

# Сервер беспроводной сети



## Содержание

|   |    |
|---|----|
| Постановка задачи.....                                | 3  |
| Предварительные замечания.....                        | 4  |
| Аппаратное обеспечение.....                           | 4  |
| Установка системы и разметка диска.....               | 7  |
| Базовая настройка сервера.....                        | 8  |
| Установка доменного имени системы.....                | 8  |
| Отключение IPv6.....                                  | 8  |
| Отключение Magic SysRq.....                           | 9  |
| Создание пользователей.....                           | 10 |
| Настройка списков доступа.....                        | 11 |
| Отключение ненужных служб.....                        | 12 |
| Настройка списка репозиторийев.....                   | 14 |
| Установка вспомогательного ПО.....                    | 15 |
| Настройка сети.....                                   | 16 |
| Настройка сетевых интерфейсов.....                    | 16 |
| Маршрутизация.....                                    | 18 |
| Управление пропускной способностью сети.....          | 20 |
| Организация беспроводной сети.....                    | 21 |
| Управление беспроводной сетевой картой.....           | 22 |
| Настройка программной точки доступа.....              | 25 |
| Автоматическая смена пароля точки доступа.....        | 27 |
| Защита от падений (автоматическое возобновление)..... | 28 |
| Проверка работоспособности и устойчивости.....        | 28 |
| Служба DNS/DHCP.....                                  | 31 |
| Настройка.....  | 31 |
| Защита от падений (автоматическое возобновление)..... | 34 |
| Проверка работоспособности и устойчивости.....        | 35 |
| Служба точного времени (ntp).....                     | 36 |
| Настройка.....  | 36 |
| Защита от падений (автоматическое возобновление)..... | 40 |
| Проверка работоспособности и устойчивости.....        | 40 |
| Установка и настройка ssh.....                        | 41 |
| Настройка.....  | 41 |

|   |    |
|---|----|
| Защита от падений (автоматическое возобновление).....             | 43 |
| Проверка работоспособности и устойчивости.....                    | 43 |
| О дополнительных службах.....                                     | 45 |
| Защита от подбора паролей (fail2ban).....                         | 45 |
| Ротация журналов (logrotate).....                                 | 45 |
| Служба точного времени (ntp).....                                 | 45 |
| Почтовая служба.....  | 46 |
| Прикладное программное обеспечение.....                           | 47 |
| Комплект прикладных программ и графическое окружение.....         | 47 |
| Автоматическое восстановление настроек интерфейса и программ..... | 48 |
| Отображение текущего пароля.....                                  | 49 |
| Автоматическое выключение системы.....                            | 49 |
| Установка и настройка сетевого фильтра (nftables).....            | 50 |
| Сетевое окружение и потенциальные угрозы.....                     | 50 |
| Структура сетевого фильтра.....                                   | 51 |
| Запуск и проверка сетевого фильтра.....                           | 52 |
| Завершение установки.....   | 63 |
| Использование сервера.....  | 63 |
| Порядок входа в систему.....                                      | 63 |
| Мониторинг состояния аппаратных средств.....                      | 63 |
| Смена внешнего сервера доменных имен.....                         | 65 |
| Получение текущего списка клиентских устройств.....               | 66 |
| Замечания по подключению клиентов.....                            | 67 |

## Постановка задачи

Требуется создать на аппаратной базе ноутбука Lenovo G560 сервер беспроводной подсети школы согласно следующей схеме сети:

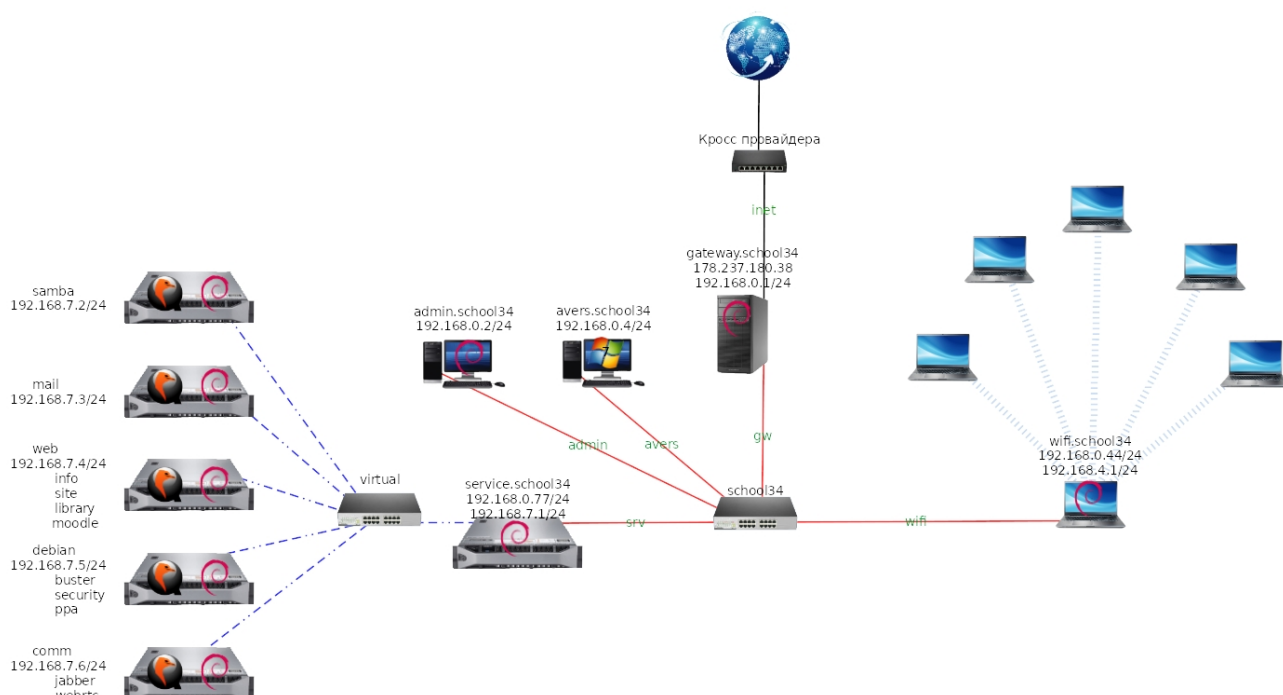


Рисунок 1: Фрагмент схемы сети

На данной схеме отображена видимая серверу беспроводной сети (server.wifi.school34) часть сети школы с указанием доменных имён и ipv4-адресов некоторых узлов. Большая часть сети скрыта, т. к. не должна быть видима ни клиентам беспроводной сети, ни её серверу. На рисунке представлен также сервер виртуализации, обслуживающий домен service.school34. На базе этого сервера развернут ряд виртуальных машин, которые предоставляют разные сетевые сервисы клиентам сети школы. В ядре сети школы следует отметить сервер АИС «Аверс» и компьютер системного администратора.

Сервер должен предоставлять клиентам беспроводную среду передачи с использованием шифрованного канала и регулярной сменой пароля для доступа к нему.

Сервер беспроводной сети должен предоставлять услуги DHCP для клиентских устройств на своём беспроводном интерфейсе, поддерживая одновременное подключение до 45 клиентов. При этом адреса им должны выдаваться в подсети 192.168.4.0/24 с учётом того, что адрес 192.168.4.1 занят самим сервером.

Сервер должен предоставлять клиентам услуги службы доменных имён, пересылая запросы о домене service.school34 (192.168.7.0/24) к server.service.school34, о домене school34 – к gateway.school34, а все остальные запросы – ко внешнему DNS-серверу. При этом должна быть предусмотрена возможность оперативной смены нефилтующего (YandexDNS) внешнего сервера на фильтрующий (SkyDNS) и обратно.

Сервер должен предоставлять клиентам услуги службы точного времени, используя для собственной синхронизации gateway.school34, который является также сервером точного времени уровня школы и имеет псевдоним ntp.school34.

Сервер должен быть доступен для удалённого управления только с компьютера системного администратора (admin.school34).

Сервер должен быть способен выступать в роли рабочей станции, для чего он должен быть оснащён соответствующим программным обеспечением: окружение рабочего стола (Plasma5), комплект офисного ПО (LibreOffice, Okular, архиваторы и интерфейс к ним, Интернет-браузеры, растровые и векторные графические редакторы и т. д.).

Должны быть приняты разумные меры для защиты и обеспечения бесперебойной работы сервера. Также должны быть приняты меры к обеспечению эргономичности графического рабочего окружения.

## Предварительные замечания

Во фрагментах терминальных сеансов и при указании выполняемых команд используются следующие соглашения:

Выполняемые при настройке системы команды выделены жирным шрифтом, отклик системы или выдержки из конфигурационных файлов и сценариев такого выделения не имеют. Все команды приводятся с полным отображением приглашения командной строки, в котором отражается пользователь, от имени которого выполняется команда, и текущий рабочий каталог. Оба этих фактора значимы в большинстве приведённых команд и листингов.

## Аппаратное обеспечение

В качестве платформы для реализации сервера беспроводной подсети выбран ноутбук Lenovo G560 из состава уже имеющегося в организации машинного парка. Далее в сокращённом виде приведены сведения об аппаратуре ноутбука:

```
root@server:/# lshw
server
description: Notebook
product: 20042 (Calpella_CRB)
vendor: LENOVO
version: Lenovo G560
serial: 2715387103105
width: 64 bits
capabilities: smbios-2.6 dmi-2.6 smp vsyscall32
...
*-core
description: Motherboard
product: Base Board Product Name
vendor: LENOVO
physical id: 0
version: Base Board Version
serial: CB09094437
slot: Base Board Chassis Location
*-firmware
description: BIOS
vendor: LENOVO
physical id: 0
```

```

version: 29CN40WW(V2.17)
...
*-memory
  description: System Memory
  physical id: 19
  slot: System board or motherboard
  size: 4GiB
  *-bank:0
    description: SODIMM DDR3 Synchronous 1067 MHz (0,9 ns)
    product: RMT3020EF48E8W1333
    physical id: 0
    serial: 100EF037
    slot: DIMM0
    size: 2GiB
    width: 64 bits
    clock: 1067MHz (0.9ns)
  *-bank:1
    description: DIMM DDR3 Synchronous 1067 MHz (0,9 ns) [empty]
...
  *-bank:2
    description: SODIMM DDR3 Synchronous 1067 MHz (0,9 ns)
    product: HMT325S6BFR8C-H9
    physical id: 2
    serial: 2420BFD8
    slot: DIMM1
    size: 2GiB
    width: 64 bits
    clock: 1067MHz (0.9ns)
  *-bank:3
    description: DIMM DDR3 Synchronous 1067 MHz (0,9 ns) [empty]
...
*-cpu
  description: CPU
  product: Intel(R) Core(TM) i3 CPU          M 370  @ 2.40GHz
  vendor: Intel Corp.
  physical id: 2c
  bus info: cpu@0
  version: Intel(R) Core(TM) i3 CPU          M 370  @ 2.40GHz
  slot: CPU
  size: 1369MHz
  capacity: 2400MHz
  width: 64 bits
  clock: 1066MHz
  capabilities: lm fpu fpu_exception wp vme de pse tsc msr pae ...
  configuration: cores=2 enabledcores=2 threads=4
...
*-pci:0
  description: Host bridge
...
  *-display
    description: VGA compatible controller
    product: GT218M [GeForce 310M]
    vendor: NVIDIA Corporation
...
  *-multimedia
    description: Audio device
    product: High Definition Audio Controller
    vendor: NVIDIA Corporation
...
  *-communication
    description: Communication controller
    product: 5 Series/3400 Series Chipset HECI Controller

```

```

...
*-multimedia
  description: Audio device
  product: 5 Series/3400 Series Chipset High Definition Audio
  vendor: Intel Corporation
...
*-pci:2
  description: PCI bridge
  product: 5 Series/3400 Series Chipset PCI Express Root Port 2
...
*-network
  description: Wireless interface
  product: AR9285 Wireless Network Adapter (PCI-Express)
  vendor: Qualcomm Atheros
...
*-pci:3
  description: PCI bridge
  product: 5 Series/3400 Series Chipset PCI Express Root Port 3
...
*-network
  description: Ethernet interface
  product: RTL8101/2/6E PCI Express Fast/Gigabit Ethernet controller
  vendor: Realtek Semiconductor Co., Ltd.
  physical id: 0
  bus info: pci@0000:07:00.0
  logical name: enp7s0
  version: 02
  serial: b8:70:f4:29:e5:83
  size: 100Mbit/s
  capacity: 100Mbit/s
...
*-sata
  description: SATA controller
  product: 5 Series/3400 Series Chipset 4 port SATA AHCI Controller
...
*-disk
  description: ATA Disk
  product: ST9320325AS
  physical id: 0
  bus info: scsi@0:0.0.0
  logical name: /dev/sda
  version: LVM1
  serial: 6VDCABKR
  size: 298GiB (320GB)
  capabilities: partitioned partitioned:dos
...

```

В качестве комментариев к приведённым сведениям необходимо отметить следующее:

Ноутбук обладает вполне достаточными для выполнения поставленных задач центральным процессором, объёмом и характеристиками оперативной памяти, жёстким диском, видеоадаптером и прочими системами.

Отдельно следует отметить сетевые карты устройства: проводной сетевой интерфейс поддерживает только Fast Ethernet, отсутствие поддержки Gigabit Ethernet – не лучшая черта устройства, однако такой сетевой карты вполне достаточно с учётом ограничений на пропускную способность, накладываемых постановкой задачи.

Аналогичное замечание можно сделать и о беспроводной сетевой карте: она далеко не новая, уровень её сигнала в режиме точки доступа достаточен для покрытия совсем

небольшой площади. Однако это нельзя считать недостатком – даже если бы она имела более широкое покрытие, его пришлось бы ограничивать в силу характера функционирования сети, характера самой организации и постановки задачи.

Подробное описание характеристик беспроводного адаптера приведено далее в разделе «Организация беспроводной сети».

Также следует отметить, что при закрытии крышки ноутбука (и соответствующих настройках операционной системы и рабочей среды) не происходит отключения беспроводной сетевой карты.

## Установка системы и разметка диска

Установка системы выполнена с графической оболочкой Plasma 5 и стандартным набором программного обеспечения к ней, а также со стандартными системными утилитами и службой ssh.

Жёсткий диск разбит на разделы следующим образом (в выводе ниже опущены строки, не касающиеся жёсткого диска):

```
root@server:/# /usr/sbin/fdisk -l /dev/sda
Disk /dev/sda: 298,1 GiB, 320072933376 bytes, 625142448 sectors
Disk model: ST9320325AS
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x4bc7d946
```

| Device    | Boot | Start     | End       | Sectors   | Size   | Id | Type                 |
|-----------|------|-----------|-----------|-----------|--------|----|----------------------|
| /dev/sda1 | *    | 2048      | 58593279  | 58591232  | 28G    | 83 | Linux                |
| /dev/sda2 |      | 58593280  | 117186559 | 58593280  | 28G    | 83 | Linux                |
| /dev/sda3 |      | 117186560 | 124999679 | 7813120   | 3,7G   | 82 | Linux swap / Solaris |
| /dev/sda4 |      | 124999680 | 625141759 | 500142080 | 238,5G | 83 | Linux                |

```
root@server:/# /sbin/parted -l /dev/sda
Model: ATA ST9320325AS (scsi)
Disk /dev/sda: 320GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:
```

| Number | Start  | End    | Size   | Type    | File system     | Flags |
|--------|--------|--------|--------|---------|-----------------|-------|
| 1      | 1049kB | 30,0GB | 30,0GB | primary | ext4            | boot  |
| 2      | 30,0GB | 60,0GB | 30,0GB | primary | ext4            |       |
| 3      | 60,0GB | 64,0GB | 4000MB | primary | linux-swaps(v1) |       |
| 4      | 64,0GB | 320GB  | 256GB  | primary | ext4            |       |

```
root@server:/# mount -l
/dev/sda1 on / type ext4 (rw,relatime,errors=remount-
ro,jqfmt=vfsv0,usrjquota=aquota.user) [rootfs]
/dev/sda2 on /var type ext4
(rw,nosuid,nodev,relatime,jqfmt=vfsv0,usrjquota=aquota.user) [var]
/dev/sda4 on /home type ext4
(rw,nosuid,nodev,relatime,jqfmt=vfsv0,usrjquota=aquota.user) [home]
```

```
root@server:/# df -h
Файловая система  Размер  Использовано  Дост  Использовано%  Смонтировано в
/dev/sda1          28G      8,0G      19G          31% /
```

|           |      |      |      |    |       |
|-----------|------|------|------|----|-------|
| /dev/sda2 | 28G  | 991M | 25G  | 4% | /var  |
| /dev/sda4 | 234G | 275M | 222G | 1% | /home |

Такая схема разметки диска выбрана по следующим соображениям:

DOS-таблица является наиболее простой и достаточной для рассматриваемого сервера, особенно с учётом года выпуска аппаратной части. На раздел подкачки выделено 4 Гб, остальное дисковое пространство разбито на три раздела, благодаря чему каталог изменяемых данных /var и домашние каталоги пользователей размещены в отдельных файловых системах. Это сделано в рамках базовых мер по защите сервера. Большая часть диска отведена именно под домашние каталоги пользователей, тогда как для остальных компонентов системы выделено минимально достаточное (с некоторым запасом) дисковое пространство.

## Базовая настройка сервера

В этом разделе описаны меры, применимые ко многим подобным серверам. В частности рассмотрены способы минимизации числа служб, настройки сетевой подсистемы и некоторых систем безопасности.

### Установка доменного имени системы

С учётом того, что в системе не предполагается использование протокола IPv6, для настройки полного имени домена следует сначала заполнить файл **/etc/hosts**, затем внести имя системы без доменной части в файл **/etc/hostname** и выполнить команду для смены имени хоста в текущей сессии. Все необходимые команды, а также содержимое конфигурационный файлов представлены в нижеследующем фрагменте терминального сеанса. Последняя команда в нём демонстрирует настроенное полное доменное имя системы (FQDN):

```
root@server:/# cat /etc/hosts
127.0.0.1      localhost.localdomain localhost
127.0.1.1      server.wifi.school34 server
root@server:/# cat /etc/hostname
server
root@server:/# hostname -f
server.wifi.school34
```

### Отключение IPv6

Отключение поддержки IPv6 ядром системы может быть выполнено как для текущей сессии, так и на постоянной основе. Для отключения на постоянной основе следует привести содержимое файла **/etc/sysctl.conf** к виду, не противоречащему следующим строкам в нём:

```
#Disabling Ipv6
net.ipv6.conf.all.disable_ipv6=1
net.ipv6.conf.default.disable_ipv6=1
net.ipv6.conf.lo.disable_ipv6=1
```

Для отключения только в текущей сессии достаточно выполнить команды:

```
root@server:/# echo '1' > /proc/sys/net/ipv6/conf/all/disable_ipv6
root@server:/# echo '1' > /proc/sys/net/ipv6/conf/default/disable_ipv6
```



```
root@server:/# echo '1' > /proc/sys/net/ipv6/conf/lo/disable_ipv6
```

Для проверки успешности отключения следует перезапустить сетевую подсистему командой

```
root@server:/# service networking restart
```

и в выводе команды

```
root@server:/# ip address show
```

убедиться в отсутствии назначенных адресов IPv6:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
2: enp7s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group
default qlen 1000
    link/ether b8:70:f4:29:e5:83 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.44/24 brd 192.168.0.255 scope global enp7s0
        valid_lft forever preferred_lft forever
3: wlp6s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group
default qlen 1000
    link/ether d0:df:9a:6a:89:18 brd ff:ff:ff:ff:ff:ff
    inet 192.168.4.1/24 brd 192.168.4.255 scope global wlp6s0
        valid_lft forever preferred_lft forever
```

*Замечание:* здесь представлен вывод при полностью настроенной сетевой подсистеме. Описание таких настроек приведено ниже, здесь достаточно отметить лишь отсутствие у интерфейсов ipv6-адресов.

## Отключение Magic SysRq

Отключение Magic SysRq может показаться избыточным или даже вредным: так, например, в случае сбоя системы не будет возможности её аварийно остановить или сбросить содержимое буферов на диск. Однако следует помнить, что сервер не предназначен для хранения важных данных. В то же время, в силу физической доступности сервера пользователям, есть вероятность того, что некий злоумышленник выполнит нажатие подобной комбинации клавиш. Это может привести к остановке сервера и отказам в обслуживании клиентов. При этом защита организовывается настолько просто, что пренебрегать ей не рационально.

Для отключения «волшебных» сочетаний в текущей сессии достаточно команды:

```
root@debian:/# echo '0' > /proc/sys/kernel/sysrq
```

Для отключения на постоянной основе достаточно привести файл **/etc/sysctl.conf** к виду, не противоречащему следующей строке из него

```
kernel.sysrq=0
```

## Создание пользователей

Далее при описании службы защищённого удалённого входа ssh указано, что вход в систему удалённо от имени суперпользователя заблокирован. Чтобы всё же иметь возможность удалённого входа, следует создать обычного пользователя, от имени которого и будет осуществляться удалённый вход. Также от имени этого пользователя будет загружаться сессия графической пользовательской оболочки, предоставляющая возможность использовать сервер как рабочую станцию. После открытия сессии в системе под такой учётной записью появляется, при необходимости, возможность повысить свои полномочия при помощи su. Ограничения на системные ресурсы для этого пользователя на данный момент не вводятся, однако по опыту эксплуатации системы это может быть сделано в дальнейшем.

Создание такого пользователя (пусть его логин – guest) можно выполнить как на этапе установки системы, так и после, например, при помощи команды

```
root@debian:/# /usr/sbin/adduser guest
```

Кроме создания пользователя, необходимо обратить внимание на несколько аспектов его работы в системе.

Во-первых, по умолчанию вновь созданный домашний каталог этого пользователя доступен для чтения всем в системе. В случае, если это неприемлемо, исправить ситуацию можно при помощи команды

```
root@debian:/# chmod 0750 /home/guest/
```

оставив, таким образом, полный доступ к каталогу для самого пользователя и доступ на чтение для членов его группы (коих кроме него самого в системе всё равно нет). Для всех остальных доступ окажется закрыт.

Вторым аспектом является использование sudo. Эта команда позволяет выполнять команды с административными привилегиями, не входя в систему от имени суперпользователя. Такой подход таит в себе некоторую опасность: злоумышленник, зная логин и пароль такого пользователя может получить весьма обширные права в системе (если политики sudo настроены) или практически неограниченные, если настройки sudo приняты по умолчанию. Для решения этой проблемы проще всего (и наиболее оптимально на подобном сервере с одним пользователем и минимальным количеством сервисов) воспользоваться следующим подходом: не использовать sudo вообще, вынуждая пользователя для повышения своих прав в системе вводить (и, соответственно знать) пароль суперпользователя.

Таким образом, чтобы лишить пользователя возможности использования sudo достаточно исключить его из группы wheel, если он в ней состоит. Сделать это можно командой usermod, оставив пользователя guest только в своей (одноимённой) группе. Убедиться в успешности изменений можно выполнив от имени этого пользователя строго после повторного входа в систему команду groups. Далее приведён фрагмент терминального сеанса, демонстрирующий все эти операции:

```
root@server:/# /usr/sbin/usermod -G guest guest
root@server:/# su guest
guest@server:/$ groups
guest
guest@server:/$ exit
```

```
exit
root@server:/#
```

Также следует установить дисковые квоты для пользователя на файловых системах, переполнение которых может быть чувствительно для работоспособности системы. В рассматриваемой системе квоты настроены следующим образом:

```
root@server:/# setquota -u guest 10G 10G 0 0 /
root@server:/# setquota -u guest 240G 240G 0 0 /home
```

При таких квотах пользователю guest разрешено использовать не более 10 Гб из 30 имеющихся в корневой файловой системе (в которой расположены каталоги /tmp и /var/tmp) и не более 240 Гб из 256 имеющихся в разделе домашних каталогов.

Убедиться в том, что квоты установлены, можно в выводе команды

```
root@server:/# quota -s --show-mntpoint guest
Disk quotas for user guest (uid 1001):
    Filesystem    space   quota   limit   grace   files   quota   limit   grace
/dev/sda1 /         600K  10240M  10240M           151      0      0
/dev/sda4 /home    242M   240G   240G           4254     0      0
```

Завершающим этапом создания пользователя guest следует считать обеспечение возможности его беспарольного локального входа в графическое окружение рабочего стола системы, что подразумевается характером использования системы. Необходимые для этого действия описаны далее в разделе «Прикладное программное обеспечение».

## Настройка списков доступа

Многие демоны используют файлы **/etc/hosts.allow** и **/etc/hosts.deny** как источники информации о том, кому разрешено пользоваться услугами этих демонов. Эти конфигурационные файлы являются частью механизма tcp wrappers. Кроме того, в системе может быть установлен демон tcpd, который сам выполняет такую проверку, прежде чем передать сетевой пакет, инициирующий соединение, целевому демону.

Чтобы выяснить, использует ли демон некоторой службы этот механизм, как правило достаточно убедиться, что исполняемый файл демона собран с использованием библиотеки libwrap. Следующий фрагмент терминального сеанса демонстрирует, что демон sshd использует эту библиотеку, а dnsmasq и ntpd – нет:

```
root@server:/# ldd /sbin/ntpd | grep libwrap
root@server:/# ldd /sbin/dnsmasq | grep libwrap
root@server:/# ldd /sbin/sshd | grep libwrap
libwrap.so.0 => /lib/x86_64-linux-gnu/libwrap.so.0 (0x00007fa52ea65000)
```

Проверить, установлен ли в системе демон tcpd, можно командой

```
root@server:/# dpkg -l | grep tcpd
```

Как видно из пустого отклика команды, этот демон не установлен.

В общем случае такая проверка выполняется файерволом, а механизм tcp wrapper считается устаревшим и уступившим место как раз сетевым экранам. Тем не менее, учитывая простоту настройки и нетребовательность к ресурсам, имеет смысл внести в указанные конфигурационные файлы записи для всех используемых системой демонов,

оставив, тем самым, решение по использованию этого механизма разработчикам демонов и сборщикам дистрибутива. Иными словами, если ими будет принято решение включить поддержку в новой версии ПО, то настраиваемая система будет автоматически готова к таким изменениям. Кроме того, никакое ПО не застраховано от ошибок разработчиков и уязвимостей. Использование этого механизма может оказаться дополнительным рубежом защиты при компрометации сетевого фильтра.

В данной системе не имеет смысла установка `tcpd`, т. к. `ssh` использует этот механизм самостоятельно, а остальные демоны просто не принимают подключений на внешнем (проводном) интерфейсе.

Зная, что подключаться по протоколу `ssh` разрешено только с компьютера системного администратора, имеющего адрес `192.168.0.2`, а по протоколам `ntr` и `dns/dhcp` – только с внутреннего (беспроводного) интерфейса, следует привести файл `/etc/hosts.allow` к следующему виду (комментарии для краткости опущены):

```
ntpd : 192.168.4.0/24 127.0.0.1
dnsmasq : 192.168.4.0/24 127.0.0.1
sshd : 192.168.0.2
```

В свою очередь файл `/etc/hosts.deny` запрещает любые другие подключения к серверу (комментарии также не приведены):

```
ALL : ALL
```

После перезапуска служб убедиться в том, что `ssh` не принимает соединений с посторонних хостов, можно перезапустив эту службу командой

```
root@debian:/# systemctl restart sshd
```

и попытавшись подключиться по `ssh` с любого отличного от сервера подсети хоста. Разумеется, на момент эксперимента файрвол не должен блокировать такое подключение.

Безусловно настраивать подобные параметры безопасности следует до подключения сервера к сети, а окончательно проверять – сразу после, по возможности защищая проверяемый сегмент сети иными средствами (например файрволом вышестоящего сервера организации).

## Отключение ненужных служб

Соблюдая правило необходимой достаточности, следует отключить и/или удалить все неиспользуемые службы, например: `rsh`, `telnet`, `rpcbind`. Проверить, установлены ли эти службы, можно одной из или сочетанием следующих (и, возможно, некоторых дополнительных) команд:

```
root@server:/# dpkg -l | grep rpcbind
root@server:/# which rsh
root@server:/# ss -l | grep telnet
```

Ни один из этих способов не универсален в том смысле, что однозначно показывает наличие или отсутствие той или иной службы. Как пример можно привести следующий фрагмент терминального сеанса:

```

root@server:/# which rsh
/usr/bin/rsh
root@server:/# ss -Htpl
LISTEN 0 32 127.0.0.1:domain 0.0.0.0:* users:(("dnsmasq",pid=745,fd=9))
LISTEN 0 32 192.168.4.1:domain 0.0.0.0:* users:(("dnsmasq",pid=745,fd=7))
LISTEN 0 128 192.168.0.44:ssh 0.0.0.0:* users:(("sshd",pid=738,fd=3))
LISTEN 0 5 127.0.0.1:ipp 0.0.0.0:* users:(("cupsd",pid=613,fd=7))
LISTEN 0 50 *:1716 *: users:(("kdeconnectd",pid=946,fd=12))
root@server:/# dpkg -l | grep rsh
ii hershey-font-gnuplot 0.1-1+b1 amd64 Hershey vector fonts renderer for gnuplot
ii hershey-fonts-data 0.1-1 all Hershey vector fonts collection
ii libhersheyfont0 0.1-1+b1 amd64 Hershey vector fonts shared library
root@server:/# ls -l /usr/bin | grep rsh
-rwxr-xr-x 1 root root 10512 дек 3 2013 hershey-font-gnuplot
lrwxrwxrwx 1 root root 21 сен 17 20:38 rsh -> /etc/alternatives/rsh
root@server:/# ls -l /etc/alternatives | grep rsh
lrwxrwxrwx 1 root root 12 сен 17 20:38 rsh -> /usr/bin/ssh
lrwxrwxrwx 1 root root 28 сен 17 20:38 rsh.1.gz -> /usr/share/man/man1/ssh.1.gz
root@server:/#

```

Разбирая приведённый фрагмент, можно отметить следующее:

Первая команда показывает, что в системе присутствует исполняемый файл rsh, однако вторая команда показывает, что демон rsh не прослушивает никаких tcp-портов. Следующая команда сообщает о том, что в системе установлено 3 пакета, в названии которых упоминается rsh, однако по их названиям очевидно, что отношения к rsh эти пакеты не имеют. При просмотре в подробном режиме свойств исполняемого файла выясняется, что он является символической ссылкой на другой файл, который также является символической ссылкой на исполняемый файл ssh. Таким образом, в данной системе rsh не установлен, вместо этого создан псевдоним для ssh.

Среди прочего в составе графического рабочего окружения в системе были установлены пакеты rpcbind и nfs-common. Т.к. развертывание nfs-сервера на рассматриваемой машине не входит в постановку задачи, как и подключение этой машины к другим серверам nfs, то оба этих пакета могут быть удалены командой:

```
root@server:/# apt-get purge rpcbind
```

Тем самым будут освобождены системные ресурсы, занимаемые ими, а также будут закрыты неиспользуемые порты.

Также следует удалить службу рабочего стола для подключения мобильных устройств (KDE Connect) и систему Avahi, обеспечивающую обнаружение сервисов в локальной сети. Avahi бесполезна на рассматриваемой машине, поскольку именно эта машина и является сервером беспроводной подсети, предоставляющим все необходимые службы, а с точки зрения ядра сети сервер беспроводной подсети хотя и является клиентом, но обо всех нужных ему службам уведомлен заранее.

Удалить ненужные службы можно командами (незначащий вывод команд сокращён):

```
root@server:/# apt-get purge kdeconnect
```

Чтение списков пакетов... Готово

...

Следующий пакет устанавливался автоматически и больше не требуется:

```
libfakekey0
```

Для его удаления используйте «apt autoremove».

```
root@server:/# apt-get purge avahi-daemon
```

```
root@server:/# apt-get purge avahi-autoipd
root@server:/# apt autoremove
```

После перезапуска сессии рабочего стола можно убедиться, что система теперь прослушивает только порты, необходимые ей для выполнения поставленных задач:

```
root@server:/# ss -tulp
Netid State Recv-Q Send-Q Local Address:Port
udp UNCONN 0 0 0.0.0.0:40203 users:(("dnsmasq",pid=691,fd=10))
udp UNCONN 0 0 192.168.4.1:domain users:(("dnsmasq",pid=691,fd=6))
udp UNCONN 0 0 127.0.0.1:domain users:(("dnsmasq",pid=691,fd=8))
udp UNCONN 0 0 0.0.0.0:bootps users:(("dnsmasq",pid=691,fd=4))
udp UNCONN 0 0 192.168.0.44:ntp users:(("ntpd",pid=666,fd=18))
udp UNCONN 0 0 192.168.4.1:ntp users:(("ntpd",pid=666,fd=22))
udp UNCONN 0 0 127.0.0.1:ntp users:(("ntpd",pid=666,fd=17))
udp UNCONN 0 0 0.0.0.0:ntp users:(("ntpd",pid=666,fd=16))
udp UNCONN 0 0 0.0.0.0:ipp users:(("cups-browsed",pid=489,fd=7))
tcp LISTEN 0 32 192.168.4.1:domain users:(("dnsmasq",pid=691,fd=7))
tcp LISTEN 0 32 127.0.0.1:domain users:(("dnsmasq",pid=691,fd=9))
tcp LISTEN 0 128 192.168.0.44:ssh users:(("sshd",pid=670,fd=3))
tcp LISTEN 0 5 127.0.0.1:ipp users:(("cupsd",pid=479,fd=6))
```

При этом следует отметить UDP-порт 40203, открытый самим dnsmasq для обращений к вышестоящим dns-серверам. Номер этого порта меняется при перезапуске системы, а сам dnsmasq не принимает на нём клиентских запросов.

Также следует отметить службу управления принтерами CUPS и её демонов cups-browsed и cupsd. На момент написания этого документа не предполагается подключать к рассматриваемой системе принтеры или, тем более, предоставлять клиентам сервис печати. Однако такое расширение функционала весьма вероятно, поэтому работа службы не блокируется, тем более, что с настройками по умолчанию служба (cupsd) печати прослушивает лишь локальный петлевой интерфейс. Демон cups-browsed, в свою очередь, лишь прослушивает сеть в ожидании широковещательной рассылки от серверов печати, готовых предоставить свои услуги. В таких условиях, достаточно лишь закрыть порт службы печати при помощи межсетевого экрана, что и будет продемонстрировано далее.

## Настройка списка репозиторийев

Сразу после установки системы для возможности установки дополнительного программного обеспечения следует заполнить список репозиторийев, используемых системой, зарегистрировать публичные части ключей локальных репозиторийев и настроить проводной сетевой адаптер системы. Описание настройки сетевых интерфейсов приведено ниже для сохранения стройности и целостности изложения. Нет препятствий тому, чтобы настроить список репозиторийев и установить дополнительное программное обеспечение после настройки проводной сети.

Конфигурационный файл `/etc/apt/sources.list`, содержащий список репозиторийев, следует привести к виду:

```
deb http://debian.service.school34/buster/ buster main contrib non-free
deb http://debian.service.school34/security/ buster/updates main contrib
deb http://debian.service.school34/ppa/ buster main contrib
```

Затем необходимо получить публичную часть ключа локального репозитория (ppa), например, загрузив из него же файл repository\_key.asc командой:

```
root@server:/# cd /tmp && wget http://debian.service.school34/ppa/repository_key.asc
```

Далее этот файл следует зарегистрировать в системе управления пакетами командой:

```
root@server:/tmp# apt-key add repository_key.asc
```

После этого уже возможно обновить сведения о пакетах, хранящихся в указанных репозиториях:

```
root@server:/tmp# apt-get update
```

Импортировать ключи остальных репозиториев не нужно, т. к. они являются зеркалами официальных репозиториев и установлены вместе с системой.

## Установка вспомогательного ПО

Весьма полезными при управлении сервером могут оказаться файловый менеджер Midnight Commander (пакет mc) и утилита tree. Для их установки достаточно выполнить команду:

```
root@server:/# apt-get install mc tree
```

Для контроля состояния сервера весьма полезны средства просмотра данных от аппаратных датчиков. Эти средства входят в состав пакета lm-sensors, который необходимо установить, а затем провести процесс определения имеющихся в системе датчиков. Это можно сделать, выполнив следующие команды и последовав появившимся при этом инструкциям:

```
root@server:/# apt-get install lm-sensors
root@server:/# sensors-detect
```

После чего данные этих датчиков можно получить при помощи команды

```
root@server:/# sensors
```

Причём эта команда может быть выполнена даже с правами непривилегированного пользователя.

Ещё одним средством контроля состояния системы является технология S.M.A.R.T – оценка состояния жёсткого диска (или иного носителя информации) встроенной аппаратурой самодиагностики.

Для использования этой технологии необходимо установить пакет smartmontools, содержащий утилиты, позволяющие получать данные от жёсткого диска, интерпритировать и отображать их, а также выполнять тестирование диска. Также в состав этого пакета входит демон smartd (и соответствующая служба systemd), который в автоматическом режиме следит за состоянием диска и информирует системного администратора. Об этом подробнее сказано в разделе «Использование сервера».

Кроме того, весьма удобно использовать программу GSmartControl – утилиту для работы со SMART, имеющую графический пользовательский интерфейс.

Установить эти программные средства можно при помощи команд:

```
root@server:/# apt-get install smartmontools
root@server:/# apt-get install gsmartcontrol
```

Их применение показано в разделе «Использование сервера».

## Настройка сети

В этом разделе описывается только настройка сетевых интерфейсов самой системы, описание создания беспроводной точки доступа приведено далее. Настройку сети можно разделить на следующие этапы:

- установка необходимого программного обеспечения;
- настройка сетевых интерфейсов;
- настройка маршрутизации и транзита пакетов;
- настройка системы управления пропускной способностью;

Для полноценной работы беспроводной сетевой карты в рассматриваемой системе требуется установить пакет `firmware-atheros`. Это можно сделать сразу после настройки проводного сетевого интерфейса или перенесением пакета вручную со сменного носителя. При настроенном сетевом интерфейсе и списке репозиториях выполнить установку можно командой:

```
root@server:/# apt-get install firmware-atheros
```

Остальные необходимые для настройки сети средства (`iproute2`, `tc`) установлены вместе с системой.

## Настройка сетевых интерфейсов

В системе присутствуют два сетевых интерфейса: проводной `enp7s0` и беспроводной `wlp6s0`, что видно из вывода следующей команды:

```
root@server:/# ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp7s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode
DEFAULT group default qlen 1000
    link/ether b8:70:f4:29:e5:83 brd ff:ff:ff:ff:ff:ff
3: wlp6s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode
DEFAULT group default qlen 1000
    link/ether d0:df:9a:6a:89:18 brd ff:ff:ff:ff:ff:ff
```

Настройка сетевых интерфейсов выполняется внесением необходимых сведений в файл **/etc/network/interfaces**. В данном случае он имеет вид (с некоторыми несущественными сокращениями):

```
source /etc/network/interfaces.d/*

#The loopback network interface
auto lo
iface lo inet loopback

#External network interface
auto enp7s0
iface enp7s0 inet static
```



```
address 192.168.0.44/24
gateway 192.168.0.1
pre-up /etc/network/neighbors
pre-up /etc/network/shaping
up      /etc/network/routes
```

```
#Internal network interface
auto wlp6s0
iface wlp6s0 inet static
    address 192.168.4.1/24
```

Тем самым проводной сетевой интерфейс, выступающий в роли внешнего (относительно клиентов) получает адрес в ядре сети, а беспроводной (внутренний) – адрес в беспроводной подсети.

Для применения нового конфигурационного файла следует перезапустить сетевую подсистему командой:

```
root@server:/# systemctl restart networking
```

Затем следует убедиться в успешности перезапуска по выводу команды:

```
root@server:/# systemctl status networking
• networking.service - Raise network interfaces
  Loaded: loaded (/lib/systemd/system/networking.service; enabled; vendor preset:
  enabled)
  Active: active (exited) since Mon 2020-03-16 13:51:51 MSK; 40s ago
  Docs: man:interfaces(5)
  Process: 1499 ExecStart=/sbin/ifup -a --read-environment (code=exited,
  status=0/SUCCESS)
  Main PID: 1499 (code=exited, status=0/SUCCESS)
```

```
map 16 13:51:45 server systemd[1]: Starting Raise network interfaces...
map 16 13:51:51 server systemd[1]: Started Raise network interfaces.
```

Наконец, увидеть новые сетевые настройки системы можно в выводе команды:

```
root@server:/# ip address show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
qlen 1000
  link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
  inet 127.0.0.1/8 scope host lo
    valid_lft forever preferred_lft forever
2: enp7s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc htb state UP group
default qlen 1000
  link/ether b8:70:f4:29:e5:83 brd ff:ff:ff:ff:ff:ff
  inet 192.168.0.44/24 brd 192.168.0.255 scope global enp7s0
    valid_lft forever preferred_lft forever
3: wlp6s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group
default qlen 1000
  link/ether d0:df:9a:6a:89:18 brd ff:ff:ff:ff:ff:ff
  inet 192.168.4.1/24 brd 192.168.4.255 scope global wlp6s0
    valid_lft forever preferred_lft forever
```

Как видно из этого вывода, в системе присутствуют три сетевых интерфейса, считая петлевой, они используют именно те адреса, которые и были им заданы, при этом адреса IPv6 не назначены. Таким образом настройки сетевых интерфейсов полностью соответствуют заданным выше требованиям.

Кроме того, перед запуском внешнего сетевого интерфейса и в его ходе выполняются три сценария: внесение статических записей в низкоуровневые адресные таблицы, настройка маршрутизации и настройка полосы пропускания для клиентских соединений.

Все сценарии расположены в том же каталоге **/etc/network**, что и основной конфигурационный файл сетевых интерфейсов, и доступны для чтения, записи и исполнения только суперпользователю. Создание этих файлов и установка их атрибутов может быть выполнена командами (в выводе последней команды отображены лишь описываемые файлы):

```
root@server:/# cd /etc/network
root@server:/etc/network# touch shaping routes neighbours
root@server:/etc/network# chmod 0700 shaping routes neighbours
root@server:/etc/network# ls -l
...
-rw-r--r-- 1 root root  548 map 25 10:48 interfaces
...
-rwx----- 1 root root  331 map 24 13:49 neighbours
-rwx----- 1 root root   67 map 25 10:51 routes
-rwx----- 1 root root 1197 фев 26 14:18 shaping
```

На данный момент сценарий создания статических записей **arp** имеет вид:

```
#!/bin/sh

ip neighbour add 192.168.0.1 dev enp7s0 lladdr 00:c0:26:a7:b9:39 nud permanent
ip neighbour add 192.168.0.2 dev enp7s0 lladdr 90:2b:34:48:08:b5 nud permanent
ip neighbour add 192.168.0.4 dev enp7s0 lladdr 94:de:80:bc:33:de nud permanent
ip neighbour add 192.168.0.77 dev enp7s0 lladdr b8:70:f4:22:b0:6b nud permanent
```

По мере замены оборудования на сервере виртуализации и основном шлюзе их аппаратные адреса должны быть заменены.

Вывод следующей команды, выполненной после перезапуска сетевой подсистемы, демонстрирует проверку статических записей об аппаратных адресах:

```
root@server:/# ip neighbour show
192.168.0.4 dev enp7s0 lladdr 94:de:80:bc:33:de PERMANENT
192.168.0.2 dev enp7s0 lladdr 90:2b:34:48:08:b5 PERMANENT
192.168.0.1 dev enp7s0 lladdr 00:c0:26:a7:b9:39 PERMANENT
192.168.0.77 dev enp7s0 lladdr b8:70:f4:22:b0:6b PERMANENT
```

## Маршрутизация

Следует отметить, что связь устройств в локальной сети организации реализована при помощи разбиения локальной сети на изолированные подсети, защищённые своими межсетевыми экранами, одновременно выполняющими роль серверов таких подсетей. Одним из таких серверов (для беспроводной подсети) и является рассматриваемая система. При этом эти серверы отвечают за маршрутизацию пакетов и выполнение преобразования адресов (NAT). С точки зрения маршрутизации каждый такой сервер должен, с одной стороны, предоставлять клиентам кратчайший маршрут до целевого узла сети, а с другой – не предоставлять маршрута к тем подсетям, взаимодействие с которыми не предусмотрено логикой работы сети и организации. В данном случае, сервер беспроводной сети должен предоставлять клиентам только маршруты до ядра

сети (в том числе главного шлюза и сервера АИС «Аверс») и до сети виртуальных серверов. Также должен быть предоставлен маршрут по умолчанию, направленный через главный шлюз организации. Здесь не рассматриваются вопросы защиты от попыток попасть в другие подсети, используя, таблицы маршрутизации других серверов, например, главного шлюза, т. к. защита от таких атак – удел межсетевого экрана самого этого шлюза. Далее в разделе «Установка и настройка сетевого фильтра (nftables)» описаны меры, противодействующие использованию этого сервера для той же цели.

Касаясь вопроса преобразования адресов, следует отметить, что никакого влияния на маршрутизацию внутри локальной сети NAT не имеет. При обращении ко внешним ресурсам симметричный NAT выполняется на главном шлюзе организации. Внутри локальной сети такое преобразование может иметь в своём применении как положительные, так и отрицательные черты. К отрицательным аспектам следует отнести повышенную нагрузку на машину, выполняющую это преобразование, и невозможность в журналах серверов выяснить исходный ip-адрес клиента. Эту же невозможность в некоторых случаях следует считать достоинством. Например, применяя NAT на сервере подсети бухгалтерии, можно добиться достаточно надёжного сокрытия информации о количестве хостов в этой подсети, их внутренних адресах, используемых каждым из них портов и т. д. Как следствие, станет невозможно определить, с какого из компьютеров бухгалтерии выполнялось подключение к локальному информационному серверу. При применении того же механизма к беспроводной подсети результатом будет сокрытие следов клиента-злоумышленника в журналах того же инфосервера. Именно поэтому в беспроводной сети NAT не используется.

Прежде всего при настройке маршрутизации следует обеспечить транзит сетевых пакетов между интерфейсами рассматриваемого сервера. Для этого следует привести конфигурационный файл **/etc/sysctl.conf** к такому виду, чтобы в нём содержалась строка

```
net.ipv4.ip_forward=1
```

Обычно для этого достаточно её просто раскомментировать. Для включения транзита пакетов в текущей сессии без перезагрузки системы достаточно выполнить команду:

```
root@server:/# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Затем следует добавить необходимые сетевые маршруты с тем, чтобы слать пакеты по кратчайшему пути. Для этого в упомянутый ранее сценарий настройки маршрутизации достаточно привести к виду:

```
root@server:/# cat /etc/network/routes
#!/bin/sh
```

```
ip route add 192.168.7.0/24 via 192.168.0.77 dev enp7s0
```

После перезапуска сетевой подсистемы при помощи следующих команд можно убедиться как в наличии, так и в работоспособности добавленного маршрута:

```
root@server:/# ip route show
default via 192.168.0.1 dev enp7s0 onlink
192.168.0.0/24 dev enp7s0 proto kernel scope link src 192.168.0.44
192.168.4.0/24 dev wlp6s0 proto kernel scope link src 192.168.4.1
192.168.7.0/24 via 192.168.0.77 dev enp7s0
root@server:/# traceroute 192.168.7.1
traceroute to 192.168.7.1 (192.168.7.1), 30 hops max, 60 byte packets
1  server.service.school34 (192.168.7.1)  0.524 ms  0.411 ms  0.316 ms
```

```

root@server:/# ping -c 3 debian.service.school34
PING debian.service.school34 (192.168.7.5) 56(84) bytes of data.
64 bytes from debian.service.school34 (192.168.7.5): icmp_seq=1 ttl=63 time=2.22 ms
64 bytes from debian.service.school34 (192.168.7.5): icmp_seq=2 ttl=63 time=1.92 ms
64 bytes from debian.service.school34 (192.168.7.5): icmp_seq=3 ttl=63 time=2.04 ms

--- debian.service.school34 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 5ms
rtt min/avg/max/mdev = 1.916/2.061/2.224/0.126 ms

```

В представленной первой командой таблице маршрутизации присутствуют также маршруты, созданные на основе настроек сетевых интерфейсов. В совокупности они составляют полный набор маршрутов, необходимых серверу для работы.

## Управление пропускной способностью сети

Настройка полосы пропускания для клиентских соединений выполнена средствами механизма traffic control и утилиты tc исходя из следующих соображений:

- трафик делится на три класса: соединения с сервером системы «Аверс», соединения с внутренним сервером виртуализации и соединения с ресурсами Интернет
- для трафика к серверу системы «Аверс» гарантируется полоса пропускания 20 Мбит/с с возможностью её расширения до 50 Мбит/с
- для трафика к внешним Интернет-ресурсам гарантируется полоса пропускания 1 Мбит/с с возможностью её расширения до 10 Мбит/с
- для трафика к внутренним серверам гарантируется полоса пропускания 40 Мбит/с с возможностью её расширения до 100 Мбит/с
- полоса пропускания для трафика к серверу системы «Аверс» ограничена сверху 50 Мбит/с с целью предотвращения DoS/DDoS-атак на этот сервер со стороны клиентов беспроводной сети
- полоса пропускания к внешним Интернет-ресурсам ограничена сверху шириной внешнего канала школы
- полоса пропускания для трафика к внутренним серверам школы ограничена сверху пропускной способностью проводного сетевого интерфейса описываемой системы
- оверкоммитинг, т.е. установка суммарной гарантированной пропускной способности, превышающей аппаратные возможности интерфейса, не допускается

Исходя из приведённых соображений сценарий настройки полосы пропускания для транзитных сетевых соединений принимает вид:

```

#!/bin/sh

#Сбрасываем текущее состояние
tc qdisc delete dev enp7s0 root
#Ставим корневую дисциплину
tc qdisc add dev enp7s0 root handle 1: htb default 13
#Создаём корневой класс
tc class add dev enp7s0 parent 1: classid 1:1 htb rate 100mbit ceil 100mbit
#Создаём подклассы: avers, service, inet
tc class add dev enp7s0 parent 1:1 classid 1:11 htb rate 20mbit ceil 50mbit
tc class add dev enp7s0 parent 1:1 classid 1:12 htb rate 40mbit ceil 100mbit
tc class add dev enp7s0 parent 1:1 classid 1:13 htb rate 1mbit ceil 10mbit
#Настраиваем дисциплины для подклассов
tc qdisc add dev enp7s0 parent 1:11 handle 10:0 sfq perturb 10

```

```

tc qdisc add dev enp7s0 parent 1:12 handle 20:0 sfq perturb 10
tc qdisc add dev enp7s0 parent 1:13 handle 30:0 sfq perturb 10
#Настраиваем фильтры для классификации трафика (по умолчанию - inet)
tc filter add dev enp7s0 protocol ip parent 1:0 prio 1 u32 match ip dst 192.168.0.4
flowid 1:11
tc filter add dev enp7s0 protocol ip parent 1:0 prio 1 u32 match ip dst
192.168.7.0/24 flowid 1:12

```

В первую очередь сценарий удаляет текущую корневую дисциплину вместе со всеми её компонентами и подключает в качестве корневой дисциплину htb (Hierarchical Token Bucket), при этом указывается, что весь неклассифицированный (default) трафик должен быть обработан с помощью дисциплин класса 1:13 (Интернет-трафик). Затем создаётся корневой класс, куда будет попадать весь трафик (это нужно для реализации заимствования). В этом классе пропускная способность ограничивается согласно аппаратным возможностям. Далее создаются три подкласса, обеспечивающие заданное разделение ширины канала. После этого создаются три дисциплины sfq (Stochastic Fairness Queueing), по одной для каждого класса. В завершение создаются два фильтра: первый классифицирует трафик к серверу АИС «Аверс», второй – к серверу виртуализации. Неклассифицированный трафик считается трафиком, направленным в Интернет.

После запуска сценария проверить состояние подсистемы контроля трафика можно следующими командами:

```

root@server:/# tc qdisc show
qdisc noqueue 0: dev lo root refcnt 2
qdisc htb 1: dev enp7s0 root refcnt 2 r2q 10 default 0x13 direct_packets_stat 0
direct_qlen 1000
qdisc sfq 30: dev enp7s0 parent 1:13 limit 127p quantum 1514b depth 127 divisor 1024
perturb 10sec
qdisc sfq 10: dev enp7s0 parent 1:11 limit 127p quantum 1514b depth 127 divisor 1024
perturb 10sec
qdisc sfq 20: dev enp7s0 parent 1:12 limit 127p quantum 1514b depth 127 divisor 1024
perturb 10sec
qdisc noqueue 0: dev wlp6s0 root refcnt 2
root@server:/# tc class show dev enp7s0
class htb 1:11 parent 1:1 leaf 10: prio 0 rate 20Mbit ceil 50Mbit burst 1600b cburst
1600b
class htb 1:1 root rate 100Mbit ceil 100Mbit burst 1600b cburst 1600b
class htb 1:13 parent 1:1 leaf 30: prio 0 rate 1Mbit ceil 10Mbit burst 1600b cburst
1600b
class htb 1:12 parent 1:1 leaf 20: prio 0 rate 40Mbit ceil 100Mbit burst 1600b cburst
1600b
root@server:/# tc filter show dev enp7s0
filter parent 1: protocol ip pref 1 u32 chain 0
filter parent 1: protocol ip pref 1 u32 chain 0 fh 800: ht divisor 1
filter parent 1: protocol ip pref 1 u32 chain 0 fh 800::800 order 2048 key ht 800 bkt
0 flowid 1:11 not_in_hw
match c0a80004/ffffffff at 16
filter parent 1: protocol ip pref 1 u32 chain 0 fh 800::801 order 2049 key ht 800 bkt
0 flowid 1:12 not_in_hw
match c0a80700/ffffffff00 at 16

```

## Организация беспроводной сети

Для организации беспроводной сети потребуется:

- убедиться, что беспроводная сетевая карта на это технически способна;
- убедиться, что беспроводная связь в системе разблокирована;
- настроить беспроводную точку доступа и подобрать параметры её работы;
- обеспечить парольную защиту точки доступа;
- убедиться в работоспособности и устойчивости точки доступа.

Далее описан каждый из этих этапов.

## Управление беспроводной сетевой картой

Прежде всего следует выяснить технические возможности имеющейся в системе беспроводной сетевой карты. Для этого достаточно выполнить команду:

```
root@server:/# iw list
Wiphy phy0
  max # scan SSIDs: 4
  max scan IEs length: 2257 bytes
  max # sched scan SSIDs: 0
  max # match sets: 0
  max # scan plans: 1
  max scan plan interval: -1
  max scan plan iterations: 0
  Retry short limit: 7
  Retry long limit: 4
  Coverage class: 0 (up to 0m)
  Device supports RSN-IBSS.
  Device supports AP-side u-APSD.
  Device supports T-DLS.
  Supported Ciphers:
  * WEP40 (00-0f-ac:1)
  * WEP104 (00-0f-ac:5)
  * TKIP (00-0f-ac:2)
  * CCMP-128 (00-0f-ac:4)
  * CCMP-256 (00-0f-ac:10)
  * GCMP-128 (00-0f-ac:8)
  * GCMP-256 (00-0f-ac:9)
  * CMAC (00-0f-ac:6)
  * CMAC-256 (00-0f-ac:13)
  * GMAC-128 (00-0f-ac:11)
  * GMAC-256 (00-0f-ac:12)
  Available Antennas: TX 0x1 RX 0x3
  Configured Antennas: TX 0x1 RX 0x3
  Supported interface modes:
  * IBSS
  * managed
  * AP
  * AP/VLAN
  * monitor
  * mesh point
  * P2P-client
  * P2P-GO
  * outside context of a BSS
  Band 1:
  Capabilities: 0x11ce
  HT20/HT40
  SM Power Save disabled
  RX HT40 SGI
  TX STBC
  RX STBC 1-stream
  Max AMSDU length: 3839 bytes
```

## DSSS/CCK HT40

Maximum RX AMPDU length 65535 bytes (exponent: 0x003)

Minimum RX AMPDU time spacing: 8 usec (0x06)

HT TX/RX MCS rate indexes supported: 0-7

Bitrates (non-HT):

- \* 1.0 Mbps
- \* 2.0 Mbps (short preamble supported)
- \* 5.5 Mbps (short preamble supported)
- \* 11.0 Mbps (short preamble supported)
- \* 6.0 Mbps
- \* 9.0 Mbps
- \* 12.0 Mbps
- \* 18.0 Mbps
- \* 24.0 Mbps
- \* 36.0 Mbps
- \* 48.0 Mbps
- \* 54.0 Mbps

Frequencies:

- \* 2412 MHz [1] (17.0 dBm)
- \* 2417 MHz [2] (17.0 dBm)
- \* 2422 MHz [3] (17.0 dBm)
- \* 2427 MHz [4] (17.0 dBm)
- \* 2432 MHz [5] (17.0 dBm)
- \* 2437 MHz [6] (17.0 dBm)
- \* 2442 MHz [7] (17.0 dBm)
- \* 2447 MHz [8] (17.0 dBm)
- \* 2452 MHz [9] (17.0 dBm)
- \* 2457 MHz [10] (17.0 dBm)
- \* 2462 MHz [11] (17.0 dBm)
- \* 2467 MHz [12] (17.0 dBm) (no IR)
- \* 2472 MHz [13] (17.0 dBm) (no IR)
- \* 2484 MHz [14] (disabled)

Supported commands:

- \* new\_interface
- \* set\_interface
- \* new\_key
- \* start\_ap
- \* new\_station
- \* new\_mpath
- \* set\_mesh\_config
- \* set\_bss
- \* authenticate
- \* associate
- \* deauthenticate
- \* disassociate
- \* join\_ibss
- \* join\_mesh
- \* remain\_on\_channel
- \* set\_tx\_bitrate\_mask
- \* frame
- \* frame\_wait\_cancel
- \* set\_wiphy\_netns
- \* set\_channel
- \* set\_wds\_peer
- \* tdls\_mgmt
- \* tdls\_oper
- \* probe\_client
- \* set\_noack\_map
- \* register\_beacons
- \* start\_p2p\_device
- \* set\_mcast\_rate
- \* connect

```

    * disconnect
    * channel_switch
    * set_qos_map
    * set_multicast_to_unicast
Supported TX frame types:
    * IBSS: 0x00 0x10 0x20 0x30 0x40 0x50 0x60 0x70 0x80 0x90 0xa0 0xb0
0xc0 0xd0 0xe0 0xf0
    * managed: 0x00 0x10 0x20 0x30 0x40 0x50 0x60 0x70 0x80 0x90 0xa0
0xb0 0xc0 0xd0 0xe0 0xf0
    * AP: 0x00 0x10 0x20 0x30 0x40 0x50 0x60 0x70 0x80 0x90 0xa0 0xb0
0xc0 0xd0 0xe0 0xf0
    * AP/VLAN: 0x00 0x10 0x20 0x30 0x40 0x50 0x60 0x70 0x80 0x90 0xa0
0xb0 0xc0 0xd0 0xe0 0xf0
    * mesh point: 0x00 0x10 0x20 0x30 0x40 0x50 0x60 0x70 0x80 0x90 0xa0
0xb0 0xc0 0xd0 0xe0 0xf0
    * P2P-client: 0x00 0x10 0x20 0x30 0x40 0x50 0x60 0x70 0x80 0x90 0xa0
0xb0 0xc0 0xd0 0xe0 0xf0
    * P2P-GO: 0x00 0x10 0x20 0x30 0x40 0x50 0x60 0x70 0x80 0x90 0xa0 0xb0
0xc0 0xd0 0xe0 0xf0
    * P2P-device: 0x00 0x10 0x20 0x30 0x40 0x50 0x60 0x70 0x80 0x90 0xa0
0xb0 0xc0 0xd0 0xe0 0xf0
Supported RX frame types:
    * IBSS: 0x40 0xb0 0xc0 0xd0
    * managed: 0x40 0xd0
    * AP: 0x00 0x20 0x40 0xa0 0xb0 0xc0 0xd0
    * AP/VLAN: 0x00 0x20 0x40 0xa0 0xb0 0xc0 0xd0
    * mesh point: 0xb0 0xc0 0xd0
    * P2P-client: 0x40 0xd0
    * P2P-GO: 0x00 0x20 0x40 0xa0 0xb0 0xc0 0xd0
    * P2P-device: 0x40 0xd0
software interface modes (can always be added):
    * AP/VLAN
    * monitor
valid interface combinations:
    * #{ managed } <= 2048, #{ AP, mesh point } <= 8, #{ P2P-client, P2P-
GO } <= 1,
        total <= 2048, #channels <= 1, STA/AP BI must match
HT Capability overrides:
    * MCS: ff ff ff ff ff ff ff ff
    * maximum A-MSDU length
    * supported channel width
    * short GI for 40 MHz
    * max A-MPDU length exponent
    * min MPDU start spacing
Device supports TX status socket option.
Device supports HT-IBSS.
Device supports SAE with AUTHENTICATE command
Device supports low priority scan.
Device supports scan flush.
Device supports AP scan.
Device supports per-vif TX power setting
P2P GO supports CT window setting
Driver supports full state transitions for AP/GO clients
Driver supports a userspace MPM
Device supports active monitor (which will ACK incoming frames)
Driver/device bandwidth changes during BSS lifetime (AP/GO mode)
Device supports configuring vdev MAC-addr on create.
Supported extended features:
    * [ RRM ]: RRM
    * [ FILS_STA ]: STA FILS (Fast Initial Link Setup)
    * [ CQM_RSSI_LIST ]: multiple CQM_RSSI_THOLD records

```



```
* [ CONTROL_PORT_OVER_NL80211 ]: control port over nl80211
* [ TXQS ]: FQ-CoDel-enabled intermediate TXQs
```

Из вывода этой команды (вывод не был сокращён) можно отметить, что wifi-карта

- имеет системное имя phy0;
- поддерживает режим интерфейса AP (Access Point);
- поддерживает скорость передачи до 54 Мбит/с (без учёта применения HT-возможностей);
- может использовать каналы с 1 по 13 включительно.

Возможность работы в режиме точки доступа – необходимое условие для реализации поставленной перед системой задачи, остальные параметры следует учитывать при настройке программной точки доступа (демона hostapd).

Также полезной в работе может оказаться утилита rfkill, предназначенная для включения/отключения беспроводной сетевой карты, а также для контроля её состояния. Для проверки состояния карты достаточно выполнить эту команду без параметров:

```
root@server:/# rfkill
ID TYPE DEVICE          SOFT      HARD
0 wlan ideapad_wlan     unblocked unblocked
1 wlan phy0             unblocked unblocked
```

Представленные два устройства на самом деле являются одним физическим – беспроводной сетевой картой, указанной в описании аппаратной составляющей системы. В этом можно убедиться, выключив эту сетевую карту при помощи механического переключателя (HARD – «жёсткое» выключение) на корпусе устройства и выполнив её повторно:

```
root@server:/# rfkill
ID TYPE DEVICE          SOFT      HARD
0 wlan ideapad_wlan     unblocked blocked
1 wlan phy0             unblocked blocked
```

То же самое наблюдается и при блокировке при помощи клавиатуры устройства (SOFT – «мягкое» выключение):

```
root@server:/# rfkill
ID TYPE DEVICE          SOFT      HARD
0 wlan ideapad_wlan     blocked  unblocked
1 wlan phy0             blocked  blocked
```

Эта команда, среди прочего, может быть полезна для дистанционной разблокировки беспроводной карты при её случайной блокировке пользователем системы, разумеется, только в случае «мягкой» блокировки.

Очевидно, что перед продолжением настройки системы и созданием точки доступа следует убедиться, что сетевая карта разблокирована.

## Настройка программной точки доступа

За создание программной точки доступа на базе настроенной беспроводной сетевой карты отвечает демон hostapd. Прежде всего его необходимо установить в систему командой:

```
root@server:/# apt-get install hostapd
```

Далее конфигурационный файл этого демона **/etc/hostapd** следует привести к виду:

```
interface=wlp6s0
driver=nl80211
ssid="wifi at school34"
country_code=RU
hw_mode=g
ieee80211n=1
ht_capab=[HT40-][SHORT-GI-40]
channel=6
wpa=2
wpa_passphrase=48C74A6B
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
auth_algs=1
macaddr_acl=0
```

В этом файле указаны следующие значения параметров:

wlp6s0 – беспроводной сетевой интерфейс системы, на котором следует организовать точку доступа.

nl80211 – драйвер сетевой карты, который следует использовать, этот драйвер подходит для большинства сетевых карт, в том числе и для используемой в системе.

ssid – название точки доступа в том виде, в котором оно отображается у клиентов при подключении.

hw\_mode = g – режим работы сетевой карты (802.11b/g/n). При этом даже если карта способна работать в режиме 802.11n, здесь всё равно следует указывать режим g, Дополнительные возможности, вводимые стандартом 802.11n, здесь подключаются с помощью следующих двух параметров.

ieee80211n – использовать режим 802.11n.

ht\_capab – это набор параметров для тонкой настройки параметров работы беспроводной сети, конкретная комбинация которых подбирается в зависимости от оборудования точки доступа, оборудования клиентских устройств, расположения точки доступа на местности, архитектуры здания, необходимой зоны покрытия, наличия в зоне покрытия предметов, способных заглушить или ослабить сигнал (зеркала, сейфы, микроволновые печи и т.д.), пересечения с другими беспроводными сетями и многих других обстоятельств. На рассматриваемой системе опытным путём были подобраны значения, обеспечивающие приемлимое качество работы беспроводной точки доступа.

channel – используемый беспроводной канал, одно из предлагаемых по умолчанию значений (6) вполне подошло для применения в рассматриваемых условиях.

Следующие далее параметры указывают на методы защиты беспроводной сети. Так используется технология Wifi Protected Access 2 (WPA2) с постоянно хранимым на самом маршрутизаторе паролем (WPA-PSK). Не вдаваясь в технические детали WiFi, следует отметить параметр wpa\_passphrase, т.к. его значение и является паролем для подключения к точке доступа. По логике функционирования описываемой беспроводной сети в рамках помещения, в котором установлен сервер беспроводной сети и клиентские устройства, не следует делать из этого пароля секрет. Однако, необходима регулярная

смена пароля, чтобы минимизировать угрозу подключения клиента, находящегося за пределами указанного помещения. Для решения этой задачи разработан описанный далее сценарий командной строки, который в процессе своей работы меняет пароль в этом файле и перезапускает службу.

Последний параметр `macaddr_acl` при указанном значении позволяет подключаться к сети любому клиентскому устройству.

Затем следует в конфигурационном файле `/etc/default/hostapd` раскомментировать строку

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

чтобы обеспечить использование приведённого выше конфигурационного файла и запуск демона при загрузке системы.

## Автоматическая смена пароля точки доступа

Автоматическая смена пароля организована следующим образом: создан сценарий, генерирующий новый пароль и размещающий его в необходимых конфигурационных файлах, организован вызов этого сценария при перезапусках службы `hostapd`, предусмотрено отображение пароля на экране ноутбука.

Сценарий, расположенный в `/usr/local/sbin/wifipasswd`, имеет вид:

```
#!/bin/bash

NEWPASSWORD=$(/usr/bin/makepasswd --string 0123456789ABCDEF --count 1 --chars 8)
/usr/bin/sed s/"^wpa_passphrase=.*$"/"wpa_passphrase=$NEWPASSWORD"/g -i /etc/hostapd/hostapd.conf
echo -n $NEWPASSWORD > /usr/local/etc/wifi.password
exit 0
```

Сценарий создаёт новый восьмисимвольный пароль, состоящий из шестнадцатеричных цифр (буквы в верхнем регистре), что обеспечивает свыше 4 миллиардов различных комбинаций. Затем сценарий изменяет конфигурационный файл беспроводной точки доступа так, чтобы тот содержал новый пароль, а также вносит этот новый пароль в файл `/usr/local/etc/wifi.password`, который при помощи виджета `RichTextViewer` отображается на Рабочем столе пользователя `guest`.

Для выполнения этого сценария перед каждым запуском службы `hostapd` необходимо эту службу дополнить. Для этого следует создать файл **`override.conf`** в каталоге `/etc/systemd/system/hostapd.service.d` со следующим содержимым:

```
[Service]
ExecStartPre=-/usr/local/sbin/wifipasswd
```

Расположение в указанном каталоге позволяет файлу с изменениями службы не быть затёртым при обновлениях `systemd`. Правильнее всего создать такой файл при помощи команды

```
root@server:/# systemctl edit hostapd
```

После завершения редактирования службы следует перезапустить её, проверить её состояние и убедиться в применении изменений например так, как, это показано в следующем фрагменте терминального сеанса:

```
root@server:/# systemctl restart hostapd
root@server:/# systemctl status hostapd
• hostapd.service - Advanced IEEE 802.11 AP and IEEE 802.1X/WPA/WPA2/EAP
Authenticator
  Loaded: loaded (/lib/systemd/system/hostapd.service; enabled; vendor preset:
enabled)
  Drop-In: /etc/systemd/system/hostapd.service.d
           └─override.conf
  Active: active (running) since Thu 2020-02-20 15:51:41 MSK; 3s ago
  Process: 2728 ExecStartPre=/usr/local/sbin/wifipasswd (code=exited,
status=0/SUCCESS)
  Process: 2731 ExecStart=/usr/sbin/hostapd -B -P /run/hostapd.pid -B $DAEMON_OPTS $
{DAEMON_CONF} (code=exited, status=0/SUCCESS)
  Main PID: 2732 (hostapd)
    Tasks: 1 (limit: 4509)
   Memory: 1.4M
    CGroup: /system.slice/hostapd.service
            └─2732 /usr/sbin/hostapd -B -P /run/hostapd.pid -B
/etc/hostapd/hostapd.conf
root@server:/# systemctl show hostapd
Type=forking
Restart=on-failure
PIDFile=/run/hostapd.pid
ExecMainStatus=0
ExecStartPre={ path=/usr/local/sbin/wifipasswd ; argv[]=/usr/local/sbin/wifipasswd ;
ignore_errors=yes ; start_time=[Thu 2020-02-20 15:51:41 MSK] ; stop_time=[Thu 2020-0
ExecStart={ path=/usr/sbin/hostapd ; argv[]=/usr/sbin/hostapd -B -P /run/hostapd.pid
-B $DAEMON_OPTS ${DAEMON_CONF} ; ignore_errors=no ; start_time=[Thu 2020-02-20
15:51:41 MS
Slice=system.slice
ControlGroup=/system.slice/hostapd.service
MemoryCurrent=1495040
```

Незначающий в рассматриваемом контексте вывод команд здесь сокращён.

## Защита от падений (автоматическое возобновление)

Как видно из вышеприведённого фрагмента терминального сеанса, автоматический перезапуск службы при её отказах был предусмотрен уже в составе базовой установки службы (Restart=on-failure).

## Проверка работоспособности и устойчивости

После завершения всех настроек следует перезапустить службу и проверить её состояние:

```
root@server:/# systemctl status hostapd
• hostapd.service - Advanced IEEE 802.11 AP and IEEE 802.1X/WPA/WPA2/EAP
Authenticator
  Loaded: loaded (/lib/systemd/system/hostapd.service; enabled; vendor preset:
enabled)
  Drop-In: /etc/systemd/system/hostapd.service.d
           └─override.conf
  Active: active (running) since Wed 2020-03-18 13:41:34 MSK; 25min ago
  Process: 644 ExecStartPre=/usr/local/sbin/wifipasswd (code=exited, status=0/SUCCESS)
```

```

Process: 712 ExecStart=/usr/sbin/hostapd -B -P /run/hostapd.pid -B $DAEMON_OPTS $
{DAEMON_CONF} (code=exited, status=0/SUCCESS)
Main PID: 716 (hostapd)
  Tasks: 1 (limit: 4509)
Memory: 7.2M
CGroup: /system.slice/hostapd.service
        └─716 /usr/sbin/hostapd -B -P /run/hostapd.pid -B /etc/hostapd/hostapd.conf

```

Как видно из вывода команды, процесс запущен, активен и имеет PID 716.

Теперь необходимо проверить реакцию службы на принудительное завершение основного процесса, «мягкое» и «жесткое» отключение беспроводного адаптера. При этом доступность сети следует контролировать с клиентского устройства. Далее приведены фрагмент терминального сеанса, отражающий ход такой проверки, описание процесса проверки и поясняющие комментарии.

Фрагмент терминального сеанса (с некоторыми незначачими сокращениями) :

```

root@server:/# kill -9 716
root@server:/# systemctl status hostapd
• hostapd.service - Advanced IEEE 802.11 AP and IEEE 802.1X/WPA/WPA2/EAP
Authenticator
  Loaded: loaded (/lib/systemd/system/hostapd.service; enabled; vendor preset:
enabled)
  Drop-In: /etc/systemd/system/hostapd.service.d
           └─override.conf
  Active: active (running) since Wed 2020-03-18 14:15:45 MSK; 1s ago
  Process: 1234 ExecStartPre=/usr/local/sbin/wifipasswd (code=exited,
status=0/SUCCESS)
  Process: 1237 ExecStart=/usr/sbin/hostapd -B -P /run/hostapd.pid -B $DAEMON_OPTS $
{DAEMON_CONF} (code=exited, status=0/SUCCESS)
  Main PID: 1238 (hostapd)
    Tasks: 1 (limit: 4509)
  Memory: 1.4M
  CGroup: /system.slice/hostapd.service
          └─1238 /usr/sbin/hostapd -B -P /run/hostapd.pid -B
/etc/hostapd/hostapd.conf
root@server:/# rfkill
ID TYPE DEVICE          SOFT    HARD
0 wlan ideapad_wlan    unblocked blocked
1 wlan phy0            unblocked blocked
root@server:/# systemctl status hostapd
• hostapd.service - Advanced IEEE 802.11 AP and IEEE 802.1X/WPA/WPA2/EAP
Authenticator
  Loaded: loaded (/lib/systemd/system/hostapd.service; enabled; vendor preset:
enabled)
  Drop-In: /etc/systemd/system/hostapd.service.d
           └─override.conf
  Active: active (running) since Wed 2020-03-18 14:15:45 MSK; 1min 52s ago
  Process: 1234 ExecStartPre=/usr/local/sbin/wifipasswd (code=exited,
status=0/SUCCESS)
  Process: 1237 ExecStart=/usr/sbin/hostapd -B -P /run/hostapd.pid -B $DAEMON_OPTS $
{DAEMON_CONF} (code=exited, status=0/SUCCESS)
  Main PID: 1238 (hostapd)
    Tasks: 1 (limit: 4509)
  Memory: 1.4M
  CGroup: /system.slice/hostapd.service
          └─1238 /usr/sbin/hostapd -B -P /run/hostapd.pid -B
/etc/hostapd/hostapd.conf
root@server:/# rfkill
ID TYPE DEVICE          SOFT    HARD

```

```

0 wlan ideapad_wlan  unblocked unblocked
1 wlan phy0          unblocked unblocked
root@server:/# rfkill
ID TYPE DEVICE      SOFT      HARD
0 wlan ideapad_wlan  blocked  unblocked
1 wlan phy0          blocked   blocked
root@server:/# rfkill
ID TYPE DEVICE      SOFT      HARD
0 wlan ideapad_wlan  unblocked unblocked
1 wlan phy0          unblocked unblocked
root@server:/# systemctl status hostapd
• hostapd.service - Advanced IEEE 802.11 AP and IEEE 802.1X/WPA/WPA2/EAP
Authenticator
  Loaded: loaded (/lib/systemd/system/hostapd.service; enabled; vendor preset:
enabled)
  Drop-In: /etc/systemd/system/hostapd.service.d
           └─override.conf
  Active: active (running) since Wed 2020-03-18 14:15:45 MSK; 3min 17s ago
  Process: 1234 ExecStartPre=/usr/local/sbin/wifipasswd (code=exited,
status=0/SUCCESS)
  Process: 1237 ExecStart=/usr/sbin/hostapd -B -P /run/hostapd.pid -B $DAEMON_OPTS $
{DAEMON_CONF} (code=exited, status=0/SUCCESS)
  Main PID: 1238 (hostapd)
    Tasks: 1 (limit: 4509)
    Memory: 1.4M
    CGroup: /system.slice/hostapd.service
            └─1238 /usr/sbin/hostapd -B -P /run/hostapd.pid -B
/etc/hostapd/hostapd.conf
root@server:/#

```

Проверка начата с того, что на клиентской машине запущена непрерывная отправка ping-запросов к основному серверу организации, путь к которому лежит через проверяемую систему. Затем принудительно завершается процесс hostapd. После этого выполняется проверка состояния этой службы, в результате которой оказывается, что служба уже перезапущена, работает с новым PID и паролем беспроводной сети. Это вынуждает клиента задержать отправку ping-пакетов и запросить новый пароль, после чего отправка запросов возобновляется.

Затем выполняется «жёсткое» отключение беспроводной сетевой карты механическим переключателем на корпусе устройства и проверяется при помощи утилиты rfkill состояние этого адаптера. Карта заблокирована, что заставляет клиента выдавать в терминал сообщения о недоступности сети. Далее беспроводной адаптер разблокируется тем же способом, опять средствами rfkill проверяется его состояние. На клиенте отправка запросов возобновляется.

После этого выполняется «мягкое» отключение сетевой карты при помощи клавиатуры устройства и проверка его состояния при помощи rfkill. Клиент снова начинает выдавать сообщения о недоступности сети. После включения сетевой карты и проверки её состояния отправка запросов на клиентском устройстве возобновляется.

Повторная проверка состояния службы свидетельствует о том, что за время отключений и включений карты сбоев и перезапусков службы hostapd не было.

Таким образом можно сделать вывод, что краткосрочные блокировки сетевой карты не влияют на работоспособность службы, а аварийный перезапуск службы ведёт к смене пароля для доступа к беспроводной сети, что может рассматриваться как дополнительная мера защиты системы в случае, если этот сбой был вызван злонамеренными действиями

некоторого клиента (например в случае появления уязвимостей в `hostapd`). В таком случае злоумышленнику придется восстанавливать подключение к сети прежде чем повторить атаку.

## Служба DNS/DHCP

### Настройка

Для реализации этой службы следует установить пакет `dnsmasq` со всеми зависимостями:

```
root@server:/# apt-get update
root@server:/# apt-get install dnsmasq
```

Затем следует привести конфигурационный файл `/etc/dnsmasq.conf` к виду:

```
listen-address=127.0.0.1
listen-address=192.168.4.1
cache-size=500
domain-needed
bind-dynamic
filterwin2k
no-resolv
no-poll
no-hosts
bogus-priv

server=77.88.8.7@enp7s0
server=/school34/192.168.0.1@enp7s0
server=/0.168.192.in-addr.arpa/192.168.0.1@enp7s0
server=/service.school34/192.168.0.77@enp7s0
server=/7.168.192.in-addr.arpa/192.168.0.77@enp7s0

local=/wifi.school34/
domain=school34,192.168.0.0/24
domain=service.school34,192.168.7.0/24
domain=wifi.school34,192.168.4.0/24

mx-target=mail.service.school34
localmx

no-dhcp-interface=lo
no-dhcp-interface=enp7s0
dhcp-range=192.168.4.11,192.168.4.55,12h
dhcp-option=6,192.168.4.1
dhcp-option=42,192.168.4.1

#log-queries
#log-facility=/var/log/dnsmasq.log
```

Тем самым, службе `dnsmasq` предписывается следующий порядок работы:

Служба принимает запросы только на локальном петлевом и внутреннем клиентском интерфейсах. Кэш DNS-записей увеличен со 150 (по умолчанию) до 500 записей, т. к. объем оперативной памяти в системе этому не препятствует, а на быстродействии службы такое увеличение сказывается положительно. Объем в 500 записей выбран как наиболее оптимальный по опыту эксплуатации сети, частью которой является беспроводная сеть, создаваемая описываемой системой.

Параметр `bind-dynamic` предписывает службе реагировать на изменение состояния сетевых интерфейсов. Таким образом, возможен запуск службы при выключенном беспроводном сетевом интерфейсе устройства с автоматическим «подхватом» этого интерфейса при его запуске. Такой подход позволяет предотвратить падение службы при физическом отключении беспроводного адаптера способом, описанным в разделе «Аппаратное обеспечение».

В своей работе служба не использует системные конфигурационные файлы встроенного DNS-клиента. Все необходимые для ее работы параметры указаны в основном конфигурационном файле службы, приведенном выше.

Службе известны три локальных домена (`school34`, `service.school34` и `wifi.school34`). Каждому из них (кроме обслуживаемого самой системой домена `wifi.school34`) сопоставлен свой DNS-сервер, обращение к которому выполняется через внешний интерфейс системы. Разрешение DNS-запросов о прочих немаршрутизируемых в Интернете адресах блокируется параметром `bogus-priv`. DNS-запросы о внешних хостах разрешаются при помощи YandexDNS, который может быть легко заменен на более строгий так, как это описано в разделе «Использование сервера».

Параметры обслуживания электронной почты заданы таким образом, чтобы почтовым адресом по умолчанию для всех клиентов беспроводной сети являлся адрес локального почтового сервера организации (`mail.service.school34`).

Протокол DHCP в беспроводной подсети раздает клиентам адреса в диапазоне 192.168.4.11 – 192.168.4.55 (45 адресов, соответственно не более 45 клиентов одновременно). Запросы на получение адреса, очевидно, принимаются только с беспроводного интерфейса системы. Кроме предоставления адреса сервер предоставляет клиенту услуги DNS-службы (`dhcp-option 6`) и службы точного времени (`dhcp-option 42`).

Служба ведет журнал в файле `/var/log/daemon.log`, где кратко отражаются сведения о запуске службы, ее настройках и т. д. Ведение подробного журнала службы в данной конфигурации отключено, однако раскомментировав последние две строки в приведенном файле, можно включить ведение журнала. При этом следует помнить, что ведение журнала, даже с учетом ротации журнальных файлов (см. «О дополнительных службах»), может служить направлением для атаки на сервер: отправка огромного количества бессмысленных запросов способна переполнить дисковое пространство сервера. В рассматриваемой системе нет необходимости в подробном журналировании происходящего, поэтому оно и отключено.

По завершении редактирования конфигурационного файла для начала работы службы остается её лишь перезапустить командой

```
root@server:/# systemctl restart dnsmasq
```

Убедиться в успешности перезапуска можно по выводу команды

```
root@server:/# systemctl status dnsmasq
```

```
• dnsmasq.service - dnsmasq - A lightweight DHCP and caching DNS server
  Loaded: loaded (/lib/systemd/system/dnsmasq.service; enabled; vendor preset:
enabled)
  Active: active (running) since Wed 2020-01-29 10:23:28 MSK; 28s ago
  Process: 1860 ExecStartPre=/usr/sbin/dnsmasq --test (code=exited, status=0/SUCCESS)
  Process: 1861 ExecStart=/etc/init.d/dnsmasq systemd-exec (code=exited,
status=0/SUCCESS)
  Process: 1870 ExecStartPost=/etc/init.d/dnsmasq systemd-start-resolvconf
```



```

(code=exited, status=0/SUCCESS)
Main PID: 1869 (dnsmasq)
  Tasks: 1 (limit: 4509)
  Memory: 1.7M
  CGroup: /system.slice/dnsmasq.service
          └─1869 /usr/sbin/dnsmasq -x /run/dnsmasq/dnsmasq.pid -u dnsmasq -7
/etc/dnsmasq.d,.dpkg-dist,.dpkg-old,.dpkg-new --local-s

янв 29 10:23:28 server dnsmasq[1869]: compile time options: IPv6 GNU-getopt DBus i18n
IDN DHCP DHCPv6 no-Lua TFTP conntrack ipset aut
янв 29 10:23:28 server dnsmasq-dhcp[1869]: DHCP, IP range 192.168.4.11 --
192.168.4.55, lease time 12h
янв 29 10:23:28 server dnsmasq[1869]: using local addresses only for domain
wifi.school34
янв 29 10:23:28 server dnsmasq[1869]: using nameserver 192.168.0.77#53 for domain
7.168.192.in-addr.arpa
янв 29 10:23:28 server dnsmasq[1869]: using nameserver 192.168.0.77#53 for domain
service.school34
янв 29 10:23:28 server dnsmasq[1869]: using nameserver 192.168.0.1#53 for domain
0.168.192.in-addr.arpa
янв 29 10:23:28 server dnsmasq[1869]: using nameserver 192.168.0.1#53 for domain
school34
янв 29 10:23:28 server dnsmasq[1869]: using nameserver 77.88.8.7#53(via enp7s0)
янв 29 10:23:28 server dnsmasq[1869]: cleared cache
янв 29 10:23:28 server systemd[1]: Started dnsmasq - A lightweight DHCP and caching
DNS server.

```

Служба работает от имени системного пользователя dnsmasq. Идентификатор процесса (PID) по умолчанию вносится в файл **/run/dnsmasq/dnsmasq.pid**. В ходе работы процесс dnsmasq реагирует на сигнал SIGUSR1 отправкой в системный журнал сообщений о текущих клиентах, статистике их запросов и т.д. Следующая команда позволяет отправить этот сигнал нужному процессу и открыть выборку из системного журнала, содержащую все упоминания о dnsmasq за последний час:

```

root@server:/# kill -s SIGUSR1 $(cat /run/dnsmasq/dnsmasq.pid) && journalctl -u
dnsmasq --since -1h

```

Эта выборка имеет вид, похожий на:

```

-- Logs begin at Fri 2020-02-28 08:55:10 MSK, end at Fri 2020-02-28 15:17:47 MSK. --
фев 28 14:48:49 server dnsmasq-dhcp[766]: DHCPDISCOVER(wlp6s0) b8:8d:12:41:29:a2
фев 28 14:48:49 server dnsmasq-dhcp[766]: DHCPOFFER(wlp6s0) 192.168.4.21
b8:8d:12:41:29:a2
фев 28 14:48:49 server dnsmasq-dhcp[766]: DHCPDISCOVER(wlp6s0) b8:8d:12:41:29:a2
фев 28 14:48:49 server dnsmasq-dhcp[766]: DHCPOFFER(wlp6s0) 192.168.4.21
b8:8d:12:41:29:a2
фев 28 14:48:50 server dnsmasq-dhcp[766]: DHCPREQUEST(wlp6s0) 192.168.4.21
b8:8d:12:41:29:a2
фев 28 14:48:50 server dnsmasq-dhcp[766]: DHCPACK(wlp6s0) 192.168.4.21
b8:8d:12:41:29:a2
фев 28 14:53:11 server dnsmasq-dhcp[766]: DHCPREQUEST(wlp6s0) 192.168.4.16
d0:df:9a:6a:4c:b4
фев 28 14:53:11 server dnsmasq-dhcp[766]: DHCPACK(wlp6s0) 192.168.4.16
d0:df:9a:6a:4c:b4 server
фев 28 14:53:47 server dnsmasq-dhcp[766]: DHCPREQUEST(wlp6s0) 192.168.4.16
d0:df:9a:6a:4c:b4
фев 28 14:53:47 server dnsmasq-dhcp[766]: DHCPACK(wlp6s0) 192.168.4.16
d0:df:9a:6a:4c:b4 server
фев 28 15:17:47 server dnsmasq[766]: time 1582892267
фев 28 15:17:47 server dnsmasq[766]: cache size 500, 0/737 cache insertions re-used

```

unexpired cache entries.

```
фев 28 15:17:47 server dnsmasq[766]: queries forwarded 268, queries answered locally 132
фев 28 15:17:47 server dnsmasq[766]: queries for authoritative zones 0
фев 28 15:17:47 server dnsmasq[766]: server 192.168.0.77#53: queries sent 0, retried or failed 0
фев 28 15:17:47 server dnsmasq[766]: server 192.168.0.1#53: queries sent 3, retried or failed 0
фев 28 15:17:47 server dnsmasq[766]: server 77.88.8.7#53: queries sent 265, retried or failed 0
```

По завершении настройки службы следует указать системе использовать собственную службу для собственных же нужд. Для этого следует привести конфигурационный файл **/etc/resolv.conf** к следующему виду:

```
nameserver 127.0.0.1
```

Затем следует перезапустить сетевую подсистему командой:

```
root@server:/# systemctl restart networking
```

Это требуется для того, чтобы порождённые самой системой dns-запросы о локальных серверах отправлялись сразу к серверу виртуализации, а не перенаправлялись туда главным шлюзом.

## Защита от падений (автоматическое возобновление)

Для организации автоматического восстановления работоспособности службы после сбоя можно воспользоваться средствами системы инициализации systemd.

В первую очередь следует проверить текущие параметры службы командой:

```
root@server:/# systemctl show dnsmasq
Type=forking
Restart=no
PIDFile=/run/dnsmasq/dnsmasq.pid
NotifyAccess=none
RestartUsec=100ms
TimeoutStartUsec=1min 30s
TimeoutStopUsec=1min 30s
```

Как видно из отклика команды (он приведён здесь в сильно сокращённом виде, содержащем важные в текущем контексте строки), перезапуск службы отключён.

Затем следует выполнить редактирование службы при помощи команды:

```
root@server:/# systemctl edit dnsmasq
```

При этом будет открыт текстовый редактор, в котором необходимо внести определённые правки. В случае подтверждения сохранения изменений при выходе из текстового редактора в системе создаётся файл **/etc/systemd/system/dnsmasq.service.d/override.conf**, в котором эти поправки и хранятся. При очередном запуске службы они переопределяют исходные значения параметров. Для решения задачи перезапуска службы при сбоях достаточно следующего содержимого этого файла:

```
[Service]
Restart=on-failure
```

Стоит отметить, что такие изменения в службе сохраняются и при установке обновлений системы.

После внесения изменений следует перезапустить службу и убедиться, что изменения параметров учтены:

```
root@server:/# systemctl restart dnsmasq
root@server:/# systemctl show dnsmasq
Type=forking
Restart=on-failure
PIDFile=/run/dnsmasq/dnsmasq.pid
NotifyAccess=none
RestartUSec=100ms
TimeoutStartUSec=1min 30s
TimeoutStopUSec=1min 30s
```

## Проверка работоспособности и устойчивости

Проверка работоспособности службы тривиальна: выполняется подключение клиента по протоколу DHCP и на нём выполняется ряд DNS-запросов. В приведённом далее фрагменте терминального сеанса, выполненного на клиентском устройстве под управлением live-диска Xubuntu 19.04, демонстрируется факт подключения к беспроводной сети, получения IP-адреса, сетевой маски, сетевых маршрутов и адресов шлюза и DNS-сервера (ожидаемо совпадающих). Затем демонстрируется способность сервера беспроводной сети отвечать на DNS-запросы:

```
xubuntu@xubuntu:~$ ip address show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp13s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN
group default qlen 1000
    link/ether 54:be:f7:6f:8f:33 brd ff:ff:ff:ff:ff:ff
3: wlp14s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group
default qlen 1000
    link/ether 80:56:f2:08:71:7b brd ff:ff:ff:ff:ff:ff
    inet 192.168.4.34/24 brd 192.168.4.255 scope global dynamic wlp14s0
        valid_lft 43174sec preferred_lft 43174sec
    inet6 fe80::364a:99ac:fb9a:e6c4/64 scope link
        valid_lft forever preferred_lft forever
xubuntu@xubuntu:~$ ip route show
default via 192.168.4.1 dev wlp14s0 proto static metric 600
169.254.0.0/16 dev wlp14s0 scope link metric 1000
192.168.4.0/24 dev wlp14s0 proto kernel scope link src 192.168.4.34 metric 600
xubuntu@xubuntu:~$ systemd-resolve --status
Link 3 (wlp14s0)
    Current Scopes: DNS LLMNR/IPv4 LLMNR/IPv6
    LLMNR setting: yes
MulticastDNS setting: no
    DNSSEC setting: no
    DNSSEC supported: no
    DNS Servers: 192.168.4.1
    DNS Domain: wifi.school34
xubuntu@xubuntu:~$ host avers.school34
avers.school34 has address 192.168.0.4
```

```
xubuntu@xubuntu:~$ host library.service.school34
library.school34 has address 192.168.7.4
xubuntu@xubuntu:~$ host samba.service.school34
samba.school34 has address 192.168.7.2
```

В представленном фрагменте часть откликов команд сокращены в неинформативной части.

Проверка устойчивости системы к сбоям демонстрируется в следующем фрагменте терминального сеанса (отклики некоторых команд сокращены в неинформативной в применении к контексту части):

```
root@server:/# ps -ef | grep dnsmasq
dnsmasq  2061      1  0 15:18 ?                00:00:00 /usr/sbin/dnsmasq -x ...
root@server:/# kill -9 2061
root@server:/# ps -ef | grep dnsmasq
dnsmasq  2082      1  0 15:18 ?                00:00:00 /usr/sbin/dnsmasq -x
root@server:/# systemctl status dnsmasq
• dnsmasq.service - dnsmasq - A lightweight DHCP and caching DNS server
  Loaded: loaded (/lib/systemd/system/dnsmasq.service; enabled; vendor preset:
enabled)
  Drop-In: /etc/systemd/system/dnsmasq.service.d
           └─override.conf
  Active: active (running) since Fri 2020-02-07 15:18:52 MSK; 21s ago
  Process: 2073 ExecStartPre=/usr/sbin/dnsmasq --test (code=exited, status=0/SUCCESS)
  Process: 2074 ExecStart=/etc/init.d/dnsmasq systemd-exec (code=exited,
```

В этом фрагменте выясняется PID текущего процесса dnsmasq (2061), затем этот процесс подвергается умышленному завершению, после чего повторно выясняется PID процесса dnsmasq. Из того, что PID = 2082 существует и отличается от предыдущего, следует однозначный вывод, что служба была перезапущена после падения. Это подтверждается также и выводом последней команды.

## Служба точного времени (ntp)

### Настройка

В первую очередь требуется установить пакеты ntp и ntpdate со всеми зависимостями:

```
root@server:/# apt-get update
root@server:/# apt-get install ntp ntpdate
```

В состав этих пакетов входит ряд программ:

- ntpd – демон,
- ntpq – стандартная программа для запросов,
- ntpdc – расширенная программа для запросов,
- ntpdate – программа-клиент для установки времени в системе по ntp,
- sntp – простой сетевой клиент
- и другие (генераторы ключей, служебные, отладочные, симуляторы и др.)

Перед настройкой ntp.conf следует сначала настроить часовой пояс (файл **/etc/localtime**), что правильнее всего выполнить при помощи команды:

```
root@server:/# dpkg-reconfigure tzdata
```

Для настройки службы точного времени следует привести конфигурационный файл **/etc/ntp.conf** к виду:

```
# /etc/ntp.conf, configuration for ntpd; see ntp.conf(5) for help

driftfile /var/lib/ntp/ntp.drift

server ntp.school34      iburst prefer
server ntp1.vniiftri.ru iburst
server ntp2.vniiftri.ru iburst
server ntp3.vniiftri.ru iburst

interface ignore all
interface listen 192.168.0.44
interface listen 192.168.4.0/24

restrict default ignore
restrict ntp.school34      noquery notrap
restrict ntp1.vniiftri.ru noquery notrap
restrict ntp2.vniiftri.ru noquery notrap
restrict ntp3.vniiftri.ru noquery notrap
restrict 127.0.0.1
restrict ::1
restrict 192.168.4.0 mask 255.255.255.0 kod notrap nomodify nopeer noquery limited
```

Пояснения к конфигурационному файлу:

**driftfile** – файл для записи частотной коррекции аппаратных часов, обновляется демоном один раз в час;

**interface** – параметр, описывающий взаимодействие с сетевыми интерфейсами системы; из нескольких таких директив к сетевому пакету применяется последняя подходящая.

**server** – используемый для синхронизации внешний сервер, таких серверов может быть задано несколько на случай недоступности одного или нескольких из них; дополнительный параметр **prefer** означает, что в случае доступности приоритет использования должен быть отдан именно этому серверу.

**iburst** – отсылать по 8 пакетов за 2 секунды вместо одного, это позволяет синхронизироваться быстрее (за несколько секунд вместо нескольких минут), однако не подходит для сетей с низкой пропускной способностью;

**kod** – kiss of death – отправлять в ответ на пакет, нарушающий ограничения по нагрузке на сервис ответный пакет с уведомлением;

**notrap** – не реализовывать функционал определения положения хоста по IPv6;

**nomodify** – отклонять запросы, которые пытаются изменить состояние сервера, разрешены только запросы, которые лишь получают ответ;

**nopeer** – отклонять не авторизованные запросы на установление связи, не касается пакетов, которые не устанавливают связь, т. е. клиентов обслуживать, но не синхронизироваться с ними;

**restrict default** – эта строка задает ограничения по умолчанию, здесь по умолчанию все пакеты кроме далее явно обозначенных игнорируются;

**restrict** – ввод ограничения на хост или сеть (хост может быть задан как именем, так и адресом, бессмысленно задавать именем хост, имеющий несколько ip адресов, как,

например, `debian.pool.ntp.org`, т.к. в этом случае будет срабатывать правило по умолчанию), сеть задается своим адресом и маской, а далее следуют ключи, ограничивающие эту сеть или хост;

`limited` – отклонять запросы на синхронизацию, если превышены ограничения на трафик, заданные командой `discard` (там по умолчанию минимальный интервал между пакетами 1 секунда, а средний – 3 секунды), если при этом установлен еще и флаг `kod`, то отправляется ответный пакет;

`noquery` – отклонять запросы от `ntpq` и `ntpd`, сервис точного времени не затрагивается;

Таким образом, конфигурационный файл вынуждает `ntpd` работать по следующим правилам:

- 1. синхронизироваться разрешено только с заранее определенными ntp-серверами
- 2. по умолчанию игнорируются все пакеты, отправляемые к серверу, кроме явно разрешенных
- 3. внешним ntp-серверам не разрешено обращаться к локальному со служебными запросами или пользоваться IPv6
- 4. соединениям с самого сервера разрешено все, т.е. локально можно выполнять любые запросы к ntp-серверу, управлять им, отслеживать его состояние
- 5. локальные клиенты допускаются только из указанной подсети, они ограничены по пропускной способности, не могут влиять на сервер времени и посылать к нему служебные запросы, им разрешено лишь получать от сервера точное время

Остается лишь перезапустить подсистему точного времени командой

```
root@server:/# systemctl restart ntp
```

и убедиться, что служба находится в работоспособном состоянии при помощи команды

```
root@server:/# systemctl status ntp
```

При отсутствии ошибок вывод этой команды будет похож на:

```
• ntp.service - Network Time Service
Loaded: loaded (/lib/systemd/system/ntp.service; enabled; vendor preset: enabled)
Active: active (running) since Fri 2020-02-28 10:54:07 MSK; 3min 8s ago
Docs: man:ntpd(8)
Process: 1864 ExecStart=/usr/lib/ntp/ntp-systemd-wrapper (code=exited,
status=0/SUCCESS)
Main PID: 1870 (ntpd)
Tasks: 2 (limit: 4509)
Memory: 1.8M
CGroup: /system.slice/ntp.service
└─1870 /usr/sbin/ntpd -p /var/run/ntpd.pid -4 -g -u 122:127

фев 28 10:54:07 server ntpd[1864]: Command line: /usr/sbin/ntpd -p /var/run/ntpd.pid
-4 -g -u 122:127
фев 28 10:54:07 server systemd[1]: Started Network Time Service.
фев 28 10:54:07 server ntpd[1870]: proto: precision = 0.175 usec (-22)
фев 28 10:54:07 server ntpd[1870]: Listen and drop on 0 v4wildcard 0.0.0.0:123
фев 28 10:54:07 server ntpd[1870]: Listen normally on 1 lo 127.0.0.1:123
фев 28 10:54:07 server ntpd[1870]: Listen normally on 2 enp7s0 192.168.0.44:123
фев 28 10:54:07 server ntpd[1870]: Listen normally on 3 wlp6s0 192.168.4.1:123
фев 28 10:54:07 server ntpd[1870]: Listening on routing socket on fd #20 for
interface updates
фев 28 10:54:07 server ntpd[1870]: kernel reports TIME_ERROR: 0x2041: Clock
```

```
Unsyncronized
фев 28 10:54:07 server ntpd[1870]: kernel reports TIME_ERROR: 0x2041: Clock
Unsyncronized
```

В этом отклике следует отметить сообщения о том, что часы не синхронизированы (Clock Unsyncronized), что ожидаемо, т. к. служба запустилась только что, а для синхронизации требуется некоторое время. Следующие команды, выполненные через две минуты после перезапуска службы, демонстрируют успешность синхронизации:

```
root@server:/# ntpdate -q localhost
server 127.0.0.1, stratum 2, offset 0.000001, delay 0.02568
28 Feb 10:57:21 ntpdate[1882]: adjust time server 127.0.0.1 offset 0.000001 sec
root@server:/# ntpq -p
remote          refid st t when poll reach  delay  offset jitter
=====
*ntp.school34    .MRS.  1 u   4   64   17 28.163  0.193  5.616
+ntp1.vniiftri.ru .MRS.  1 u   3   64   17 28.331  0.856  0.970
+ntp2.vniiftri.ru .MRS.  1 u   5   64   17 30.655  2.230  0.694
```

В частности, здесь указано, что системные часы синхронизированы с локальным сервером с точностью до 0.000001 секунды, при этом локальный сервер имеет уровень stratum 2, отклонение его часов от часов вышестоящего сервера ntp.school34 0.193 миллисекунды, а дисперсия отклонений по результатам нескольких последних запросов составила 5.616 миллисекунд.

При заданных настройках демон ntpd тем не менее прослушивает все интерфейсы системы, но игнорирует пакеты согласно своему конфигурационному файлу. В том числе демон прослушивает и IPv6-адреса. В этом можно убедиться в выводе команды:

```
root@server:/# ss -ulp | grep ntp
UNCONN      0      0      192.168.4.1:ntp      0.0.0.0:*
UNCONN      0      0      192.168.0.44:ntp     0.0.0.0:*
UNCONN      0      0        127.0.0.1:ntp    0.0.0.0:*
UNCONN      0      0         0.0.0.0:ntp      0.0.0.0:*
UNCONN      0      0          [::]:ntp          [::]:*
```

Для принудительного отключения поддержки протокола IPv6 демоном следует привести конфигурационный файл **/etc/default/ntp** к виду:

```
NTPD_OPTS='-4 -g'
```

После перезапуска службы можно убедиться, что протокол IPv6 больше не используется:

```
root@server:/# systemctl restart ntp
root@server:/# ss -ul | grep ntp
UNCONN      0      0      192.168.4.1:ntp      0.0.0.0:*
UNCONN      0      0      192.168.0.44:ntp     0.0.0.0:*
UNCONN      0      0        127.0.0.1:ntp      0.0.0.0:*
UNCONN      0      0         0.0.0.0:ntp        0.0.0.0:*
```

Свой системный журнал служба ведет в файле **/var/log/daemon.log**, в который записывают свои сообщения и другие службы. Для получения выборки сообщений, связанных со службой ntp, за некоторый интервал времени (например 08:00-10:30 текущего дня) следует воспользоваться командой

```
root@server:/# journalctl -u ntp --since 08:00 --until 10:30
```

## Защита от падений (автоматическое возобновление)

Для организации автоматического восстановления работоспособности службы после сбоя можно воспользоваться средствами системы инициализации systemd.

В первую очередь следует проверить текущие параметры службы командой:

```
root@server:/# systemctl show ntp
Type=forking
Restart=no
NotifyAccess=none
RestartUSec=100ms
TimeoutStartUSec=1min 30s
TimeoutStopUSec=1min 30s
```

Как видно из отклика команды (он приведён здесь в сильно сокращённом виде, содержащем важные в текущем контексте строки), перезапуск службы отключён.

Затем следует выполнить редактирование службы при помощи команды:

```
root@server:/# systemctl edit ntp
```

При этом будет открыт текстовый редактор, в котором необходимо внести определённые правки. В случае подтверждения сохранения изменений при выходе из текстового редактора в системе создаётся файл `/etc/systemd/system/ntp.service.d/override.conf`, в котором эти поправки и хранятся. При очередном запуске службы они переопределяют исходные значения параметров. Для решения задачи перезапуска службы при сбоях достаточно следующего содержимого этого файла:

```
[Service]
Restart=on-failure
```

Стоит отметить, что такие изменения в службе сохраняются и при установке обновлений системы.

После внесения изменений следует перезапустить службу и убедиться, что изменения параметров учтены:

```
root@server:/# systemctl restart ntp
root@server:/# systemctl show ntp
Type=forking
Restart=on-failure
NotifyAccess=none
RestartUSec=100ms
TimeoutStartUSec=1min 30s
TimeoutStopUSec=1min 30s
```

## Проверка работоспособности и устойчивости

Проверка работоспособности службы тривиальна: с подключённого к беспроводной сети клиента (в данном случае ноутбук под управлением Debian 10, на котором умышленно искажено системное время) выполняется запрос на корректировку времени. Эти действия отражены в следующем фрагменте терминального сеанса:

```
root@debian:/# date --set="16:03"
Пт фев 28 16:03:00 MSK 2020
root@debian:/# /sbin/ntpdate 192.168.4.1
28 Feb 15:51:26 ntpdate[1393]: adjust time server 192.168.4.1 offset 0.000285 sec
```



Проверка устойчивости системы к сбоям демонстрируется в следующем фрагменте терминального сеанса (отклики некоторых команд сокращены в неинформативной в применении к контексту части):

```
root@server:/# systemctl status ntp
• ntp.service - Network Time Service
  Active: active (running) since Fri 2020-02-28 11:29:19 MSK; 4h 29min ago
Main PID: 2049 (ntpd)
  Tasks: 2 (limit: 4509)
  Memory: 1.7M
  CGroup: /system.slice/ntp.service
          └─2049 /usr/sbin/ntpd -p /var/run/ntpd.pid -4 -g -u 122:127
root@server:/# kill -9 2049
root@server:/# systemctl status ntp
• ntp.service - Network Time Service
  Active: active (running) since Fri 2020-02-28 15:58:46 MSK; 3s ago
Main PID: 2522 (ntpd)
  Tasks: 2 (limit: 4509)
  Memory: 1.7M
  CGroup: /system.slice/ntp.service
          └─2522 /usr/sbin/ntpd -p /var/run/ntpd.pid -4 -g -u 122:127
```

В этом фрагменте проверяется статус службы точного времени, из ответного сообщения выясняется PID текущего процесса ntpd (2049), затем этот процесс подвергается умышленному завершению, после чего повторно проверяется статус службы и выясняется PID вновь запущенного процесса ntpd (2522). Работоспособность службы также подтверждается повторной успешной синхронизацией с сервером того же клиента.

## Установка и настройка ssh

Установка ssh может быть выполнена как на этапе установки системы, так и после при помощи команд

```
root@server:/# apt-get update
root@server:/# apt-get install ssh
```

На данном сервере ssh используется как единственное средство удалённого входа и администрирования. Учитывая положение сервера в сети, ssh принимает соединения только на проводном (недоступном для клиентов) интерфейсе. Тем не менее, должны быть предприняты существенные меры по защите системы удалённого администрирования. Эти меры распределены на три эшелона: сетевой фильтр, базовую настройку сервера и, собственно, настройку самой службы ssh.

О настройке сетевого экранирования сказано ниже, о базовой настройке сервера – выше при описании списков доступа и статических арг-записей.

Касательно настройки самой службы следует выделить два направления: защита от несанкционированного доступа и обеспечение безотказного функционирования.

## Настройка

Основной конфигурационный файл службы **/etc/ssh/sshd\_config** следует привести к виду:

```
Port 22
ListenAddress 192.168.0.44
AddressFamily inet
```

```
Protocol 2
PermitRootLogin no
AllowUsers guest
PasswordAuthentication yes
PubkeyAuthentication no
KerberosAuthentication no
HostbasedAuthentication no
IgnoreRhosts yes
PermitEmptyPasswords no
X11Forwarding yes
```

Тем самым обеспечивается следующий порядок работы сервера ssh:

Port 22 – сервер принимает сообщения на 22 порту по протоколу tcp.

Protocol 2 – используется протокол ssh версии 2.

AddressFamily inet – разрешено использование протокола IPv4, использование IPv6 отключено.

PermitRootLogin no – удалённый вход в систему от имени суперпользователя запрещён.

AllowUsers guest – удалённый вход разрешён только для пользователя guest.

PasswordAuthentication yes – разрешена аутентификация только по паролю, остальные методы запрещены, т. к. не используются.

PermitEmptyPasswords no – запрещено использование пустых паролей.

X11Forwarding yes – разрешена передача трафика по протоколу X11, т. к. он используется сервером.

После приведения этого файла к заданному виду следует перезапустить службу ssh, затем убедиться, что она успешно запущена и принимает соединения только по протоколу IPv4. Это можно сделать следующими командами:

```
root@debian:/# systemctl restart sshd
root@debian:/# systemctl status sshd
• ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
  Active: active (running) since Mon 2020-02-03 16:13:52 MSK; 8s ago
    Docs: man:sshd(8)
          man:sshd_config(5)
  Process: 1832 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 1833 (sshd)
   Tasks: 4 (limit: 4509)
  Memory: 7.4M
  CGroup: /system.slice/ssh.service
          └─1248 sshd: guest [priv]
            └─1304 sshd: guest@pts/0
              └─1307 -bash
                └─1833 /usr/sbin/sshd -D
```

```
фев 03 16:13:52 server systemd[1]: This usually indicates unclean termination of a previous run, or service implementation deficiencies.
```

```
фев 03 16:13:52 server systemd[1]: Starting OpenBSD Secure Shell server...
```

```
фев 03 16:13:52 server sshd[1833]: Server listening on 192.168.0.44 port 22.
```

```
фев 03 16:13:52 server systemd[1]: Started OpenBSD Secure Shell server.
```

```
root@debian:/# ss -Htulp | grep ssh
```

```
tcp LISTEN 0 128 192.168.0.44:ssh 0.0.0.0:* users: ("sshd",pid=1833,fd=3))
```

Свой системный журнал служба ведёт в файле `/var/log/daemon.log`, в который записывают свои сообщения и другие службы. Для получения выборки сообщений, связанных со службой `ssh`, за некоторый интервал времени (например за текущий день) следует воспользоваться командой

```
root@server:/#journalctl -u ssh --since today
```

## Защита от падений (автоматическое возобновление)

Управлением службами в системе занимается менеджер загрузки и служб `systemd`, именно он и обеспечивает (при должной настройке) автоматическое возобновление служб в случае их падения. В рассматриваемой системе такая настройка выполнена изначально сборщиками дистрибутива операционной системы.

## Проверка работоспособности и устойчивости

Для проверки состояния службы можно воспользоваться командой

```
root@debian:/# systemctl status ssh
```

или командой

```
root@debian:/# service ssh status
```

которая является, по сути, обёрткой вышеприведённой. На рассматриваемой системе в силу значения переменной окружения `PATH` последняя команда должна выполняться в следующей форме (если не дополнять значение `PATH`):

```
root@debian:/# /sbin/service ssh status
```

Чтобы проверить, какие порты, адреса и интерфейсы прослушивает `ssh`, можно воспользоваться командой

```
root@debian:/# ss -Htulp | grep ssh
```

Следующий фрагмент терминального сеанса демонстрирует попытки входа в систему удалённо с разрешённого хоста под разными учётными записями:

```
administrator@admin:~$ ssh guest@server.wifi.school34
guest@server.wifi.school34's password:
Last login: Fri Feb 21 14:01:07 2020 from 192.168.0.2
```

```
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
```

```
guest@server:~$ logout
```

```
Connection to server.wifi.school34 closed.
```

```
administrator@admin:~$ ssh root@server.wifi.school34
```

```
root@server.wifi.school34's password:
```

```
Permission denied, please try again.
```

```
root@server.wifi.school34's password:
```

```
Permission denied, please try again.
```

```
root@server.wifi.school34's password:
```

```
root@server.wifi.school34: Permission denied (password,keyboard-interactive).
administrator@admin:~$
```

Как видно из этого фрагмента, вход от имени guest выполнен успешно, а вход от имени root не выполнен даже несмотря на ввод правильного пароля.

При попытке входа с хоста, отличного от разрешённого, клиент получает сообщение (при отключённом файерволе рассматриваемого сервера):

```
xubuntu@xubuntu:~$ ssh guest@server.wifi.school34
ssh_exchange_identification: read: Connection reset by peer
```

Это является результатом работы списков контроля доступа. При включённом файерволе клиент получает иное сообщение:

```
xubuntu@xubuntu:~$ ssh guest@server.wifi.school34
ssh: connect to host guest@server.wifi.school34 port 22: Connection timed out
```

Однако в обоих случаях в подключении к серверу отказано.

Один из способов проверки автоматического восстановления работоспособности приведён в следующем терминальном сеансе:

```
root@server:/# date
Пт фев 21 16:11:47 MSK 2020
root@server:/# ps -ef | grep ssh
root      708      1   0 11:14 ?           00:00:00 /usr/sbin/sshd -D
guest    1192    1159   0 11:14 ?           00:00:00 /usr/bin/ssh-agent /usr/bin/startkde
root     4075    4053   0 16:11 pts/2     00:00:00 grep ssh
root@server:/# kill -9 708
root@server:/# ps -ef | grep ssh
guest    1192    1159   0 11:14 ?           00:00:00 /usr/bin/ssh-agent /usr/bin/startkde
root     4083      1   0 16:12 ?           00:00:00 /usr/sbin/sshd -D
root     4086    4053   0 16:12 pts/2     00:00:00 grep ssh
root@server:/# systemctl status sshd
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2020-02-21 16:12:29 MSK; 15s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Process: 4082 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 4083 (sshd)
    Tasks: 1 (limit: 4509)
   Memory: 1.1M
    CGroup: /system.slice/ssh.service
            └─4083 /usr/sbin/sshd -D

фев 21 16:12:29 server systemd[1]: Starting OpenBSD Secure Shell server...
фев 21 16:12:29 server sshd[4083]: Server listening on 192.168.0.44 port 22.
фев 21 16:12:29 server systemd[1]: Started OpenBSD Secure Shell server.
root@server:/#
```

Здесь сначала отображены системные дата и время, затем получен список выполняющихся процессов в системе, отфильтрованный по упоминанию ssh. Как можно видеть из этого списка, в системе работает процесс sshd с pid 708. Следующая команда убивает этот процесс. Далее демонстрируется, что процесс действительно убит, но уже запущен новый процесс с PID 4083 и служба успешно перезапущена.

## О дополнительных службах

### Защита от подбора паролей (fail2ban)

Поскольку в силу характера функционирования сервера единственным механизмом удалённого входа в систему является ssh, то от использования fail2ban вполне безопасно отказаться. Система хорошо защищена файерволом (описано далее) с указанием (как по ip, так и по mac-адресу) для него единственного хоста, с которого разрешен вход по ssh. Сам ssh-сервер системы имеет такие же ограничения, кроме того предприняты меры по защите от агр-атак (статическая запись), сам удалённый вход разрешен только для непривилегированного пользователя, учётная запись которого защищена паролем, а повышение привилегий требует знания пароля суперпользователя.

### Ротация журналов (logrotate)

Системные журналы могут служить средством атаки на сервер с целью заполнить его дисковое пространство и парализовать, тем самым, его работу. Этому противостоит система ротации журнальных файлов.

При входе в систему по протоколу ssh соответствующая запись вносится в файл `/var/log/auth.log`. Учитывая предпринятые меры защиты ssh, можно утверждать, что атака возможна только с единственного не менее защищённого хоста из ядра сети (компьютер системного администратора).

Остальные доступные клиентам службы (dnsmasq и ntp) не ведут собственных журналов клиентских обращений и, тем самым, не могут быть направлением для такой атаки.

Как итог следует отметить, что настройки системы по умолчанию (ротация еженедельно) здесь вполне применимы.

### Служба точного времени (ntp)

Сервер службы точного времени не только предоставляет услуги клиентам, но и следит за точностью системных часов машины, на которой развёрнут. Поэтому невозможна и бессмысленна работа других средств синхронизации часов в описываемой системе. В частности, простейший ntp-клиент, который входит в состав systemd, оказывается после установки и настройки ntpd в неработоспособном состоянии и должен быть отключён:

```
root@server:/# systemctl status systemd-timesyncd
• systemd-timesyncd.service - Network Time Synchronization
  Loaded: loaded (/lib/systemd/system/systemd-timesyncd.service; enabled; vendor
  preset: enabled)
  Drop-In: /usr/lib/systemd/system/systemd-timesyncd.service.d
           └─disable-with-time-daemon.conf
  Active: inactive (dead)
  Condition: start condition failed at Mon 2020-03-02 10:09:05 MSK; 15min ago
             └─ ConditionFileIsExecutable=!/usr/sbin/ntpd was not met
  Docs: man:systemd-timesyncd.service(8)
```

```
map 02 10:09:05 server systemd[1]: Condition check resulted in Network Time
Synchronization being skipped.
```

```
root@server:/# systemctl disable systemd-timesyncd
Removed /etc/systemd/system/dbus-org.freedesktop.timesync1.service.
Removed /etc/systemd/system/sysinit.target.wants/systemd-timesyncd.service.
```

## Почтовая служба

Почтовая служба, даже локальная, может стать направлением для атаки: так, например, злоумышленник может генерировать огромный вал писем локально в системе, чем спровоцировать переполнение файловой системы и отказ обслуживания. Поэтому даже локальную почтовую службу следует тщательно настраивать или отключать вовсе, если она не используется.

В рассматриваемой системе вместе с другими компонентами окружения рабочего стола установлен агент пересылки почты (MTA) `exim4`. В перспективе планируется введение в эксплуатацию внутреннего почтового сервера организации, что позволит многим системным службам (например, описанной далее системе мониторинга состояния жёсткого диска) уведомлять системного администратора о проблемах. При этом использование MTA пользователями не предполагается: на машине с общим доступом гораздо разумнее и практичнее пользоваться веб-интерфейсом почтового сервера организации.

С точки зрения безопасности MTA может быть использован для переполнения почтовых ящиков пользователей, т. е. заполнения дискового пространства машины.

На данном этапе имеет смысл полностью удалить MTA из системы, тем более что это не ведет к удалению других компонентов системы. Итак, следующий сокращённый в незначительном выводе фрагмент терминального сеанса демонстрирует процедуру проверки возможности удаления и непосредственно удаления MTA:

```
root@server:/# dpkg -l | grep exim
ii  exim4-base 4.92-8+deb10u3 amd64 support files for all Exim MTA (v4) packages
ii  exim4-config 4.92-8+deb10u3 all configuration for the Exim MTA (v4)
ii  exim4-daemon-light 4.92-8+deb10u3 amd64 lightweight Exim MTA (v4) daemon
root@server:/# apt-get -s purge exim4-daemon-light
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Следующие пакеты устанавливались автоматически и больше не требуются:
  exim4-base exim4-config keyutils libgnutls-dane0 libnfsidmap2 libunbound8
Для их удаления используйте «apt autoremove».
Следующие пакеты будут УДАЛЕНЫ:
  exim4-daemon-light*
Обновлено 0 пакетов, установлено 0 новых пакетов, для удаления отмечено 1 пакетов, и
117 пакетов не обновлено.
Purge exim4-daemon-light [4.92-8+deb10u3]
root@server:/# apt-get purge exim4-daemon-light
Чтение списков пакетов... Готово
...
root@server:/# apt autoremove
...
Следующие пакеты будут УДАЛЕНЫ:
  exim4-base exim4-config keyutils libgnutls-dane0 libnfsidmap2 libunbound8
Обновлено 0 пакетов, установлено 0 новых пакетов, для удаления отмечено 6 пакетов, и
116 пакетов не обновлено.
...
root@server:/#
```

Так как некоторые системные службы используют в своей работе почтовую систему, следует убедиться, что их работоспособность не нарушена. Так после успешной перезагрузки системы (в смысле отсутствия сообщений об ошибках запуска служб)

следует проверить их состояние. На рассматриваемой системе такими службами являются cron и smartd (служба описана ниже):

```
root@server:/# systemctl status cron
```

```
• cron.service - Regular background program processing daemon
  Loaded: loaded (/lib/systemd/system/cron.service; enabled; vendor preset: enabled)
  Active: active (running) since Fri 2020-03-13 16:50:49 MSK; 7min ago
```

```
...
map 13 16:51:15 server CRON[538]: (CRON) info (No MTA installed, discarding output)
...
```

```
root@server:/# systemctl status smartd
```

```
• smartd.service - Self Monitoring and Reporting Technology (SMART) Daemon
  Loaded: loaded (/lib/systemd/system/smartd.service; enabled; vendor preset:
enabled)
```

```
  Active: active (running) since Fri 2020-03-13 16:50:49 MSK; 7min ago
```

```
...
```

Несущественный в рассматриваемом контексте вывод команд здесь также сокращён.

В перспективе после развёртывания внутреннего почтового сервера организации МТА вновь может быть установлен в системе и настроен соответствующим образом.

## Прикладное программное обеспечение

Несмотря на то, что рассматриваемый ноутбук исполняет роль сервера беспроводной подсети, он не задействует большую часть своих вычислительных ресурсов. Кроме того, самым логичным решением по отображению регулярно меняющегося пароля беспроводной сети является использование собственного экрана ноутбука. Поэтому для него предусмотрена возможность выполнять роль рабочей станции. При этом требуется обеспечить сохранение приемлимого состояния графического рабочего окружения, а также гарантировать сохранение работоспособности системы независимо от действий непривилегированного пользователя. Также необходимо разместить в системе достаточный комплект прикладного программного обеспечения.

## Комплект прикладных программ и графическое окружение

В качестве окружения рабочего стола выбрана Plasma5 для сохранения общности с другими linux-системами организации. Для работы в роли рабочей станции на ноутбук установлено среди прочего следующее программное обеспечение:

- офисный комплект LibreOffice 6.1;
- офисный комплект Calligra Suite 3.1.0;
- программа для просмотра pdf-документов Okular 1.3.2;
- универсальная программа работы со сканером XSane 0.999;
- система оптического распознавания символов OCRFeeder 0.8.1;
- электронный словарь GoldenDict 1.5.0;
- настольная издательская программа Scribus 1.4.8;
- растровый графический редактор Gimp 2.10.8;
- векторный графический редактор Inkscape 0.92;
- интернет-обозреватель Mozilla Firefox 68;
- интернет-обозреватель Chromium 78;
- звуковой редактор Audacity 2.2.2;
- звуковой редактор Kwave 18.08;

- проигрыватель мультимедиа vlc 3.0.8;
- программа для записи и копирования оптических дисков K3b 18.08;
- клиент для подключения к удалённому рабочему столу KRDC 18.04.

Также установлен большой набор вспомогательных программных средств: простые текстовые редакторы, калькулятор, календарь, программы для захвата видео с web-камеры и экрана, программа для создания снимков экрана и т. д.

Для удобства использования проведена настройка графического пользовательского интерфейса, в том числе:

- общая настройка интерфейса;
- настроен наиболее оптимальный план электропитания – отключено выключение системы при простое, переход в ждущий режим, гашение экрана и т. д.;
- ярлыки наиболее часто используемых программ вынесены на панель задач;
- ярлыки сетевых сервисов (файловый сервер, АИС «Аверс» и т. д.) вынесены на рабочий стол;
- при помощи виджета RichTextViewer обеспечено отображение текущего пароля беспроводной сети;
- выполнен первый запуск наиболее часто используемых программ;
- обеспечена автоматическая смена обоев рабочего стола по временам года и перед праздниками;
- обеспечено автоматическое выключение сервера по завершении рабочего дня.

Некоторые из этих мер, реализация которых была нетривиальна, описаны далее.

## Автоматическое восстановление настроек интерфейса и программ

Автоматическое восстановление настроек интерфейса и программ реализовано при помощи заготовленного архива с эталонными настройками и сценария, разработанного для замещения текущих настроек таковыми из эталонного архива. Одновременно с восстановлением настроек выполняется также очистка локального кэша, хранимого в домашнем каталоге пользователя, журналов браузеров и всех созданных пользователем guest файлов. На последнее особенно следует обратить внимание – никакие созданные пользователем документы, никакие его загрузки не сохраняются после работы сценария. Это сделано с тем, что система работает в гостевом режиме и не ассоциирована ни с одним конкретным пользователем. В случае, если имя пользователя в системе – не guest, происходит только восстановление настроек без удаления файлов.

Всё это реализовано при помощи сценария `restore-settings` собственной разработки. Он установлен в систему в виде одноимённого пакета из локального репозитория. Сам сценарий расположен в файле `/usr/bin/restore-settings`, подробная инструкция к нему встроена в пакет в виде man-страницы на русском и английском языках. Получить пакет можно по ссылке <https://sourceforge.net/projects/restoresettings/>.

Вызов сценария выполняется при каждой перезагрузке системы при помощи демона `cron`, для чего в его конфигурационный файл `/etc/crontab` внесены строки:

```
#Restoring GUI settings, clearing cache, removing guest's files
@reboot root restore-settings guest /home/guest /opt/templates/guest.tar guest 0755
```

Также следует отметить, что такое восстановление настроек не мешает как отображению меняющегося пароля беспроводной сети, так и смене обоев рабочего стола. Дело в том,



что принудительному восстановлению подвергаются лишь файлы в домашнем каталоге пользователя `guest`, а там расположены лишь постоянные ссылки на действительно изменяющееся содержимое. О файле, в котором хранится пароль, сказано выше в разделе «Организация беспроводной сети», что же касается обоев рабочего стола, то в их качестве выбран файл `/usr/local/share/wallpapers/current.jpg`, который является жёсткой ссылкой на одно из изображений в том же каталоге. Изменяя эту ссылку по расписанию, удаётся добиться желаемого эффекта. Некоторое затруднение могут доставлять лишь файлы, хранимые в системном кэше KDE (Plasma5), но их достаточно очищать одновременно с изменением ссылки. Для организации всего вышеперечисленного достаточно разместить в каталоге задания демона `cron` следующие два файла:

```
root@server:/# cat /etc/cron.d/wallpapers
#Changing wallpapers by seasons and holidays
* * 29-31 8 * root ln -f /usr/local/share/wallpapers/1september.jpg
    /usr/local/share/wallpapers/current.jpg
* * 1 9 * root ln -f /usr/local/share/wallpapers/1september.jpg
    /usr/local/share/wallpapers/current.jpg
root@server:/# cat /etc/cron.d/guest
#Changing wallpapers by seasons and holidays
* * 29-31 8 * root rm -rf /var/tmp/kdecache-guest
* * 1 9 * root rm -rf /var/tmp/kdecache-guest
```

Здесь оба файла представлены в сокращённом виде двумя строками каждый (в первом файле строки ввиду формата страницы разбиты на две).

## Отображение текущего пароля

Отображение текущего пароля доступа к беспроводной сети, как уже было сказано выше, выполняется при помощи `plasma5`-виджета `RichTextViewer`, т. е. крупным шрифтом на контрастном фоне прямо на рабочем столе, что позволяет клиентам, находящимся в одном помещении с сервером беспроводной сети, легко узнавать его без лишних действий.

## Автоматическое выключение системы

Автоматическое выключение системы в нерабочее время также выполняется средствами демона `cron`, для чего в его конфигурационный файл `/etc/crontab` внесены строки:

```
#Shutting down at night
50 20 * * * root shutdown +10
```

Они обеспечивают выключение системы в 21:00 с уведомлением пользователей за 10 минут до этого. Как следствие при следующем включении компьютера обеспечивается очистка домашнего каталога, восстановление настроек, смена пароля беспроводной сети. Очевидно, что выключение системы на ночь преследует три цели:

- энергосбережение;
- экономия ресурса аппаратной платформы;
- противодействие длительным атакам на подбор пароля.

## Установка и настройка сетевого фильтра (nftables)

### Сетевое окружение и потенциальные угрозы

Составление свода правил сетевого фильтра следует начинать с выяснения всех протоколов и портов, используемых системой и клиентами. Список сетевых портов и протоколов, на которых система готова принимать соединения, можно получить следующим образом:

```
root@server:/tmp# ss -tuln
```

| Netid | State  | Recv-Q | Send-Q | Local Address:Port | Peer Address:Port |
|-------|--------|--------|--------|--------------------|-------------------|
| udp   | UNCONN | 0      | 0      | 127.0.0.1:53       | 0.0.0.0:*         |
| udp   | UNCONN | 0      | 0      | 192.168.4.1:53     | 0.0.0.0:*         |
| udp   | UNCONN | 0      | 0      | 0.0.0.0:67         | 0.0.0.0:*         |
| udp   | UNCONN | 0      | 0      | 192.168.0.44:123   | 0.0.0.0:*         |
| udp   | UNCONN | 0      | 0      | 192.168.4.1:123    | 0.0.0.0:*         |
| udp   | UNCONN | 0      | 0      | 127.0.0.1:123      | 0.0.0.0:*         |
| udp   | UNCONN | 0      | 0      | 0.0.0.0:123        | 0.0.0.0:*         |
| udp   | UNCONN | 0      | 0      | 0.0.0.0:41410      | 0.0.0.0:*         |
| udp   | UNCONN | 0      | 0      | 0.0.0.0:631        | 0.0.0.0:*         |
| tcp   | LISTEN | 0      | 32     | 127.0.0.1:53       | 0.0.0.0:*         |
| tcp   | LISTEN | 0      | 32     | 192.168.4.1:53     | 0.0.0.0:*         |
| tcp   | LISTEN | 0      | 128    | 192.168.0.44:22    | 0.0.0.0:*         |
| tcp   | LISTEN | 0      | 5      | 127.0.0.1:631      | 0.0.0.0:*         |

Следует сделать ряд замечаний:

Во-первых, следует закрыть сетевым фильтром порт 67, работающий по протоколу UDP, для всех входящих через проводной интерфейс пакетов. Этот порт используется протоколом DHCP, который, согласно постановке задачи, должен обслуживать лишь клиентов беспроводной сети. Этот порт открыт демоном `dnsmasq`, который также выполняет такую проверку, однако дополнительная защита файерволом лишней явно не будет.

Во-вторых, требуется для обоих интерфейсов на данном этапе закрыть порт 631 для протоколов `tcp` и `udp`. Доступным этот порт должен остаться лишь для локального петлевого интерфейса. Более того, даже если позднее в рамках системы будет реализован сервер печати для клиентов беспроводной сети, порт протокола UDP для обоих интерфейсов открывать излишне: через него демон `cups-browsed` готов принять сведения о доступных в сети принтерах. Но с учётом структуры сети если таковые и будут реализованы, то сведения о них будут внесены в систему на постоянной основе.

В-третьих, не лишним будет закрыть для новых соединений по протоколу точного времени (порт 123) проводной интерфейс, не полагаясь только на устойчивость самого демона `ntpd`. При этом способность самого демона инициировать соединения с вышестоящими серверами точного времени не пострадает.

В-четвертых, для снижения вероятности атак на отказ в обслуживании, следует ограничить максимальное количество пакетов в единицу времени для каждой службы.

Наконец, следует обеспечить максимальную защиту критической для безопасности системы службы `ssh`, а именно, допускать установление соединений с ней только с проверкой `ip`-адреса, аппаратного адреса, интерфейса, с которого пришёл пакет и с соблюдением ограничения на количество таких пакетов в единицу времени.

Отдельно следует рассмотреть вопрос о транзите пакетов. Во избежание попыток обхода ограничений, накладываемых на беспроводную сеть постановкой задачи, следует оставить возможность транзита сетевых пакетов только к серверу АИС «Аверс» (по tcp-порту 8082), к виртуальной подсети серверов организации (без ограничений по портам на данном этапе) и в Интернет (только к портам из определённого списка, далее в этом разделе они будут называться «белыми»).

При этом следует понимать, что под Интернетом понимается вообще любой сетевой адрес, кроме вышеобозначенных, в том числе и адреса из других локальных подсетей, о которых серверу беспроводной сети вообще ничего не известно. Казалось бы, что злоумышленник, зная структуру сети организации, сможет отправить пакет на один из таких «белых» портов к некоторому компьютеру, например, административной подсети. Однако, в силу того, что сервер беспроводной сети не имеет маршрута в эту подсеть в своей таблице маршрутизации, а также в силу того, что прямое направление пакета к серверу административной подсети через его адрес в ядре сети невозможно, единственным маршрутом останется главный шлюз, файервол которого пресечёт отправку опасного пакета. Кроме того, межсетевое экранирование организовано и на сервере атакуемой подсети.

Тем не менее, даже при такой организации сетевого фильтра нельзя быть уверенным в его непроницаемости. Отказ от внесения в список «белых» портов порта 53 (DNS) не гарантирует, что некоторый злоумышленник, имеющий возможность подключиться к беспроводной сети в качестве клиента, не попытается обойти такое правило, например, следующим образом: организовать вне школьной сети собственный, доступный из Интернета DNS-сервер, принимающий запросы не на стандартном порту 53, а, например, на порту 443 (https). Файервол сервера беспроводной сети окажется в этом случае бессислен. Возможна и более простая атака: злоумышленник имеет запись об интересующем его запрещённом ресурсе в dns-кэше своего устройства и соединяется с ним, не пользуясь услугами сервера доменных имён. Ещё проще злоумышленнику обратиться к внешнему ресурсу по URL, содержащему ip-адрес этого ресурса. По той же причине нельзя исключать и обращение ко внешним прокси-серверам, например, файервол никак не может определить, что скрывается за, например, 198.51.100.34:443. Это может быть и веб-интерфейс почтового сервера, и сайт библиотеки, и прокси-сервер, принимающий соединения на таком порту.

## **Структура сетевого фильтра**

Опираясь на вышесказанное и руководствуясь принципом запрета по умолчанию, можно так охарактеризовать структуру сетевого фильтра:

- определён набор портов, которые считаются безопасными,
- определены таблицы фильтрации для протоколов IPv4 и IPv6,
- в каждой из таблиц определены цепочки правил для входящих, исходящих и транзитных пакетов,
- для всех цепочек таблицы IPv6 задана политика сброса пакетов по умолчанию,
- для входящей и транзитной цепочек таблицы IPv4 определена политика сброса пакетов по умолчанию, для исходящей цепочки по умолчанию установлена доверительная политика,
- разрешены приём и транзит пакетов, относящихся к уже установленным соединениям,

- наконец внесены разрешающие правила для описанного выше трафика.

В итоге сценарий атомарной загрузки правил приобрёл вид:

```
root@server:/# cat /etc/nftables.conf
```

```
#!/usr/sbin/nft -f
```

```
flush ruleset
```

```
define safe_ports = {http, https, ftp, ftp-data, ftps, ftps-data}
```

```
table ip filter {
    chain input {
        type filter hook input priority 0;
        policy drop;
        iif lo accept
        ct state established,related accept
        iif enp7s0 icmp type echo-request limit rate 10/second accept
        iif wlp6s0 icmp type echo-request limit rate 100/second accept
        iif enp7s0 ether saddr 90:2b:34:48:08:b5 ip saddr 192.168.0.2 tcp dport ssh ct
state new limit rate 1/second accept
        iif wlp6s0 udp dport bootps limit rate 300/second accept
        iif wlp6s0 udp dport domain limit rate 300/second accept
        iif wlp6s0 tcp dport domain limit rate 300/second accept
        iif wlp6s0 udp dport ntp limit rate 300/second accept
    }
    chain output {
        type filter hook output priority 0;
        policy accept;
    }
    chain forward {
        type filter hook forward priority 0;
        policy drop;
        ct state established,related accept
        ip saddr 192.168.4.0/24 ip daddr 192.168.7.0/24 accept
        ip saddr 192.168.4.0/24 ip daddr 192.168.0.4 tcp dport 8082 accept
        ip saddr 192.168.4.0/24 tcp dport $safe_ports accept
    }
}

table ip6 filter {
    chain input {
        type filter hook input priority 0;
        policy drop;
    }
    chain output {
        type filter hook output priority 0;
        policy drop;
    }
    chain forward {
        type filter hook forward priority 0;
        policy drop;
    }
}
```

## Запуск и проверка сетевого фильтра

Автоматическая загрузка правил файервола обеспечена установщиком пакета nftables и не требует ручного вмешательства. Проверить текущий набор правил можно командой

```
root@debian:/# /usr/sbin/nft list ruleset
```

Вывод этой команды с точностью до комментариев, инструкций предварительной очистки и подстановки списка разрешённых портов должен совпадать с приведённым выше сценарием загрузки правил файервола.

Убедиться в том, что правила были загружены до запуска сетевых интерфейсов, можно просмотрев содержимое файла **/var/log/daemon.log**, выдержка из которого с некоторыми сокращениями, но с сохранением порядка следования представленных, имеет вид:

```
May 14 09:11:28 server systemd[1]: Starting nftables...
May 14 09:11:28 server systemd[1]: Starting Show Plymouth Boot Screen...
May 14 09:11:28 server systemd[1]: Started nftables.
...
May 14 09:11:28 server systemd[1]: Reached target Local File Systems.
May 14 09:11:28 server systemd[1]: Starting Raise network interfaces...
May 14 09:11:28 server systemd[1]: Reached target System Initialization.
May 14 09:11:33 server systemd[1]: Started Raise network interfaces.
...
May 14 09:11:33 server systemd[1]: Reached target Network.
...
```

Также посмотреть состояние службы можно командой:

```
root@server:/# systemctl status nftables
```

```
• nftables.service - nftables
  Loaded: loaded (/lib/systemd/system/nftables.service; enabled; vendor preset:
enabled)
  Active: active (exited) since Thu 2020-05-14 09:11:28 MSK; 19min ago
  Docs: man:nft(8)
        http://wiki.nftables.org
  Process: 1633 ExecStart=/usr/sbin/nft -f /etc/nftables.conf (code=exited,
status=0/SUCCESS)
  Main PID: 1633 (code=exited, status=0/SUCCESS)
```

```
мая 14 09:11:28 server systemd[1]: Starting nftables...
мая 14 09:11:28 server systemd[1]: Started nftables.
```

Проверка работоспособности межсетевого экрана может быть разделена на тривиальную проверку доступности клиенту разрешённых сервисов, в том числе и расположенных вне беспроводной подсети, и проверку сетевым сканером nmap доступности тех или иных портов сервера с трёх разных направлений:

- с компьютера системного администратора
- с произвольного компьютера из ядра сети
- с клиентского компьютера из беспроводной подсети

Тестирование автоматизировано при помощи сценария **/tmp/nmap.sh**, содержимое которого прозрачно угадывается из его вывода, т. к. выполняемые команды отображаются с префиксом **user@host**.

Протокол сканирования с компьютера системного администратора:

```
root@admin:/tmp# chmod a+x nmap.sh
```

```
root@admin:/tmp# ./nmap.sh
```

Пинг-сканирование

```
user@host:/# nmap -sP 192.168.0.44
```

```
Starting Nmap 7.70 ( https://nmap.org ) at 2020-05-25 12:00 MSK
```

```
Nmap scan report for server.wifi.school34 (192.168.0.44)
```

```
Host is up (0.00033s latency).
```

```
MAC Address: B8:70:F4:29:E5:83 (Compal Information (kunshan))
```

Nmap done: 1 IP address (1 host up) scanned in 0.30 seconds

**TCP SYN пинг**

**user@host:/# nmap -PS 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:00 MSK

Nmap scan report for server.wifi.school34 (192.168.0.44)

Host is up (0.00029s latency).

Not shown: 999 filtered ports

PORT STATE SERVICE

22/tcp open ssh

MAC Address: B8:70:F4:29:E5:83 (Compal Information (kunshan))

Nmap done: 1 IP address (1 host up) scanned in 10.64 seconds

**TCP ACK пинг**

**user@host:/# nmap -PA 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:00 MSK

Nmap scan report for server.wifi.school34 (192.168.0.44)

Host is up (0.00031s latency).

Not shown: 999 filtered ports

PORT STATE SERVICE

22/tcp open ssh

MAC Address: B8:70:F4:29:E5:83 (Compal Information (kunshan))

Nmap done: 1 IP address (1 host up) scanned in 12.05 seconds

**UDP пинг**

**user@host:/# nmap -PU 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:00 MSK

Nmap scan report for server.wifi.school34 (192.168.0.44)

Host is up (0.00032s latency).

Not shown: 999 filtered ports

PORT STATE SERVICE

22/tcp open ssh

MAC Address: B8:70:F4:29:E5:83 (Compal Information (kunshan))

Nmap done: 1 IP address (1 host up) scanned in 12.11 seconds

**Различные типы пинг-пакетов ICMP**

**user@host:/# nmap -PE 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:00 MSK

Nmap scan report for server.wifi.school34 (192.168.0.44)

Host is up (0.00031s latency).

Not shown: 999 filtered ports

PORT STATE SERVICE

22/tcp open ssh

MAC Address: B8:70:F4:29:E5:83 (Compal Information (kunshan))

Nmap done: 1 IP address (1 host up) scanned in 12.08 seconds

**user@host:/# nmap -PP 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:00 MSK

Nmap scan report for server.wifi.school34 (192.168.0.44)

Host is up (0.00027s latency).

Not shown: 999 filtered ports

PORT STATE SERVICE

22/tcp open ssh

MAC Address: B8:70:F4:29:E5:83 (Compal Information (kunshan))

Nmap done: 1 IP address (1 host up) scanned in 4.46 seconds

**user@host:/# nmap -PM 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:00 MSK

Nmap scan report for server.wifi.school34 (192.168.0.44)

Host is up (0.00029s latency).

Not shown: 999 filtered ports

PORT STATE SERVICE

22/tcp open ssh

MAC Address: B8:70:F4:29:E5:83 (Compal Information (kunshan))

Nmap done: 1 IP address (1 host up) scanned in 10.42 seconds

**Пинг с использованием протокола IP**

**user@host:/# nmap -PO 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:01 MSK

Nmap scan report for server.wifi.school34 (192.168.0.44)

Host is up (0.00029s latency).

Not shown: 999 filtered ports

PORT STATE SERVICE

22/tcp open ssh

MAC Address: B8:70:F4:29:E5:83 (Compal Information (kunshan))

Nmap done: 1 IP address (1 host up) scanned in 4.48 seconds

**ARP пинг**

**user@host:/# nmap -PR 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:01 MSK

Nmap scan report for server.wifi.school34 (192.168.0.44)

Host is up (0.00031s latency).

Not shown: 999 filtered ports

PORT STATE SERVICE

22/tcp open ssh

MAC Address: B8:70:F4:29:E5:83 (Compal Information (kunshan))

Nmap done: 1 IP address (1 host up) scanned in 18.38 seconds

**TCP connect сканирование**

**user@host:/# nmap -sT 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:01 MSK

Nmap scan report for server.wifi.school34 (192.168.0.44)

Host is up (0.00036s latency).

Not shown: 999 filtered ports

PORT STATE SERVICE

22/tcp open ssh

MAC Address: B8:70:F4:29:E5:83 (Compal Information (kunshan))

Nmap done: 1 IP address (1 host up) scanned in 4.85 seconds

**UDP сканирование**

**user@host:/# nmap -sU 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:01 MSK

Nmap scan report for server.wifi.school34 (192.168.0.44)

Host is up (0.00027s latency).

All 1000 scanned ports on server.wifi.school34 (192.168.0.44) are open|filtered

MAC Address: B8:70:F4:29:E5:83 (Compal Information (kunshan))

Nmap done: 1 IP address (1 host up) scanned in 21.52 seconds

**TCP NULL сканирование**

**user@host:/# nmap -sN 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:01 MSK

Nmap scan report for server.wifi.school34 (192.168.0.44)

Host is up (0.00029s latency).

All 1000 scanned ports on server.wifi.school34 (192.168.0.44) are open|filtered

MAC Address: B8:70:F4:29:E5:83 (Compal Information (kunshan))

Nmap done: 1 IP address (1 host up) scanned in 21.46 seconds

**TCP FIN сканирование**

**user@host:/# nmap -sF 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:02 MSK

Nmap scan report for server.wifi.school34 (192.168.0.44)

Host is up (0.00030s latency).

All 1000 scanned ports on server.wifi.school34 (192.168.0.44) are open|filtered

MAC Address: B8:70:F4:29:E5:83 (Compal Information (kunshan))

Nmap done: 1 IP address (1 host up) scanned in 21.47 seconds

#### **TCP Xmas сканирование**

**user@host:/# nmap -sX 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:02 MSK

Nmap scan report for server.wifi.school34 (192.168.0.44)

Host is up (0.00026s latency).

All 1000 scanned ports on server.wifi.school34 (192.168.0.44) are open|filtered

MAC Address: B8:70:F4:29:E5:83 (Compal Information (kunshan))

Nmap done: 1 IP address (1 host up) scanned in 21.56 seconds

#### **TCP ACK сканирование**

**user@host:/# nmap -sA 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:03 MSK

Nmap scan report for server.wifi.school34 (192.168.0.44)

Host is up (0.00029s latency).

Not shown: 999 filtered ports

| PORT | STATE | SERVICE |
|------|-------|---------|
|------|-------|---------|

|        |            |     |
|--------|------------|-----|
| 22/tcp | unfiltered | ssh |
|--------|------------|-----|

MAC Address: B8:70:F4:29:E5:83 (Compal Information (kunshan))

Nmap done: 1 IP address (1 host up) scanned in 10.78 seconds

#### **TCP Window сканирование**

**user@host:/# nmap -sW 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:03 MSK

Nmap scan report for server.wifi.school34 (192.168.0.44)

Host is up (0.00027s latency).

Not shown: 999 filtered ports

| PORT | STATE | SERVICE |
|------|-------|---------|
|------|-------|---------|

|        |        |     |
|--------|--------|-----|
| 22/tcp | closed | ssh |
|--------|--------|-----|

MAC Address: B8:70:F4:29:E5:83 (Compal Information (kunshan))

Nmap done: 1 IP address (1 host up) scanned in 4.50 seconds

#### **TCP сканирование Мэймона**

**user@host:/# nmap -sM 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:03 MSK

Nmap scan report for server.wifi.school34 (192.168.0.44)

Host is up (0.00026s latency).

All 1000 scanned ports on server.wifi.school34 (192.168.0.44) are open|filtered

MAC Address: B8:70:F4:29:E5:83 (Compal Information (kunshan))

Nmap done: 1 IP address (1 host up) scanned in 21.47 seconds

#### **Сканирование протокола IP**

**user@host:/# nmap -sO 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:03 MSK

Nmap scan report for server.wifi.school34 (192.168.0.44)

Host is up (0.00026s latency).

Not shown: 255 open|filtered protocols

| PROTOCOL | STATE | SERVICE |
|----------|-------|---------|
|----------|-------|---------|

|   |      |      |
|---|------|------|
| 1 | open | icmp |
|---|------|------|

MAC Address: B8:70:F4:29:E5:83 (Compal Information (kunshan))

Nmap done: 1 IP address (1 host up) scanned in 3.48 seconds

#### **Определение версий ОС и служб в сочетании с TCP SYN сканирование**

**user@host:/# nmap -O -sV -sS 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:03 MSK

Nmap scan report for server.wifi.school34 (192.168.0.44)

Host is up (0.00031s latency).

Not shown: 999 filtered ports

| PORT | STATE | SERVICE | VERSION |
|------|-------|---------|---------|
|------|-------|---------|---------|

|        |      |     |  |
|--------|------|-----|--|
| 22/tcp | open | ssh | OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0) |
|--------|------|-----|--|

MAC Address: B8:70:F4:29:E5:83 (Compal Information (kunshan))

Warning: OSScan results may be unreliable because we could not find at least 1 open



and 1 closed port  
Device type: general purpose  
Running: Linux 2.6.X  
OS CPE: cpe:/o:linux:linux\_kernel:2.6  
OS details: Linux 2.6.18 - 2.6.22  
Network Distance: 1 hop  
Service Info: OS: Linux; CPE: cpe:/o:linux:linux\_kernel

OS and Service detection performed. Please report any incorrect results at  
<https://nmap.org/submit/> .

Nmap done: 1 IP address (1 host up) scanned in 13.10 seconds

Протокол сканирования с компьютера из ядра сети (а именно с сервера виртуализации):

#### Пинг-сканирование

**user@host:/# nmap -sP 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:28 MSK

Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn

Nmap done: 1 IP address (0 hosts up) scanned in 0.48 seconds

#### TCP SYN пинг

**user@host:/# nmap -PS 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:28 MSK

Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn

Nmap done: 1 IP address (0 hosts up) scanned in 0.52 seconds

#### TCP ACK пинг

**user@host:/# nmap -PA 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:28 MSK

Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn

Nmap done: 1 IP address (0 hosts up) scanned in 0.51 seconds

#### UDP пинг

**user@host:/# nmap -PU 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:28 MSK

Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn

Nmap done: 1 IP address (0 hosts up) scanned in 0.51 seconds

#### Различные типы пинг-пакетов ICMP

**user@host:/# nmap -PE 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:28 MSK

Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn

Nmap done: 1 IP address (0 hosts up) scanned in 0.51 seconds

**user@host:/# nmap -PP 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:28 MSK

Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn

Nmap done: 1 IP address (0 hosts up) scanned in 0.51 seconds

**user@host:/# nmap -PM 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:28 MSK

Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn

Nmap done: 1 IP address (0 hosts up) scanned in 0.52 seconds

#### Пинг с использованием протокола IP

**user@host:/# nmap -PO 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:28 MSK

Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn

Nmap done: 1 IP address (0 hosts up) scanned in 0.51 seconds

#### ARP пинг

**user@host:/# nmap -PR 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:28 MSK

Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn

Nmap done: 1 IP address (0 hosts up) scanned in 0.52 seconds

#### TCP connect сканирование

**user@host:/# nmap -sT 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:28 MSK

Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn

Nmap done: 1 IP address (0 hosts up) scanned in 0.50 seconds

#### UDP сканирование

**user@host:/# nmap -sU 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:28 MSK

Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn

Nmap done: 1 IP address (0 hosts up) scanned in 0.50 seconds

#### TCP NULL сканирование

**user@host:/# nmap -sN 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:28 MSK

Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn

Nmap done: 1 IP address (0 hosts up) scanned in 0.51 seconds

#### TCP FIN сканирование

**user@host:/# nmap -sF 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:28 MSK

Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn

Nmap done: 1 IP address (0 hosts up) scanned in 0.51 seconds

#### TCP Xmas сканирование

**user@host:/# nmap -sX 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:28 MSK

Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn

Nmap done: 1 IP address (0 hosts up) scanned in 0.51 seconds

#### TCP ACK сканирование

**user@host:/# nmap -sA 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:28 MSK

Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn

Nmap done: 1 IP address (0 hosts up) scanned in 0.50 seconds

#### TCP Window сканирование

**user@host:/# nmap -sW 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:28 MSK

Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn

Nmap done: 1 IP address (0 hosts up) scanned in 0.51 seconds

#### TCP сканирование Мэймона

**user@host:/# nmap -sM 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:28 MSK

Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn

Nmap done: 1 IP address (0 hosts up) scanned in 0.50 seconds

#### Сканирование протокола IP

**user@host:/# nmap -sO 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:28 MSK

Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn

Nmap done: 1 IP address (0 hosts up) scanned in 0.48 seconds

#### Определение версий ОС и служб в сочетании с TCP SYN сканирование

**user@host:/# nmap -O -sV -sS 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 12:28 MSK

Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn

Nmap done: 1 IP address (0 hosts up) scanned in 0.99 seconds

Дополнительное сканирование, рекомендованное выводом nmap, дало тот же результат:

**user@host:/# nmap -Pn 192.168.0.44**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-25 13:41 MSK

Nmap done: 1 IP address (0 hosts up) scanned in 0.52 seconds

Протокол сканирования с компьютера-клиента:

#### Пинг-сканирование

**user@host:/# nmap -sP 192.168.4.1**

Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-28 13:58 MSK

Nmap scan report for 192.168.4.1

Host is up (0.0057s latency).

MAC Address: D0:DF:9A:6A:89:18 (Liteon Technology)

Nmap done: 1 IP address (1 host up) scanned in 0.28 seconds

TCP SYN пинг

**user@host:/# nmap -PS 192.168.4.1**  
Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-28 13:58 MSK  
Nmap scan report for 192.168.4.1  
Host is up (0.0013s latency).  
Not shown: 999 filtered ports  
PORT STATE SERVICE  
53/tcp open domain  
MAC Address: D0:DF:9A:6A:89:18 (Liteon Technology)

Nmap done: 1 IP address (1 host up) scanned in 10.34 seconds

#### **TCP ACK пинг**

**user@host:/# nmap -PA 192.168.4.1**  
Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-28 13:58 MSK  
Nmap scan report for 192.168.4.1  
Host is up (0.0013s latency).  
Not shown: 999 filtered ports  
PORT STATE SERVICE  
53/tcp open domain  
MAC Address: D0:DF:9A:6A:89:18 (Liteon Technology)

Nmap done: 1 IP address (1 host up) scanned in 7.85 seconds

#### **UDP пинг**

**user@host:/# nmap -PU 192.168.4.1**  
Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-28 13:58 MSK  
Nmap scan report for 192.168.4.1  
Host is up (0.0013s latency).  
Not shown: 999 filtered ports  
PORT STATE SERVICE  
53/tcp open domain  
MAC Address: D0:DF:9A:6A:89:18 (Liteon Technology)

Nmap done: 1 IP address (1 host up) scanned in 13.33 seconds

#### **Различные типы пинг-пакетов ICMP**

**user@host:/# nmap -PE 192.168.4.1**  
Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-28 13:58 MSK  
Nmap scan report for 192.168.4.1  
Host is up (0.0012s latency).  
Not shown: 999 filtered ports  
PORT STATE SERVICE  
53/tcp open domain  
MAC Address: D0:DF:9A:6A:89:18 (Liteon Technology)

Nmap done: 1 IP address (1 host up) scanned in 6.69 seconds

**user@host:/# nmap -PP 192.168.4.1**  
Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-28 13:58 MSK  
Nmap scan report for 192.168.4.1  
Host is up (0.0019s latency).  
Not shown: 999 filtered ports  
PORT STATE SERVICE  
53/tcp open domain  
MAC Address: D0:DF:9A:6A:89:18 (Liteon Technology)

Nmap done: 1 IP address (1 host up) scanned in 12.20 seconds

**user@host:/# nmap -PM 192.168.4.1**  
Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-28 13:59 MSK  
Nmap scan report for 192.168.4.1  
Host is up (0.0013s latency).  
Not shown: 999 filtered ports  
PORT STATE SERVICE  
53/tcp open domain  
MAC Address: D0:DF:9A:6A:89:18 (Liteon Technology)

Nmap done: 1 IP address (1 host up) scanned in 7.90 seconds  
**Пинг с использование протокола IP**  
**user@host:/# nmap -PO 192.168.4.1**  
Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-28 13:59 MSK  
Nmap scan report for 192.168.4.1  
Host is up (0.0013s latency).  
Not shown: 999 filtered ports  
PORT STATE SERVICE  
53/tcp open domain  
MAC Address: D0:DF:9A:6A:89:18 (Liteon Technology)

Nmap done: 1 IP address (1 host up) scanned in 12.39 seconds  
**ARP пинг**  
**user@host:/# nmap -PR 192.168.4.1**  
Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-28 13:59 MSK  
Nmap scan report for 192.168.4.1  
Host is up (0.0013s latency).  
Not shown: 999 filtered ports  
PORT STATE SERVICE  
53/tcp open domain  
MAC Address: D0:DF:9A:6A:89:18 (Liteon Technology)

Nmap done: 1 IP address (1 host up) scanned in 12.81 seconds  
**TCP connect сканирование**  
**user@host:/# nmap -sT 192.168.4.1**  
Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-28 13:59 MSK  
Nmap scan report for 192.168.4.1  
Host is up (0.0024s latency).  
Not shown: 999 filtered ports  
PORT STATE SERVICE  
53/tcp open domain  
MAC Address: D0:DF:9A:6A:89:18 (Liteon Technology)

Nmap done: 1 IP address (1 host up) scanned in 4.83 seconds  
**UDP сканирование**  
**user@host:/# nmap -sU 192.168.4.1**  
Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-28 13:59 MSK  
Nmap scan report for 192.168.4.1  
Host is up (0.0011s latency).  
Not shown: 999 open|filtered ports  
PORT STATE SERVICE  
123/udp open ntp  
MAC Address: D0:DF:9A:6A:89:18 (Liteon Technology)

Nmap done: 1 IP address (1 host up) scanned in 9.14 seconds  
**TCP NULL сканирование**  
**user@host:/# nmap -sN 192.168.4.1**  
Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-28 13:59 MSK  
Nmap scan report for 192.168.4.1  
Host is up (0.0011s latency).  
All 1000 scanned ports on 192.168.4.1 are open|filtered  
MAC Address: D0:DF:9A:6A:89:18 (Liteon Technology)

Nmap done: 1 IP address (1 host up) scanned in 21.45 seconds  
**TCP FIN сканирование**  
**user@host:/# nmap -sF 192.168.4.1**  
Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-28 14:00 MSK  
Nmap scan report for 192.168.4.1  
Host is up (0.0011s latency).  
All 1000 scanned ports on 192.168.4.1 are open|filtered  
MAC Address: D0:DF:9A:6A:89:18 (Liteon Technology)

Nmap done: 1 IP address (1 host up) scanned in 21.47 seconds  
**TCP Xmas сканирование**  
user@host:/# nmap -sX 192.168.4.1  
Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-28 14:00 MSK  
Nmap scan report for 192.168.4.1  
Host is up (0.0011s latency).  
All 1000 scanned ports on 192.168.4.1 are open|filtered  
MAC Address: D0:DF:9A:6A:89:18 (Liteon Technology)

Nmap done: 1 IP address (1 host up) scanned in 21.46 seconds  
**TCP ACK сканирование**  
user@host:/# nmap -sA 192.168.4.1  
Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-28 14:01 MSK  
Nmap scan report for 192.168.4.1  
Host is up (0.0021s latency).  
Not shown: 999 filtered ports  
PORT STATE SERVICE  
53/tcp unfiltered domain  
MAC Address: D0:DF:9A:6A:89:18 (Liteon Technology)

Nmap done: 1 IP address (1 host up) scanned in 12.22 seconds  
**TCP Window сканирование**  
user@host:/# nmap -sW 192.168.4.1  
Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-28 14:01 MSK  
Nmap scan report for 192.168.4.1  
Host is up (0.0016s latency).  
Not shown: 999 filtered ports  
PORT STATE SERVICE  
53/tcp closed domain  
MAC Address: D0:DF:9A:6A:89:18 (Liteon Technology)

Nmap done: 1 IP address (1 host up) scanned in 18.08 seconds  
**TCP сканирование Мэймона**  
user@host:/# nmap -sM 192.168.4.1  
Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-28 14:01 MSK  
Nmap scan report for 192.168.4.1  
Host is up (0.0013s latency).  
Not shown: 999 open|filtered ports  
PORT STATE SERVICE  
53/tcp closed domain  
MAC Address: D0:DF:9A:6A:89:18 (Liteon Technology)

Nmap done: 1 IP address (1 host up) scanned in 10.78 seconds  
**Сканирование протокола IP**  
user@host:/# nmap -sO 192.168.4.1  
Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-28 14:01 MSK  
Nmap scan report for 192.168.4.1  
Host is up (0.0015s latency).  
Not shown: 255 open|filtered protocols  
PROTOCOL STATE SERVICE  
1 open icmp  
MAC Address: D0:DF:9A:6A:89:18 (Liteon Technology)

Nmap done: 1 IP address (1 host up) scanned in 4.58 seconds  
**Определение версий ОС и служб в сочетании с TCP SYN сканирование**  
user@host:/# nmap -O -sV -sS 192.168.4.1  
Starting Nmap 7.70 ( <https://nmap.org> ) at 2020-05-28 14:01 MSK  
Nmap scan report for 192.168.4.1  
Host is up (0.0015s latency).  
Not shown: 999 filtered ports  
PORT STATE SERVICE VERSION  
53/tcp open domain dnsmasq 2.80

```
MAC Address: D0:DF:9A:6A:89:18 (Liteon Technology)
Warning: OSScan results may be unreliable because we could not find at least 1 open
and 1 closed port
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.10 - 4.11, Linux 3.2 - 4.9
Network Distance: 1 hop
```

OS and Service detection performed. Please report any incorrect results at  
<https://nmap.org/submit/> .  
Nmap done: 1 IP address (1 host up) scanned in 20.59 seconds

Анализируя протоколы проверки сетевым сканером, можно сделать следующие выводы:

При сканировании системы с компьютера системного администратора доступным оказался только порт службы ssh, что и следовало ожидать. Кроме того, оказались возможны проверка доступности системы при помощи протокола ICMP и получение его аппаратного адреса, что также не противоречит постановке задачи и ожидаемым результатам. При определении версий программного обеспечения служба ssh не только позволила определить собственную версию, но и сообщила версию операционной системы. Однако считать это недостатком не следует, так как при проверке системы с постороннего компьютера из ядра сети получить этих сведений не удалось.

При сканировании системы клиентом беспроводной сети удалось обнаружить только порты 123/udp (служба точного времени) и 53/tcp (служба доменных имён). При определении версий программного обеспечения удалось выяснить, что услуги dns-сервера предоставляет dnsmasq версии 2.80, работающий под управлением ОС Linux с версией ядра 3.X-4.X. Такие результаты также следует считать полностью соответствующими поставленным требованиям.

Кроме защиты самого сервера беспроводной сети, межсетевой экран должен повышать безопасность и клиентов этой сети. Так попытка найти маршрут с сервера виртуализации к одному из клиентских устройств, имеющему адрес 192.168.4.29, завершилась неудачей, как и попытка обнаружения этого клиента утилитой ping:

```
root@virtserver:/# traceroute 192.168.4.29
traceroute to 192.168.4.29 (192.168.4.29), 30 hops max, 60 byte packets
 1  192.168.0.1 (192.168.0.1)  0.547 ms  0.501 ms  0.468 ms
 2  server.wifi.school34 (192.168.0.44)  0.415 ms  0.384 ms  0.359 ms
 3  * * *
...
30 * * *
root@virtserver:/# ping 192.168.4.29
PING 192.168.4.29 (192.168.4.29) 56(84) bytes of data.
From 192.168.0.1: icmp_seq=2 Redirect Host(New nexthop: 192.168.0.44)
...
^C
--- 192.168.4.29 ping statistics ---
21 packets transmitted, 0 received, 100% packet loss, time 482ms
```

*Замечание:* здесь имя сервера виртуализации (server, server.service.school34) в приглашении командной строки изменено во избежание путаницы с именем сервера беспроводной сети (server, server.wifi.school34) на virtserver.

При этом клиент имеет свободный доступ к виртуальным серверам, расположенным на этом сервере виртуализации.

Одновременно с тем, и клиент не имеет доступа к устройствам в других подсетях, а его связь с другими устройствами в беспроводной сети невозможна, если сервер беспроводной сети отключён от неё (например, при помощи средств управления питанием сетевой карты).

## Завершение установки

В завершение установки имеет смысл обновить систему, очистить локальный кэш пакетов и удалить пакеты, которые системе стали не нужны. Это можно сделать командами:

```
root@server:/# apt-get upgrade
root@server:/# apt-get autoremove
root@server:/# apt-get clean
```

## Использование сервера

Описанные далее сценарии, команды и методики предназначены для использования в процессе функционирования системы для контроля её состояния или смены режимов функционирования. Дальнейшее описание можно воспринимать как краткое руководство (howto).

## Порядок входа в систему

В силу описанных выше настроек ssh и сетевого фильтра предусмотрен следующий порядок удалённого входа в систему: выполнить вход с компьютера системного администратора от имени guest, а затем при необходимости выполнить повышение привилегий до root локально. Эти действия продемонстрированы в следующем фрагменте терминального сеанса:

```
administrator@admin:~$ ssh guest@server.wifi.school34
guest@server.wifi.school34's password:
Last login: Tue Mar  3 14:26:42 2020 from 192.168.0.2
```

```
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
```

```
guest@server:~$ su
Password:
root@server:/home/guest#
```

Для возможности использования программ с графическим пользовательским интерфейсом в команде создания ssh-соединения следует использовать ключ -X:

```
administrator@admin:~$ ssh -X guest@server.wifi.school34
```

## Мониторинг состояния аппаратных средств

Для получения сведений от аппаратных датчиков достаточно выполнить от имени любого пользователя команду sensors, например:

```

guest@server:/# sensors
coretemp-isa-0000
Adapter: ISA adapter
Core 0:      +40.0°C (high = +80.0°C, crit = +90.0°C)
Core 2:      +43.0°C (high = +80.0°C, crit = +90.0°C)
acpitz-acpi-0
Adapter: ACPI interface
temp1:      +42.0°C (crit = +127.0°C)
nouveau-pci-0100
Adapter: PCI adapter
GPU core:    +0.85 V (min = +0.80 V, max = +1.00 V)
temp1:      +50.0°C (high = +95.0°C, hyst = +3.0°C)
              (crit = +105.0°C, hyst = +5.0°C)
              (emerg = +135.0°C, hyst = +5.0°C)

```

Для получения информации от встроенной аппаратуры самодиагностики жёсткого диска можно воспользоваться следующими выполняемыми с правами суперпользователя командами:

Поиск всех S.M.A.R.T-совместимых устройств:

```

root@server:/# smartctl --scan
/dev/sda -d scsi # /dev/sda, SCSI device

```

Получение информации об устройстве:

```

root@server:/# smartctl /dev/sda -i
smartctl 6.6 2017-11-05 r4594 [x86_64-linux-4.19.0-5-amd64] (local build)
Copyright (C) 2002-17, Bruce Allen, Christian Franke, www.smartmontools.org

```

```

=== START OF INFORMATION SECTION ===
Model Family:      Seagate Momentus 5400.6
Device Model:      ST9320325AS
Serial Number:     6VDCABKR
LU WWN Device Id:  5 000c50 03dc5905c
Firmware Version:  0011LVM1
User Capacity:     320 072 933 376 bytes [320 GB]
Sector Size:       512 bytes logical/physical
Rotation Rate:     5400 rpm
Device is:         In smartctl database [for details use: -P show]
ATA Version is:    ATA8-ACS T13/1699-D revision 4
SATA Version is:   SATA 2.6, 3.0 Gb/s (current: 3.0 Gb/s)
Local Time is:     Thu Mar 19 11:31:16 2020 MSK
SMART support is:  Available - device has SMART capability.
SMART support is:  Enabled

```

Получение подробных сведений об устройстве, относящихся к S.M.A.R.T (объёмный вывод команды опущен для краткости):

```

root@server:/# smartctl /dev/sda -a

```

Получение ещё более подробных сведений об устройстве, относящихся к S.M.A.R.T и не только (ещё более объёмный вывод команды также опущен для краткости):

```

root@server:/# smartctl /dev/sda -x

```

Кроме того, возможно использование графического пользовательского интерфейса к этим и другим командам – GSmartControl. При этом запуск этой утилиты также потребует знания пароля суперпользователя.



## Смена внешнего сервера доменных имен

В процессе функционирования сервера беспроводной сети часто может возникать необходимость смены внешнего DNS-сервера. В основном режиме использования он пользуется услугами YandexDNS с минимальной фильтрацией ресурсов по их доменным именам, однако нередко может возникать необходимость усиливать режим фильтрации. Для этого надо переключиться на более «строгий» SkyDNS. Процесс переключения состоит из трёх шагов:

- изменение адреса внешнего сервера в основном конфигурационном файле `dnsmasq`
- перезапуск службы `dnsmasq`
- проверка успешности перезапуска службы

Для упрощения этого процесса в системе присутствует сценарий `/usr/local/sbin/dnsmode` :

```
#!/bin/bash

if [ $1 = "free" ]
then
    sed s/"193\.58\.251\.251"/"77\.88\.8\.7"/ -i /etc/dnsmasq.conf
    systemctl restart dnsmasq
elif [ $1 = "safe" ]
then
    sed s/"77\.88\.8\.7"/"193\.58\.251\.251"/ -i /etc/dnsmasq.conf
    systemctl restart dnsmasq
fi
systemctl status --no-pager dnsmasq
exit 0
```

Этот сценарий следует использовать с единственным параметром командной строки, принимающим значения «free» или «safe». При указании любых других значений (в том числе и пустого) сценарий только выводит текущее состояние службы. В случае значения «free» происходит переключение на YandexDNS, при значении «safe» – на SkyDNS.

Следующий смешанный (в смысле выполнения команд как на сервере, так и на клиенте) фрагмент терминального сеанса демонстрирует поведение сценария:

```
root@server:/# dnsmode free
```

```
• dnsmasq.service - dnsmasq - A lightweight DHCP and caching DNS server
Loaded: loaded (/lib/systemd/system/dnsmasq.service; enabled; vendor preset:
enabled)
```

```
Drop-In: /etc/systemd/system/dnsmasq.service.d
└─override.conf
```

```
Active: active (running) since Tue 2020-03-03 15:06:04 MSK; 9ms ago
```

```
...
```

```
map 03 15:06:04 server dnsmasq[3752]: using nameserver 77.88.8.7#53(via enp7s0)
```

```
xubuntu@xubuntu:~$ host vk.com
```

```
vk.com has address 87.240.139.194
```

```
vk.com has address 87.240.190.67
```

```
vk.com has address 87.240.190.72
```

```
vk.com has address 93.186.225.208
```

```
vk.com has address 87.240.137.158
```

```
vk.com has address 87.240.190.78
```

```
vk.com mail is handled by 0 mx.vk.com.
```

```
vk.com mail is handled by 20 mx2.vk.com.
```

```
root@server:/# dnsmode safe
```

```
• dnsmasq.service - dnsmasq - A lightweight DHCP and caching DNS server
  Loaded: loaded (/lib/systemd/system/dnsmasq.service; enabled; vendor preset:
  enabled)
```

```
  Drop-In: /etc/systemd/system/dnsmasq.service.d
    └─override.conf
```

```
  Active: active (running) since Tue 2020-03-03 15:06:51 MSK; 8ms ago
```

```
...
mar 03 15:06:51 server dnsmasq[3782]: using nameserver 193.58.251.251#53(via enp7s0)
```

```
xubuntu@xubuntu:~$ host vk.com
```

```
vk.com has address 193.58.251.1
```

Так вначале служба DNS переводится в свободный режим, затем на клиенте выполняется dns-запрос о домене социальной сети, на который приходит полный ответ. Затем сервер переводится в фильтрующий режим и на клиенте выполняется тот же запрос. Теперь клиент вместо реальных адресов получает адрес страницы блокировки SkyDNS.

При этом следует отметить, во-первых, то, что в выводе сценария явно указан используемый внешний сервер доменных имён.

Во-вторых, следует уточнить, что уже установленные клиентом соединения при таком переключении не разрываются, блокируется лишь возможность новых соединений с неблагонадёжными ресурсами.

## Получение текущего списка клиентских устройств

При подключении клиента и выдаче ему сетевого адреса информация об этом клиенте заносится в файл `/var/lib/misc/dnsmasq.leases`. Формат этого файла таков: каждому клиенту ставится в соответствие одна строка, в которой последовательно перечислены время до окончания срока аренды адреса, аппаратный адрес клиента, выданный ему IPv4-адрес (рассматриваемая система не работает с IPv6), имя клиентского хоста, собственное или выданное в аренду вместе с адресом, и идентификатор dhcp-клиента. В некоторый момент работы системы этот файл имел вид:

```
root@server:/# cat /var/lib/misc/dnsmasq.leases
```

```
1583282460 cc:61:e5:fa:e1:cd 192.168.4.29 android-8a69d96bb881b05c *
```

```
1583279954 d0:df:9a:6a:4c:b4 192.168.4.16 debian *
```

Это соответствует android-смартфону и ноутбуку с именем хоста debian.

Просмотр этого файла, соответственно, позволяет выявить клиентов, которые получили адрес, но ещё не отказались от него. При этом не гарантируется, что клиент, указанный в этом файле, в данный момент действительно находится в сети. Клиент может оказаться вне зоны покрытия беспроводной сети и, тем самым, потерять связь с ней. Однако в этом случае не произошло корректного закрытия подключения к сети, и клиент может вернуться в неё без повторного прохождения процедуры получения адреса, разумеется, если за время его отсутствия не истекло время аренды адреса. С другой стороны, клиентское устройство может безвозвратно покинуть сеть без корректного закрытия соединения, а сведения о нём по-прежнему будут содержаться в этом файле. Таким образом этот файл не является гарантированно достоверным источником информации о находящихся в рассматриваемый момент в сети клиентах. Дополнительно к нему можно воспользоваться командой

```
root@server:/# ip neighbour show dev wlp6s0
```

В выводе этой команды отображаются аппаратные адреса сетевых устройств, известных серверу (arp-кэш). Сведения о таких устройствах обновляются гораздо чаще, чем данные об арендуемых адресах, соответственно эти сведения могут помочь выявить клиентов, которые действительно находятся в сети или покинули её относительно недавно.

Ещё одним средством проверки присутствия клиента в сети является команда `ping`, однако и она не гарантирована от того, что клиентское устройство, например защищённое файерволом, откликнется на её запрос.

Ранее при описании настройки службы `dnsmasq` был также приведён способ с отправкой сигнала `SIGUSR1` процессу `dnsmasq`.

Наконец, использование сканера сети (например, `nmap`) позволит получить ответ с ещё большей степенью достоверности, однако, опять же, без гарантий, т. к. клиентское устройство также способно этому противодействовать.

Тем не менее, отбрасывая слишком маловероятные с учётом условий функционирования сети ситуации, можно отметить, что использование первых трёх средств в подавляющем большинстве случаев оказывается достаточно, чтобы выявить текущих активных клиентов беспроводной сети.

## **Замечания по подключению клиентов**

Наконец, необходимо сделать замечание о подключении клиентских устройств: только само клиентское устройство определяет тот набор служб, которым оно готово пользоваться. Сервер лишь может эти службы предлагать. Так, например, `dhcpcd`-клиент `windows`-устройств не использует полученное от сервера имя хоста. Другие устройства вполне могут игнорировать и доменное имя, и предлагаемые серверы точного времени и доменных имён. В большинстве случаев это не таит в себе каких-либо сложностей для самой сети. Исключением являются лишь попытки обхода фильтрации по доменным именам или попытки использования сторонних прокси-серверов, чему, однако, противостоит межсетевой экран.

Также в функционале сервера не заложена возможность предоставлять клиентам информацию об услугах других серверов локальной сети. Таким образом ярлыки на рабочих столах, закладки в интернет-обозревателях и т. п. остаются на усмотрение самих клиентов.