем файл eachhour для редактирования и пропишем в нём следующий скрипт (запись сообщения в системный журнал) (Рис. 1.8): #!/bin/sh logger This message is written at \$(date)

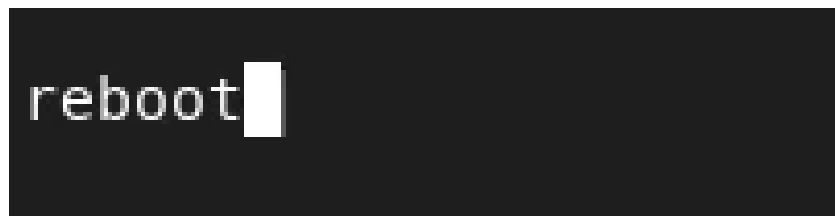


Рис. 2.7: шаг 7


```
RETHleed attacks, data leaks possible!
2.0037001 Warning: Unmaintained hardware is detected: e1000:100E:8086 @ 00
00:00:03.0
2.3450881 [drm:vmw_host_printf [vmwgfx]] *ERROR* Failed to send host log me
sage.
8.4096931 selinux-autorelabel[720]: *** Warning -- SELinux targeted policy relabel is required
8.4098801 selinux-autorelabel[720]: *** Relabeling could take a very long time, depending on
e
8.4099951 selinux-autorelabel[720]: *** system size and speed of hard drives.
8.4202661 selinux-autorelabel[720]: Running: /sbin/fixfiles -T 0 restore
21.9423221 selinux-autorelabel[726]: Relabeling /boot /dev /dev/hugepages /dev/mqueue /dev/p
/dev/shm /run /sys /sys/fs/cgroup /sys/fs/pstore /sys/kernel/debug /sys/kernel/tracing
```

Рис. 2.8: шаг 8

Сделаем файл сценария eachhour исполняемым: `chmod +x eachhour` Теперь перейдём в каталог `/etc/cron.d` и создадим в нём файл с расписанием eachhour : `cd /etc/cron.d touch eachhour` Откроем этот файл для редактирования и поместим в него следующее содержимое: `11 * * * root logger This message is written from /etc/cron.d` Сохраним изменения

```
SELinuxfs mount: /sys/fs/selinux
SELinux root directory: /etc/selinux
Loaded policy name: targeted
Current mode: enforcing
Mode from config file: enforcing
Policy MLS status: enabled
Policy deny unknown status: allowed
Memory protection checking: actual (secure)
Max kernel policy version: 33

Process contexts:
Current context: unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
Init context: system_u:system_r:init_t:s0
/usr/sbin/sshd system_u:system_r:sshd_t:s0-s0:c0.c1023

File contexts:
Controlling terminal: unconfined_u:object_r:user_devpts_t:s0
/etc/passwd system_u:object_r:passwd_file_t:s0
/etc/shadow system_u:object_r:shadow_t:s0
/bin/bash system_u:object_r:shell_exec_t:s0
/bin/login system_u:object_r:login_exec_t:s0
/bin/sh system_u:object_r:bin_t:s0 -> system_u:object_r:shell_e
xec_t:s0
/sbin/agetty system_u:object_r:getty_exec_t:s0
/sbin/init system_u:object_r:bin_t:s0 -> system_u:object_r:init ex
```

Рис. 2.9: шаг 9

Ответы на контрольные вопросы: 1. Как настроить задание cron, чтобы оно выполнялось раз в 2 недели? `00 00 1,15 * * logger task` 2. Как настроить задание cron, чтобы оно выполнялось 1-го и 15-го числа каждого месяца в 2 часа ночи? `00 02 1,15 * * logger task` 3. Как настроить задание cron, чтобы оно выполнялось каждые 2 минуты каждый день? `/2 * * * logger task` 4. Как настроить задание cron, чтобы оно выполнялось 19 сентября ежегодно? `* * 19 9 logger task` 5. Как настроить задание cron, чтобы оно выполнялось каждый четверг сентября ежегодно? `* * * 4 logger task` 6. Какая команда позволяет вам назначить задание cron для

пользователя alice? Приведите подтверждающий пример. * * * * alice logger task

7. Как указать, что пользователю bob никогда не разрешено назначать задания через cron? Приведите подтверждающий пример. записать его в /etc/cron.deny 9 8.

Вам нужно убедиться, что задание выполняется каждый день, даже если сервер во время выполнения временно недоступен. Как это сделать? Найти задание в логах grep cron /var/log/messages 9. Какая команда позволяет узнать, запланированы ли какие-либо задания на выполнение планировщиком atd? atq

3 Выводы

В ходе выполнения лабораторной работы были получены навыки работы с планировщиками событий `stop` и `at`.

Список литературы