Отчёт о лабораторной работе

Лабораторная работа 15

Мошаров Денис Максимович

Содержание

Цель работы

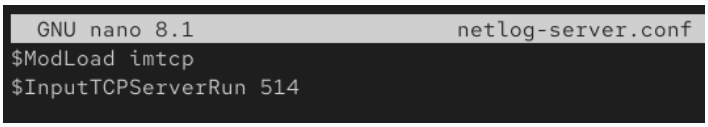
Получение навыков по работе с журналами системных событий

Выполнение лабораторной работы

На сервере перейдем в каталог конфигурации rsyslog и создадим файл netlog-server.conf для настройки централизованного сбора логов (рис. [-@fig:001]).

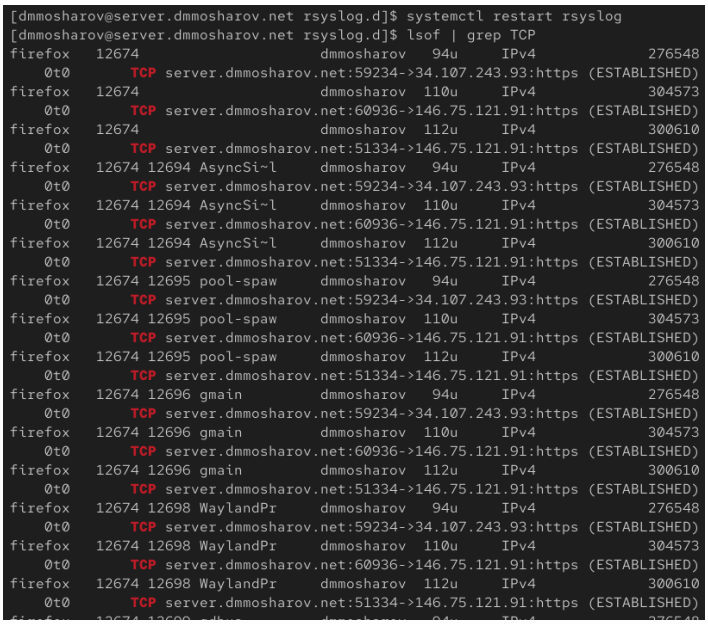
Создание файла конфигурации на сервере

В открывшемся редакторе внесем настройки для включения модуля imtcp и прослушивания порта 514 по протоколу TCP (рис. [-@fig:002]).



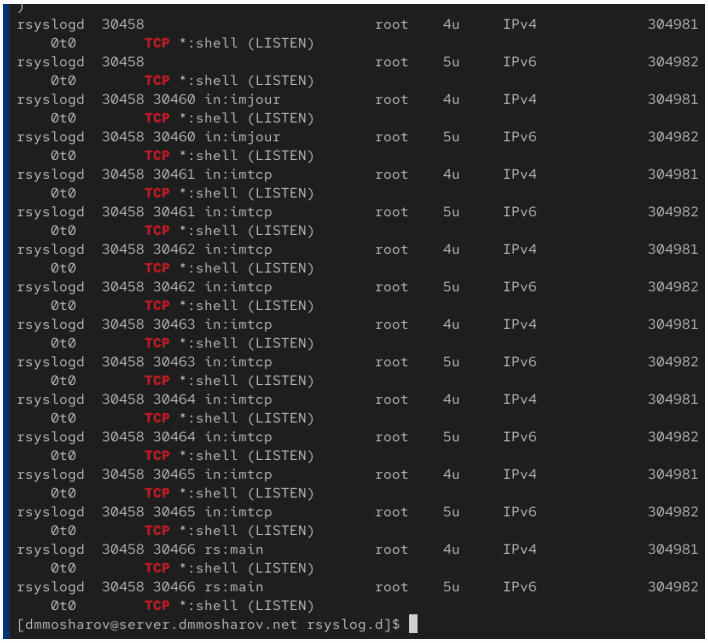
Редактирование конфигурации rsyslog на сервере

Перезапустим службу rsyslog для применения изменений и начнем проверку прослушиваемых портов с помощью утилиты lsof, отфильтровав вывод по TCP (рис. [-@fig:003]).



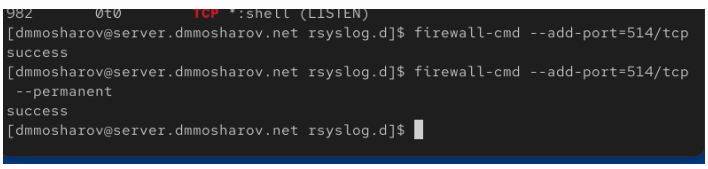
Перезапуск службы и начало проверки портов

В продолжении вывода команды lsof убедимся, что процессы rsyslogd успешно прослушивают порт 514 на всех интерфейсах (рис. [-@fig:004]).



Подтверждение прослушивания порта 514

Настроим межсетевой экран на сервере, разрешив прием соединений по порту 514/tcp, и сохраним правило как постоянное (рис. [-@fig:005]).



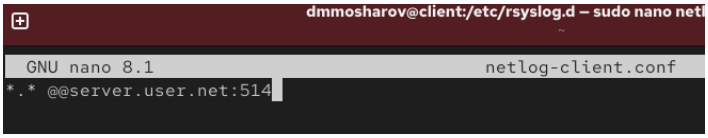
Настройка firewall на сервере

Перейдем на клиентскую машину. В каталоге /etc/rsyslog.d создадим файл конфигурации netlog-client.conf (рис. [-@fig:006]).

```
[dmmosharov@client.dmmosharov.net ~]$ cd /etc/rsyslog.d
[dmmosharov@client.dmmosharov.net rsyslog.d]$ touch netlog-client.conf
touch: cannot touch 'netlog-client.conf': Permission denied
[dmmosharov@client.dmmosharov.net rsyslog.d]$ sudo touch netlog-client.conf
[sudo] password for dmmosharov:
[dmmosharov@client.dmmosharov.net rsyslog.d]$ sudo nano netlog-client.conf
[dmmosharov@client.dmmosharov.net rsyslog.d]$
```

Создание файла конфигурации на клиенте

В файле конфигурации пропишем правило отправки всех журналов (.) на сервер по адресу server.nsandryushin.net через порт 514 по протоколу TCP (рис. [-@fig:007]).



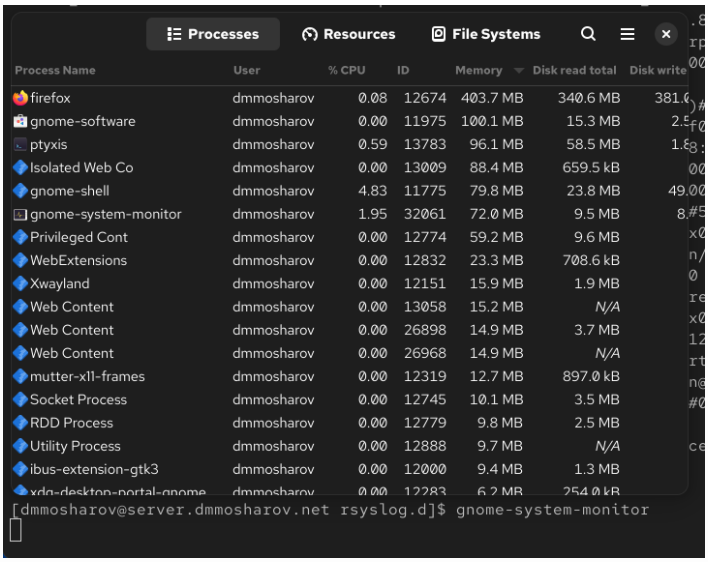
Редактирование конфигурации rsyslog на клиенте

Вернемся на сервер и запустим отслеживание файла /var/log/messages в реальном времени, чтобы убедиться в поступлении логов (рис. [-@fig:008]).

```
[dmmosharov@server.dmmosharov.net rsyslog.d]$ sudo tail -f /var/log/messages
[sudo] password for dmmosharov:
Feb 11 12:50:28 server ptyxis[13783]: context mismatch in svga_surface_destro
y
Feb 11 12:50:28 server systemd-coredump[31940]: Process 31935 (VBoxClient) of
 user 1001 dumped core.#012#012Module libXau.so.6 from rpm libXau-1.0.11-8.el
10.x86_64#012Module libxcb.so.1 from rpm libxcb-1.17.0-3.el10.x86_64#012Modul
e libX11.so.6 from rpm libX11-1.8.10-1.el10.x86_64#012Module libffi.so.8 from
 rpm libffi-3.4.4-9.el10.x86_64#012Module libwayland-client.so.0 from rpm way
land-1.23.0-2.el10.x86_64#012Stack trace of thread 31938:#012#0  0x0000000000
41db4b n/a (n/a + 0x0)#012#1  0x000000000041dac4 n/a (n/a + 0x0)#012#2  0x000
0000000450a7c n/a (n/a + 0x0)#012#3  0x0000000000435890 n/a (n/a + 0x0)#012#4
  0x00007f086c153b68 start_thread (libc.so.6 + 0x94b68)#012#5  0x00007f086c1c
46bc __clone3 (libc.so.6 + 0x1056bc)#012#012Stack trace of thread 31935:#012#
0  0x00007f086c1c24bd syscall (libc.so.6 + 0x1034bd)#012#1  0x00000000004347a
2 n/a (n/a + 0x0)#012#2  0x00000000004506c6 n/a (n/a + 0x0)#012#3  0x00000000
00405123 n/a (n/a + 0x0)#012#4  0x00007f086c0e930e __libc_start_call_main (li
bc.so.6 + 0x2a30e)#012#5  0x00007f086c0e93c9 __libc_start_main@@GLIBC_2.34 (l
ibc.so.6 + 0x2a3c9)#012#6  0x00000000004044aa n/a (n/a + 0x0)#012ELF object b
inary architecture: AMD x86-64
Feb 11 12:50:28 server systemd[1]: systemd-coredump@1611-31939-0.service: Dea
ctivated successfully.
Feb 11 12:50:32 server systemd-logind[991]: Existing logind session ID 5 used
 by new audit session, ignoring.
Feb 11 12:50:32 server systemd[1]: Created slice user-0.slice - User Slice of
 UID 0.
Feb 11 12:50:32 server systemd[1]: Starting user-runtime-dir@0.service - User
 Runtime Directory /run/user/0...
Feb 11 12:50:32 server systemd-logind[991]: New session c11 of user root.
Feb 11 12:50:32 server systemd[1]: Finished user-runtime-dir@0.service - User
 Runtime Directory /run/user/0.
Feb 11 12:50:32 server systemd[1]: Starting user@0.service - User Manager for
 UID 0...
Feb 11 12:50:32 server systemd-logind[991]: New session 14 of user root.
Feb 11 12:50:32 server systemd[31949]: Queued start job for default target de
fault.target.
Feb 11 12:50:32 server systemd[31949]: Created slice app.slice - User Applica
```

Просмотр логов на сервере через tail

Запустим графическую утилиту мониторинга системы. В открывшемся окне gnome-system-monitor просмотрим список активных процессов и потребляемые ими ресурсы (рис. [-@fig:009]).



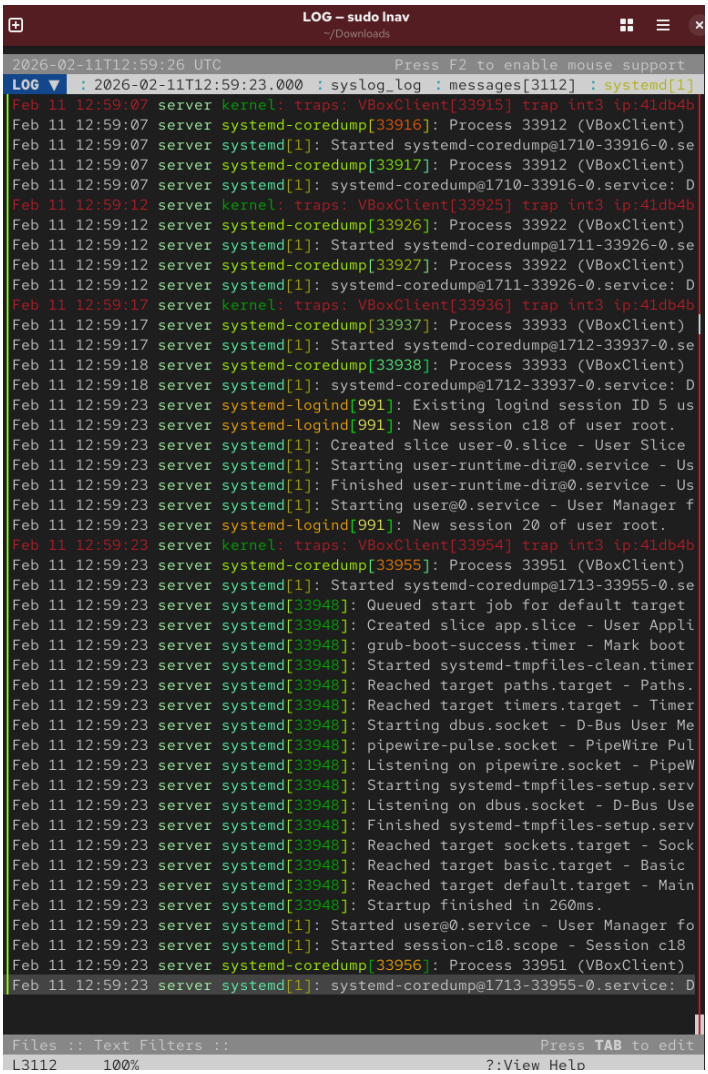
Интерфейс Gnome System Monitor

Установим продвинутый просмотрщик логов lnav на сервере, скопировав исполняемый файл в системную директорию /usr/bin (рис. [-@fig:010]).

```
[dmmosharov@server.dmmosharov.net rsyslog.d]$ cd ~/Downloads/
[dmmosharov@server.dmmosharov.net Downloads]$ sudo cp lnav /usr/bin
cp: cannot stat 'lnav': No such file or directory
[dmmosharov@server.dmmosharov.net Downloads]$ sudo cp lnav /usr/bin
[dmmosharov@server.dmmosharov.net Downloads]$ lnav
✗ error: default syslog file is not readable -- /var/log/messages
[dmmosharov@server.dmmosharov.net Downloads]$ sudo lnav
[dmmosharov@server.dmmosharov.net Downloads]$
```

Ручная установка lnav на сервере

Запустим lnav на сервере и просмотрим журналы в удобном структурированном виде с подсветкой синтаксиса (рис. [-@fig:011]).



Просмотр логов через lnav на сервере

Приступим к автоматизации настроек. На сервере в каталоге /vagrant/provision создадим структуру папок для хранения конфигов и скопируем туда текущий файл настроек. Также создадим скрипт netlog.sh (рис. [-@fig:014]).

```
[dmmosharov@server.dmmosharov.net Downloads]$ cd /vagrant/provision/server
[dmmosharov@server.dmmosharov.net server]$ mkdir -p /vagrant/provision/server
/netlog/etc/rsyslog.d
[dmmosharov@server.dmmosharov.net server]$ cp -R /etc/rsyslog.d/netlog-server
.conf /vagrant/provision/server/netlog/etc/rsyslog.d
[dmmosharov@server.dmmosharov.net server]$ cd /vagrant/provision/server
[dmmosharov@server.dmmosharov.net server]$ touch netlog.sh
[dmmosharov@server.dmmosharov.net server]$ chmod +x netlog.sh
[dmmosharov@server.dmmosharov.net server]$ nano netlog.sh
[dmmosharov@server.dmmosharov.net server]$
```

Подготовка файлов для автоматизации сервера

Напишем скрипт netlog.sh, который будет копировать конфигурационные файлы, восстанавливать контекст безопасности SELinux, настраивать фаервол и перезапускать службу rsyslog (рис. [-@fig:015]).

```
 GNU nano 8.1                                netlog.sh
#!/bin/bash
echo "Provisioning script $0"
echo "Copy configuration files"
cp -R /vagrant/provision/server/netlog/etc/* /etc
restorecon -vR /etc
echo "Configure firewall"
firewall-cmd --add-port=514/tcp
firewall-cmd --add-port=514/tcp --permanent
echo "Start rsyslog service"
systemctl restart rsyslog
```

Скрипт provisioning для сервера

На клиенте выполним аналогичные действия: создадим структуру каталогов в /vagrant/provision, скопируем конфиг клиента и создадим установочный скрипт (рис. [-@fig:016]).

```
[dmmosharov@client.dmmosharov.net rsyslog.d]$ cd /vagrant/provision/cli
ent
[dmmosharov@client.dmmosharov.net client]$ mkdir -p /vagrant/provision/
client/netlog/etc/rsyslog.d
[dmmosharov@client.dmmosharov.net client]$ cp -R /etc/rsyslog.d/netlog-
client.conf /vagrant/provision/client/netlog/etc/rsyslog.d/
[dmmosharov@client.dmmosharov.net client]$ cd /vagrant/provision/client
[dmmosharov@client.dmmosharov.net client]$ touch netlog.sh
[dmmosharov@client.dmmosharov.net client]$ chmod +x netlog.sh
[dmmosharov@client.dmmosharov.net client]$ nano netlog.sh
```

Подготовка файлов для автоматизации клиента

В скрипте netlog.sh для клиента пропишем установку пакета lnav, копирование конфигурации rsyslog, восстановление контекста SELinux и перезапуск службы (рис. [-@fig:017]).

```
dmmosharov@client:/vagrant/provision/client – nano netlog.sh
 GNU nano 8.1                    netlog.sh
#!/bin/bash
echo "Provisioning script $0"
echo "Install needed packages"
dnf -y install lnav
echo "Copy configuration files"
cp -R /vagrant/provision/client/netlog/etc/* /etc
restorecon -vR /etc
echo "Start rsyslog service"
systemctl restart rsyslog
```

Скрипт provisioning для клиента

Наконец, откроем Vagrantfile и добавим в конфигурации сервера и клиента блоки vm.provision типа shell, указывающие на созданные нами скрипты netlog.sh (рис. [-@fig:018]).

```
  server.vm.provision "server netlog",
                      type: "shell",
                      preserve_order: true,
                      path: "provision/server/netlog.sh"
end

## Client configuration
config.vm.define "client", autostart: false do |client|
  client.vm.box = "rockylinux10"
  client.vm.hostname = 'client'

  client.vm.boot_timeout = 1440

  client.ssh.insert_key = false
  client.ssh.username = 'vagrant'
  client.ssh.password = 'vagrant'

  client.vm.network :private_network,
                    type: "dhcp",
                    virtualbox__intnet: true

  client.vm.provider :virtualbox do |virtualbox|
    virtualbox.customize ["modifyvm", :id, "--vrde", "on"]
    virtualbox.customize ["modifyvm", :id, "--vrdeport", "3392"]
  end

  client.vm.provision "client dummy",
                      type: "shell",
                      preserve_order: true,
                      path: "provision/client/01-dummy.sh"

  client.vm.provision "client routing",
                      type: "shell",
                      preserve_order: true,
                      run: "always",
                      path: "provision/client/01-routing.sh"
  client.vm.provision "client mail",
                      type: "shell",
                      preserve_order: true,
                      path: "provision/client/mail.sh"
  client.vm.provision "client ntp",
                      type: "shell",
                      preserve_order: true,
                      path: "provision/client/ntp.sh"

  client.vm.provision "client nfs",
                      type: "shell",
                      preserve_order: true,
                      path: "provision/client/nfs.sh"

  client.vm.provision "SMB client",
                      type: "shell",
                      preserve_order: true,
                      path: "provision/client/smb.sh"

  server.vm.provision "client netlog",
                      type: "shell",
                      preserve_order: true,
                      path: "provision/client/netlog.sh"
```

Настройка Vagrantfile

Выводы

В результате выполнения лабораторной работы были получены навыки использования журналов системных событий