Лабораторная работа №15

Презентация

Мосолов А.Д.

13 декабря 2025

Российский университет дружбы народов, Москва, Россия

Докладчик

- Мосолов Александр Денисович
- Студент, НПИбд02-23
- Российский университет дружбы народов
- 1132236128@pfur.ru

Цель

Получение навыков по работе с журналами системных событий.

Задание

1. Настройте сервер сетевого журналирования событий.
2. Настройте клиент для передачи системных сообщений в сетевой журнал на сервере.
3. Просмотрите журналы системных событий с помощью нескольких программ.
4. Напишите скрипты для Vagrant, фиксирующие действия по установке и настройке сетевого сервера журналирования.

Создание файла конфигурации на сервере

Начинаю настройку сервера сетевого журнала. Для этого перехожу в каталог конфигурации rsyslog и создаю файл `netlog-server.conf`, который будет содержать настройки для приема логов по сети.

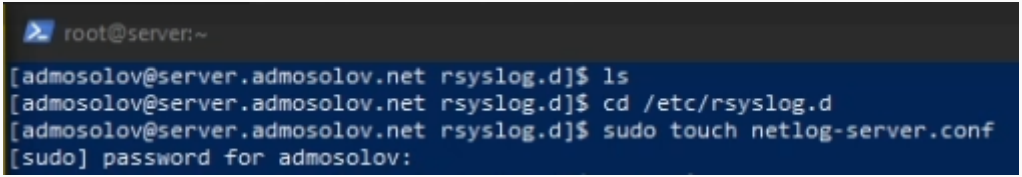


Рис. 1: Создание файла конфигурации на сервере

Настройка приема логов по TCP

Редактирую созданный файл конфигурации `/etc/rsyslog.d/netlog-server.conf`. Включаю модуль `imtcp` для поддержки протокола TCP и указываю прослушивание порта 514. Это стандартный порт для службы syslog при использовании надежного транспортного протокола.

Рис. 2: Настройка приема логов по TCP

Перезапуск службы rsyslog

После внесения изменений в конфигурацию необходимо перезапустить службу `rsyslog`, чтобы новые параметры вступили в силу.

```
[root@server.admosolov.net rsyslog.d]# systemctl restart rsyslog
```

Рис. 3: Перезапуск службы rsyslog

Проверка открытых портов

Проверяю, что служба действительно начала слушать нужный порт. Использую команду `lsof` с фильтрацией по TCP. Видно, что процесс `rsyslogd` прослушивает порт 514 (обозначен как `shell` в выводе `lsof` для этого порта).

```
named      116516 116774 libuv-wor          named   37u     IPv6          157573      0t0       TCP localhost:domain (LISTEN)
named      116516 116774 libuv-wor          named   38u     IPv6          157574      0t0       TCP localhost:domain (LISTEN)
named      116516 116774 libuv-wor          named   39u     IPv4          157575      0t0       TCP localhost:rndc (LISTEN)
named      116516 116774 libuv-wor          named   40u     IPv4          157576      0t0       TCP localhost:rndc (LISTEN)
named      116516 116774 libuv-wor          named   41u     IPv6          157577      0t0       TCP localhost:rndc (LISTEN)
named      116516 116774 libuv-wor          named   42u     IPv6          157578      0t0       TCP localhost:rndc (LISTEN)
named      116516 116774 libuv-wor          named   45u     IPv4          165371      0t0       TCP server.admosolov.net:domain (L
ISTEN)
named      116516 116774 libuv-wor          named   47u     IPv4          165372      0t0       TCP server.admosolov.net:domain (L
ISTEN)
sshd-sess 117764                             root    4u      IPv4          166268      0t0       TCP server.admosolov.net:ssh->_gat
eway:50162 (ESTABLISHED)
sshd-sess 117768                          vagrant    4u      IPv4          166268      0t0       TCP server.admosolov.net:ssh->_gat
eway:50162 (ESTABLISHED)
rsyslogd  117923                             root    4u      IPv4          166873      0t0       TCP *:shell (LISTEN)
rsyslogd  117923                             root    5u      IPv6          166874      0t0       TCP *:shell (LISTEN)
rsyslogd  117923 117926 in:imjour            root    4u      IPv4          166873      0t0       TCP *:shell (LISTEN)
rsyslogd  117923 117926 in:imjour            root    5u      IPv6          166874      0t0       TCP *:shell (LISTEN)
rsyslogd  117923 117927 in:imtcp             root    4u      IPv4          166873      0t0       TCP *:shell (LISTEN)
rsyslogd  117923 117927 in:imtcp             root    5u      IPv6          166874      0t0       TCP *:shell (LISTEN)
rsyslogd  117923 117928 w0/imtcp             root    4u      IPv4          166873      0t0       TCP *:shell (LISTEN)
```

Рис. 4: Проверка открытых портов

Настройка Firewall на сервере

Чтобы сервер мог принимать соединения извне, настраиваю межсетевой экран (firewall). Добавляю правило, разрешающее входящий трафик на порт 514/tcp, и закрепляю его как постоянное (`--permanent`).

```
[root@server.admosolov.net rsyslog.d]# firewall-cmd --add-port=514/tcp
success
[root@server.admosolov.net rsyslog.d]# firewall-cmd --add-port=514/tcp --permanent
success
[root@server.admosolov.net rsyslog.d]#
```

Рис. 5: Настройка Firewall на сервере

Создание конфигурации на клиенте

Перехожу к настройке клиента. На виртуальной машине клиента захожу под суперпользователем, перехожу в каталог конфигурации rsyslog и создаю файл `netlog-client.conf`.

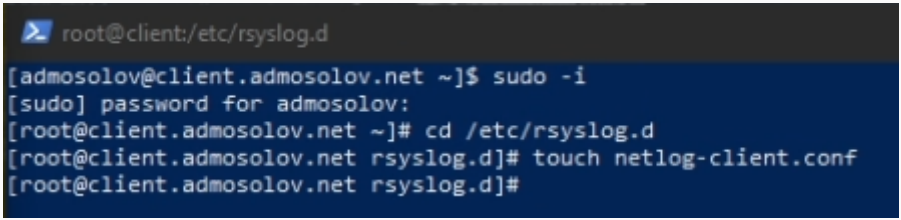


Рис. 6: Создание конфигурации на клиенте

Настройка пересылки логов на сервер

В файле конфигурации клиента прописываю правило пересылки всех логов (`*.*`) на удаленный сервер. Использую формат `@@server.admosolov.net:514` (две «собачки» означают использование протокола TCP).

Рис. 7: Настройка пересылки логов на сервер

Перезапуск rsyslog на клиенте

Для применения настроек перезапускаю службу журналирования на клиенте командой `systemctl restart rsyslog`.

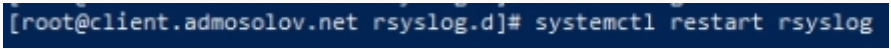


Рис. 8: Перезапуск rsyslog на клиенте

Мониторинг логов на сервере

Возвращаюсь на сервер для проверки получения логов. Запускаю команду `tail -f /var/log/messages` для просмотра сообщений в реальном времени. В логе видны записи о перезапуске службы.

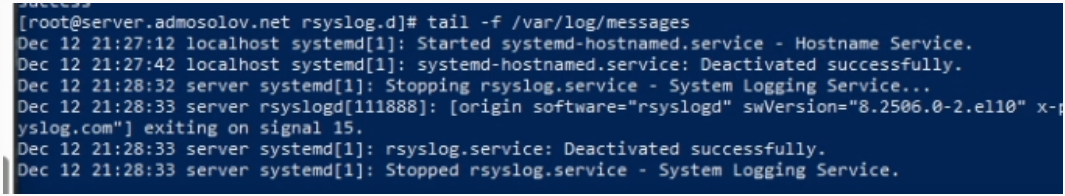


Рис. 9: Мониторинг логов на сервере

Отправка тестового сообщения

Чтобы убедиться в работоспособности сетевого журналирования, отправляю тестовое сообщение с клиента с помощью утилиты `logger`. Текст сообщения: “TEST 2: serv CHECK”.

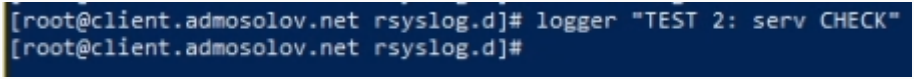


Рис. 10: Отправка тестового сообщения

Получение тестового сообщения на сервере

Проверяю журнал на сервере. Вижу, что сообщение, отправленное с хоста `client`, успешно записалось в файл `/var/log/messages` на сервере. Это подтверждает корректность настройки.

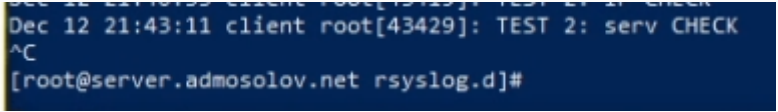


Рис. 11: Получение тестового сообщения на сервере

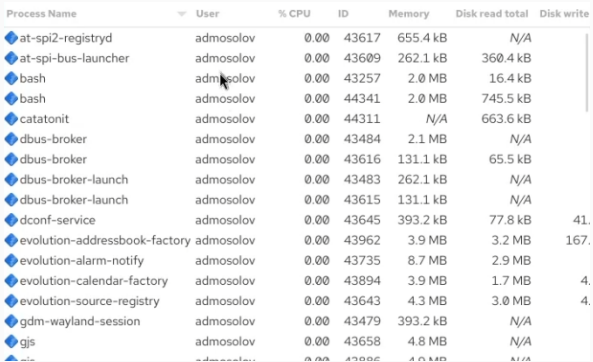
Запуск gnome-system-monitor

Перехожу к заданию по использованию графических утилит. На клиенте запускаю `gnome-system-monitor` (Системный монитор) для визуального наблюдения за процессами.

Рис. 12: Запуск gnome-system-monitor

Интерфейс системного монитора

В открывшемся окне системного монитора просматриваю список запущенных процессов, потребление ресурсов (CPU, память). Это позволяет отслеживать активность системы в графическом режиме.



Process Name	User	% CPU	ID	Memory	Disk read total	Disk write
at-spi2-registryd	admosolov	0.00	43617	655.4 kB	N/A	
at-spi-bus-launcher	admosolov	0.00	43609	262.1 kB	360.4 kB	
bash	admosolov	0.00	43257	2.0 MB	16.4 kB	
bash	admosolov	0.00	44341	2.0 MB	745.5 kB	
catatonit	admosolov	0.00	44311	N/A	663.6 kB	
dbus-broker	admosolov	0.00	43484	2.1 MB	N/A	
dbus-broker	admosolov	0.00	43616	131.1 kB	65.5 kB	
dbus-broker-launch	admosolov	0.00	43483	262.1 kB	N/A	
dbus-broker-launch	admosolov	0.00	43615	131.1 kB	N/A	
dconf-service	admosolov	0.00	43645	393.2 kB	77.8 kB	41.
evolution-addressbook-factory	admosolov	0.00	43962	3.9 MB	3.2 MB	167.
evolution-alarm-notify	admosolov	0.00	43735	8.7 MB	2.9 MB	
evolution-calendar-factory	admosolov	0.00	43894	3.9 MB	1.7 MB	4.
evolution-source-registry	admosolov	0.00	43643	4.3 MB	3.0 MB	4.
gdm-wayland-session	admosolov	0.00	43479	393.2 kB	N/A	
gjs	admosolov	0.00	43658	4.8 MB	N/A	

Рис. 13: Интерфейс системного монитора

Просмотр логов через journalctl

Также просматриваю журналы событий на сервере с помощью утилиты `journalctl`. Использую флаг `-f` для отслеживания новых записей. В логе отображаются действия менеджера пакетов PackageKit.

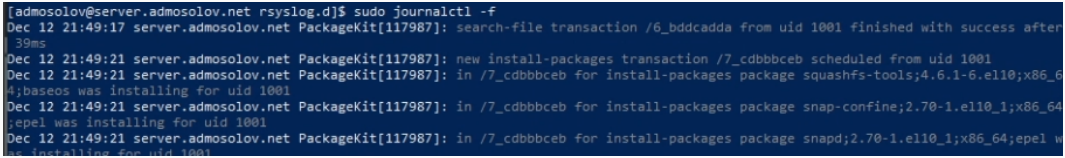


Рис. 14: Просмотр логов через journalctl

Подготовка каталогов для провижининга сервера

Приступаю к автоматизации процесса настройки через Vagrant. На сервере создаю структуру каталогов внутри `/vagrant/provision/server/` для хранения конфигурационных файлов и копирую туда текущий конфиг `netlog-server.conf`.

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

Рис. 15: Подготовка каталогов для провижининга сервера

Скрипт автоматизации для сервера

Создаю скрипт `netlog.sh` для сервера. В нем прописываю команды копирования конфигурации в `/etc/`, восстановления контекста безопасности SELinux, настройки firewall и перезапуска службы rsyslog.

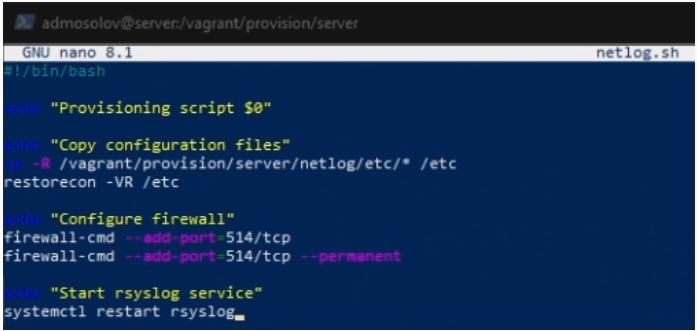


Рис. 16: Скрипт автоматизации для сервера

Подготовка каталогов для провижининга клиента

Аналогичные действия выполняю для клиента. Создаю структуру папок в `/vagrant/provision/client/` и копирую туда настроенный файл `netlog-client.conf`.

```
[root@client.admosolov.net rsyslog.d]# cd /vagrant/provision/client
[root@client.admosolov.net client]# mkdir -p /vagrant/provision/client/netlog/etc/rsyslog.d
[root@client.admosolov.net client]# cp /etc/rsyslog.d/netlog-client.conf /vagrant/provision/client/netlog/etc/rsyslog.d/
[root@client.admosolov.net client]# touch netlog.sh
[root@client.admosolov.net client]# chmod +x netlog.sh
[root@client.admosolov.net client]# nano netlog.sh
```

Рис. 17: Подготовка каталогов для провижининга клиента

Скрипт автоматизации для клиента

Создаю скрипт `netlog.sh` для клиента. Он включает установку утилиты `lnav` (для удобного просмотра логов), копирование конфигурации, настройку SELinux и перезапуск службы.

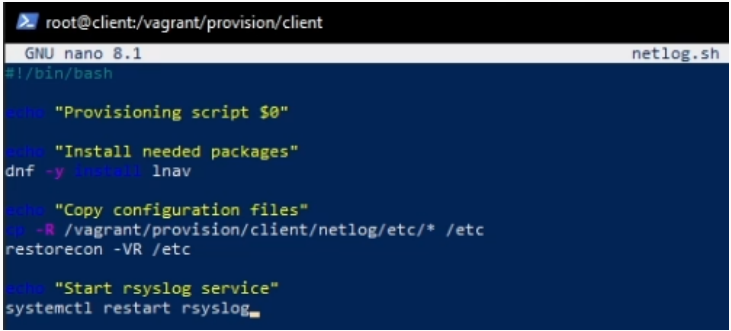


Рис. 18: Скрипт автоматизации для клиента

Настройка Vagrantfile для сервера

В завершение, вношу изменения в основной файл `Vagrantfile` для виртуальной машины `server`. Добавляю секцию provision типа `shell`, указывающую на созданный скрипт `provision/server/netlog.sh`.

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

Рис. 19: Настройка Vagrantfile для сервера

Настройка Vagrantfile для клиента

Также добавляю секцию provision в `Vagrantfile` для виртуальной машины `client`, указывая путь к скрипту `provision/client/netlog.sh`. Теперь настройка журналирования будет выполняться автоматически при развертывании машин.

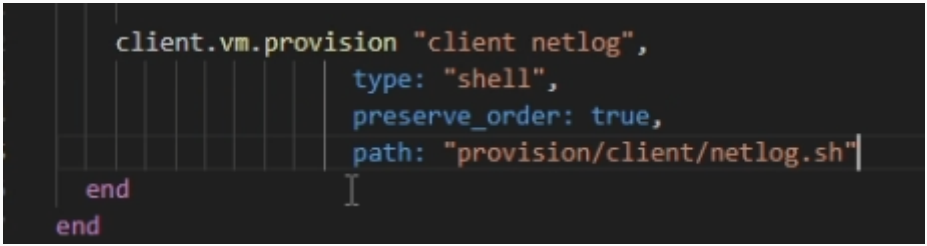


Рис. 20: Настройка Vagrantfile для клиента

Выводы

В ходе выполнения лабораторной работы я освоил принципы централизованного сбора логов в Linux. Я настроил сервер rsyslog для приема сообщений по протоколу TCP и клиент для их отправки. Также я научился анализировать логи с помощью `tail`, `journalctl` и графических утилит, а также написал скрипты автоматизации настройки окружения для Vagrant.