

## МОДУЛЬ №3

### 1. Выполните миграцию на новый контроллер домена BR-SRV с HQ-SRV, являющийся наследием

К этому заданию мы вернёмся позже (возможно), спасибо разработчикам за их “прекрасные” формулировки.

### 2. Выполните настройку центра сертификации на базе HQ-SRV

Аналогично! Ждём книгу Уймина по 3 модулю. (Спойлер: не дождёмся)

### 3. Перенастройте ip-туннель с базового до уровня туннеля, обеспечивающего шифрование трафика

Прекрасное задание, приступаем к выполнению!

#### HQ-RTR

Для начала необходимо установить пакет на наш роутер HQ-RTR:

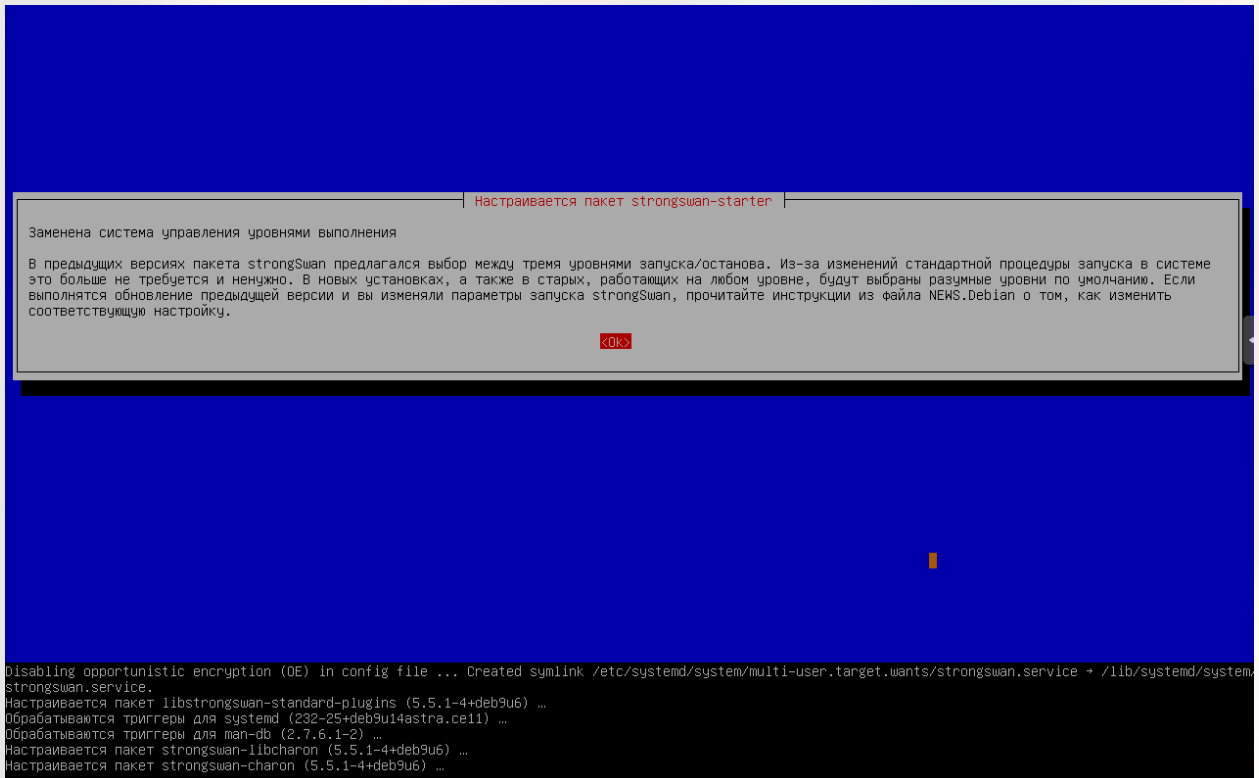
```
apt update
```

```
apt install strongswan libcharon-extra-plugins -y
```

После начала установки вылезет окно предупреждения о преднастройке этой службы, всё пройдёт автоматически. Жмём на:

**OK**

AUTHORS:  
NECHAEV  
NAUMOV  
NAGORNOVA



Теперь нам нужно создать файл, в котором будет прописано создание туннеля.

Пишем в консоль:

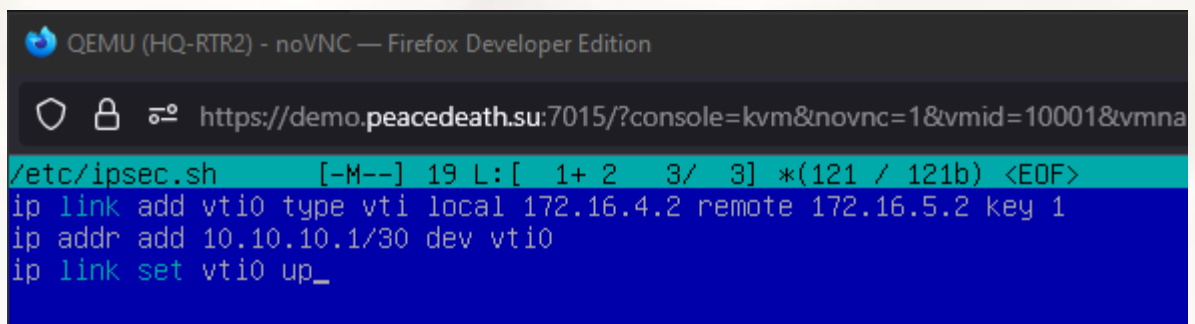
**mcedit /etc/ipsec.sh**

И заполняем его следующими строками:

**ip link add vti0 type vti local 172.16.4.2 remote 172.16.5.2 key 1**

**ip addr add 10.10.10.1/30 dev vti0**

**ip link set vti0 up**



Далее этот файл нужно сделать исполняемым:

**chmod +x /etc/ipsec.sh**

```
root@hq-rtr:~# chmod +x /etc/ipsec.sh
root@hq-rtr:~#
```

Теперь нужно отредактировать сам файл конфигурации **ipsec.conf**, в нём будут храниться основные параметры:

**mcedit /etc/ipsec.conf**

И вносим следующие строки:

**conn tunnel**

**leftupdown=/etc/ipsec.sh**

**left=172.16.4.2**

**leftsubnet=0.0.0.0/0**

**right=172.16.5.2**

**rightsubnet=0.0.0.0/0**

**authby=secret**

**keyexchange=ikev2**

**auto=start**

**mark=1**

**type=tunnel**

**esp=aes256-sha256-modp1024**

AUTHORS:  
NECHAEV  
NAUMOV  
NAGORNOVA

```
QEMU (HQ-RTR2) - noVNC — Firefox Developer Edition
https://demo.peacedeath.su:7015/?console=kvm&novnc=1&...
/etc/ipsec.conf [----] 0 L:[ 1+24 25/ 44] *(427 / 89)
# ipsec.conf - strongSwan IPsec configuration file

# basic configuration

config setup
<-----># strictcrpolicy=yes
<-----># uniqueids = no

# Add connections here.

# Sample VPN connections

conn tunnel
    leftupdown=/etc/ipsec.sh
    left=172.16.4.2
    leftsubnet=0.0.0.0/0
    right=172.16.5.2
    rightsubnet=0.0.0.0/0
    authby=secret
    keyexchange=ikev2
    auto=start
    mark=1
    type=tunnel
    esp=aes256-sha256-modp1024

#conn sample-self-signed
#    leftsubnet=10.1.0.0/16
#    leftcert=selfCert.der
#    leftsendcert=never
#    right=192.168.0.2
#    rightsubnet=10.2.0.0/16
#    rightcert=peerCert.der
#    auto=start

#conn sample-with-ca-cert
#    leftsubnet=10.1.0.0/16
#    leftcert=myCert.pem
#    right=192.168.0.2
#    rightsubnet=10.2.0.0/16
#    rightid="C=CH, O=Linux strongSwan CN=peer name"
#    auto=start

include /var/lib/strongswan/ipsec.conf.inc
```

AUTHORS:

NECHAEV

NAUMOV

NAGORNOV

Далее нужно настроить файл **ipsec.secrets**.

Вносим туда строку:

**172.16.4.2 172.16.5.2 : PSK "123qweR%"**

```
QEMU (HQ-RTR2) - noVNC — Firefox Developer Edition
https://demo.peacedeath.su:7015/?console=kvm&novnc=1&vmid=10001&vmname=HQ-RTR2&node=althome&resize=off
/etc/ipsec.secrets [-M--] 38 L:[ 1+ 7 8/ 12] *(305 / 354b) 0010 0x00A
# This file holds shared secrets or RSA private keys for authentication.
#
# RSA private key for this host, authenticating it to any other host
# which knows the public part.
#
# this file is managed with debconf and will contain the automatically created private key
172.16.4.2 172.16.5.2 : PSK "123qweR%"_
include /var/lib/strongswan/ipsec.secrets.inc
```

Ещё один конфиг **charon.conf**, открываем его.

И редактируем в нём следующую строку, приводя к виду:

**install\_routes = no**

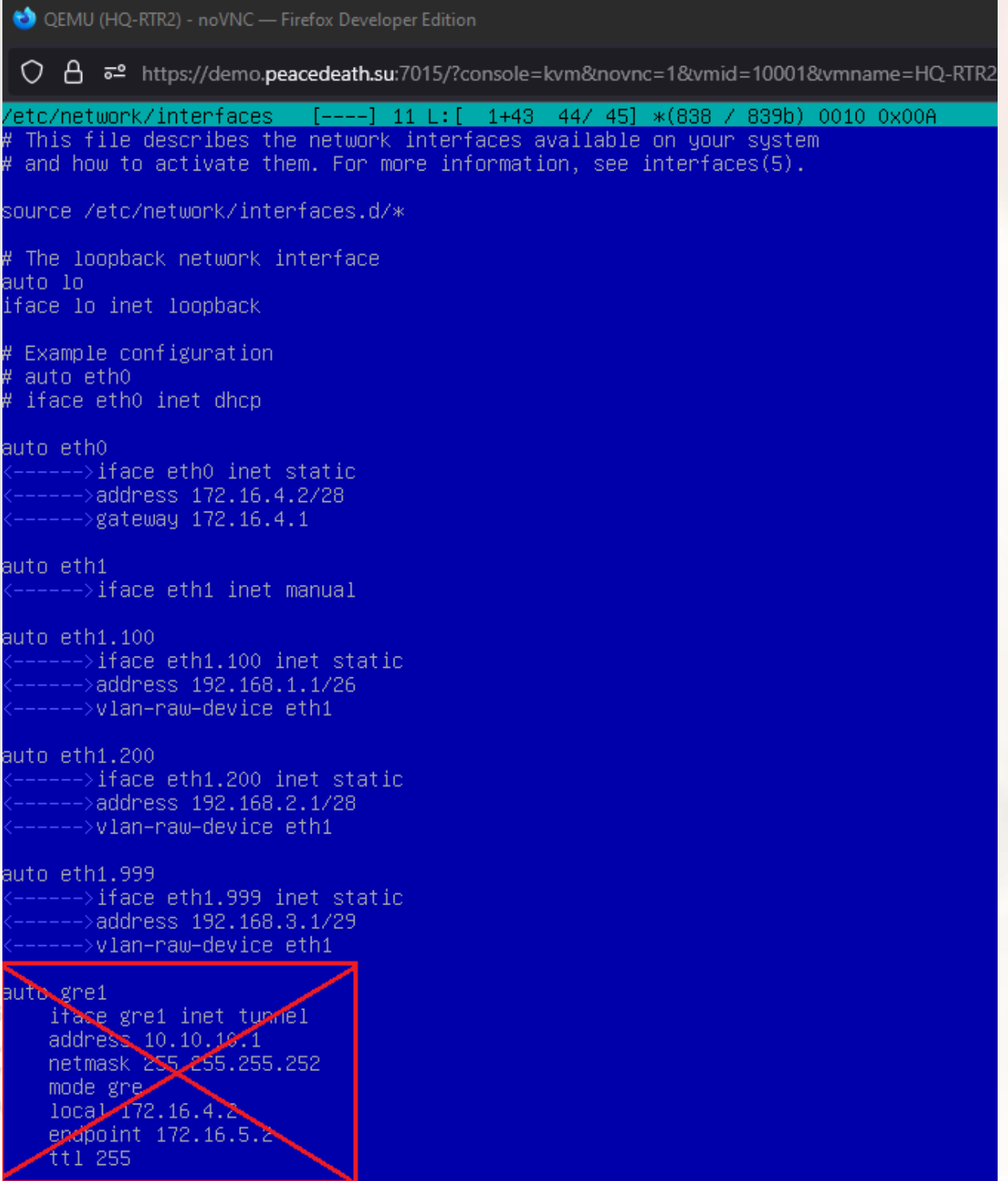
```
QEMU (HQ-RTR2) - noVNC — Firefox Developer Edition
https://demo.peacedeath.su:7015/?console=kvm&novnc=1&vmid=10001&vmname=HQ-RTR2&node=althome&resize=off
/etc/strongswan.d/charon.conf [-M--] 23 L:[ 76+35 111/343] *(3777/10008b) 0010 0x00A
# i_dont_care_about_security_and_use_aggressive_mode_psk = no
#
# Whether to ignore the traffic selectors from the kernel's acquire events
# for IKEv2 connections (they are not used for IKEv1).
# ignore_acquire_ts = no
#
# A space-separated list of routing tables to be excluded from route
# lookups.
# ignore_routing_tables =
#
# Maximum number of IKE_SAs that can be established at the same time before
# new connection attempts are blocked.
# ikesa_limit = 0
#
# Number of exclusively locked segments in the hash table.
# ikesa_table_segments = 1
#
# Size of the IKE_SA hash table.
# ikesa_table_size = 1
#
# Whether to close IKE_SA if the only CHILD_SA closed due to inactivity.
# inactivity_close_ike = no
#
# Limit new connections based on the current number of half open IKE_SAs,
# see IKE_SA_INIT DROPPING in strongswan.conf(5).
# init_limit_half_open = 0
#
# Limit new connections based on the number of queued jobs.
# init_limit_job_load = 0
#
# Causes charon daemon to ignore IKE initiation requests.
# initiator_only = no
#
# Install routes into a separate routing table for established IPsec
# tunnels.
install_routes = no
#
# Install virtual IP addresses.
# install_virtual_ip = yes
#
# The name of the interface on which virtual IP addresses should be
# installed
```

Важно удалить предыдущий туннель, который мы создавали в первом модуле, он больше нам не нужен.

Заходим в конфиг:

**mcedit /etc/network/interfaces**

И удаляем всё, что относится к **gre1**.



```
QEMU (HQ-RTR2) - noVNC — Firefox Developer Edition
https://demo.peacedeath.su:7015/?console=kvm&novnc=1&vmid=10001&vmname=HQ-RTR2
/etc/network/interfaces [----] 11 L:[ 1+43 44/ 45] *(838 / 839b) 0010 0x00A
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# Example configuration
# auto eth0
# iface eth0 inet dhcp

auto eth0
<----->iface eth0 inet static
<----->address 172.16.4.2/28
<----->gateway 172.16.4.1

auto eth1
<----->iface eth1 inet manual

auto eth1.100
<----->iface eth1.100 inet static
<----->address 192.168.1.1/26
<----->vlan-raw-device eth1

auto eth1.200
<----->iface eth1.200 inet static
<----->address 192.168.2.1/28
<----->vlan-raw-device eth1

auto eth1.999
<----->iface eth1.999 inet static
<----->address 192.168.3.1/29
<----->vlan-raw-device eth1

auto gre1
iface gre1 inet tunnel
address 10.10.10.1
netmask 255.255.255.252
mode gre
local 172.16.4.2
endpoint 172.16.5.2
ttl 255
```

Чтобы не перезагружать машину, удалим существующий туннель командой:

**ip tunnel del gre1**

И удалим его ещё и из **frr**, заходя в режим конфигурации:

**vysh**

**conf t**

**no interface gre1**

```
root@hq-rtr:~# ip tunnel del gre1
root@hq-rtr:~# vtysh

Hello, this is FRRouting (version 6.0.2).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

hq-rtr.au-team.irpo# conf t
hq-rtr.au-team.irpo(config)# no interface gre1
hq-rtr.au-team.irpo(config)#
```

И осталось только перезагрузить службу **ipsec**:

**ipsec restart**

```
root@hq-rtr:~# ipsec restart
Stopping strongSwan IPsec...
Starting strongSwan 5.5.1 IPsec [starter]...
root@hq-rtr:~# _
```

## **BR-RTR**

Таким же образом устанавливаем пакеты уже на роутер **BR-RTR**:

**apt update**

**apt install strongswan libcharon-extra-plugins -y**

```

QEMU (BR-RTR2) - noVNC — Firefox Developer Edition
https://demo.peacedeath.su:7015/?console=kvm&novnc=1&vmid=10002&vmname=BR-RTR2&node=althome&resize=off

root@br-rtr:~# apt-get update
Сущ:1 https://dl.astralinux.ru/astra/stable/2.12_x86-64/repository ore1 InRelease
Чтение списков пакетов... Готово
root@br-rtr:~# apt-get install strongswan libcharon-extra-plugins
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Следующий пакет устанавливался автоматически и больше не требуется:
  libopts25
Для его удаления используйте «apt autoremove».
Будут установлены следующие дополнительные пакеты:
  libstrongswan libstrongswan-standard-plugins strongswan-charon strongswan-libcharon strongswan-starter
Предлагаемые пакеты:
  libstrongswan-extra-plugins
НОВЫЕ пакеты, которые будут установлены:
  libcharon-extra-plugins libstrongswan libstrongswan-standard-plugins strongswan strongswan-charon strongswan-libcharon strongswan-starter
обновлено 0, установлено 7 новых пакетов, для удаления отмечено 0 пакетов, и 0 пакетов не обновлено.
Необходимо скачать 1 443 кБ архивов.
После данной операции, объём занятого дискового пространства возрастёт на 4 277 кБ.
Хотите продолжить? [Д/н] y
Пол:1 https://dl.astralinux.ru/astra/stable/2.12_x86-64/repository ore1/main amd64 libstrongswan amd64 5.5.1-4+deb9u6 [388 kB]
Пол:2 https://dl.astralinux.ru/astra/stable/2.12_x86-64/repository ore1/main amd64 strongswan-starter amd64 5.5.1-4+deb9u6 [233 kB]
Пол:3 https://dl.astralinux.ru/astra/stable/2.12_x86-64/repository ore1/main amd64 strongswan-libcharon amd64 5.5.1-4+deb9u6 [280 kB]
Пол:4 https://dl.astralinux.ru/astra/stable/2.12_x86-64/repository ore1/main amd64 strongswan-charon amd64 5.5.1-4+deb9u6 [87,4 kB]
Пол:5 https://dl.astralinux.ru/astra/stable/2.12_x86-64/repository ore1/main amd64 libcharon-extra-plugins amd64 5.5.1-4+deb9u6 [236 kB]
Пол:6 https://dl.astralinux.ru/astra/stable/2.12_x86-64/repository ore1/main amd64 libstrongswan-standard-plugins amd64 5.5.1-4+deb9u6 [125 kB]
Пол:7 https://dl.astralinux.ru/astra/stable/2.12_x86-64/repository ore1/main amd64 strongswan all 5.5.1-4+deb9u6 [93,0 kB]
Получено 1 443 кБ за 2с (709 кБ/с)
Предварительная настройка пакетов ...
Выбор ранее не выбранного пакета libstrongswan.
(Чтение базы данных ... на данный момент установлено 71449 файлов и каталогов.)
Подготовка к распаковке ../0-libstrongswan_5.5.1-4+deb9u6_amd64.deb ...
Распаковывается libstrongswan (5.5.1-4+deb9u6) ...
Выбор ранее не выбранного пакета strongswan-starter.
Подготовка к распаковке ../1-strongswan-starter_5.5.1-4+deb9u6_amd64.deb ...
Распаковывается strongswan-starter (5.5.1-4+deb9u6) ...

```

Таким же образом создаём файл **ipsec.sh**, в котором будет храниться наш туннель:

**mcedit /etc/ipsec.sh**

И вносим туда строки:

**ip link add vti0 type vti local 172.16.5.2 remote 172.16.4.2 key 1**

**ip addr add 10.10.10.2/30 dev vti0**

**ip link set vti0 up**

```

QEMU (BR-RTR2) - noVNC — Firefox Developer Edition
https://demo.peacedeath.su:7015/?console=kvm&novnc=1&vmid=10002&vmname=BR-RTR2&node=althome&resize=off

/etc/ipsec.sh [----] 2 L:[ 1+ 2 3/ 3] *(104 / 121b) 0032 0x020
ip link add vti0 type vti local 172.16.5.2 remote 172.16.4.2 key 1
ip addr add 10.10.10.2/30 dev vti0
ip link set vti0 up

```

И делаем его исполняемым:

AUTHORS:

NECHAEV

NAUMOV

NAGORNOVA

**chmod +x /etc/ipsec.sh**

```
root@br-rtr:~# chmod +x /etc/ipsec.sh
```

Теперь нужно отредактировать файл конфигурации **ipsec.conf**, в нём будут храниться основные параметры:



**mcedit /etc/ipsec.conf**

И вносим следующие строки:

**conn tunnel**

**leftupdown=/etc/ipsec.sh**

**left=172.16.5.2**

**leftsubnet=0.0.0.0/0**

**right=172.16.4.2**

**rightsubnet=0.0.0.0/0**

**authby=secret**

**keyexchange=ikev2**

**auto=start**

**mark=1**

**type=tunnel esp=aes256-sha256-modp1024**

AUTHORS:  
NECHAEV  
NAUMOV  
NAGORNOVA

```
QEMU (BR-RTR2) - noVNC — Firefox Developer Edition
https://demo.peacedeath.su:7015/?console=kvm&novnc=1&vmid=
/etc/ipsec.conf [-----] 0 L:[ 1+24 25/ 44] *(427 / 899b) 00
# ipsec.conf - strongSwan IPsec configuration file

# basic configuration

config setup
<-----># strictcr1policy=yes
<-----># uniqueids = no

# Add connections here.

# Sample VPN connections

conn tunnel
    leftupdown=/etc/ipsec.sh
    left=172.16.5.2
    leftsubnet=0.0.0.0/0
    right=172.16.4.2
    rightsubnet=0.0.0.0/0
    authby=secret
    keyexchange=ikev2
    auto=start
    mark=1
    type=tunnel
    esp=aes256-sha256-modp1024

#conn sample-self-signed
#    leftsubnet=10.1.0.0/16
#    leftcert=selfCert.der
#    leftsendcert=never
#    right=192.168.0.2
#    rightsubnet=10.2.0.0/16
#    rightcert=peerCert.der
#    auto=start

#conn sample-with-ca-cert
#    leftsubnet=10.1.0.0/16
#    leftcert=myCert.pem
#    right=192.168.0.2
#    rightsubnet=10.2.0.0/16
#    rightid="C=CH, O=Linux strongSwan CN=peer name"
#    auto=start

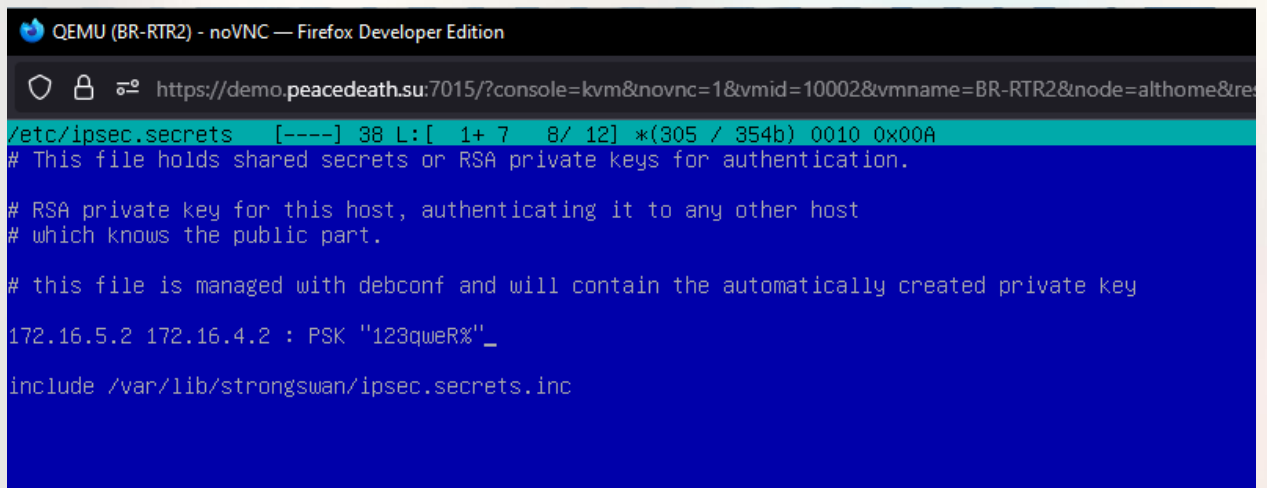
include /var/lib/strongswan/ipsec.conf.inc
```

Снова переходим к **ipsec.secrets**:

**mcedit /etc/ipsec.secrets**

Вносим туда строку:

**172.16.5.2 172.16.4.2 : PSK "123qweR%"**



```
QEMU (BR-RTR2) - noVNC — Firefox Developer Edition
https://demo.peacedeath.su:7015/?console=kvm&novnc=1&vmid=10002&vmname=BR-RTR2&node=althome&res...
/etc/ipsec.secrets  [----] 38 L:[ 1+ 7 8/ 12] *(305 / 354b) 0010 0x00A
# This file holds shared secrets or RSA private keys for authentication.
# RSA private key for this host, authenticating it to any other host
# which knows the public part.
# this file is managed with debconf and will contain the automatically created private key
172.16.5.2 172.16.4.2 : PSK "123qweR%"_
include /var/lib/strongswan/ipsec.secrets.inc
```

Открываем файл **charon.conf**:

**mcedit /etc/strongswan.d/charon.conf**

Приводим в нём строку к следующему виду:

**install\_routes = no**

AUTHORS:  
NECHAEV  
NAUMOV  
NAGORNOVA

```
QEMU (BR-RTR2) - noVNC — Firefox Developer Edition
https://demo.peacedeath.su:7015/?console=kvm&novnc=1&vmid=10002&vmname=BR-RTR2&node
/etc/strongswan.d/charon.conf [-M--] 23 L: [ 76+35 111/343] *(3777/10008b) 0010 0x00A
# i_dont_care_about_security_and_use_aggressive_mode_psk = no

# Whether to ignore the traffic selectors from the kernel's acquire events
# for IKEv2 connections (they are not used for IKEv1).
# ignore_acquire_ts = no

# A space-separated list of routing tables to be excluded from route
# lookups.
# ignore_routing_tables =

# Maximum number of IKE_SAs that can be established at the same time before
# new connection attempts are blocked.
# ikesa_limit = 0

# Number of exclusively locked segments in the hash table.
# ikesa_table_segments = 1

# Size of the IKE_SA hash table.
# ikesa_table_size = 1

# Whether to close IKE_SA if the only CHILD_SA closed due to inactivity.
# inactivity_close_ike = no

# Limit new connections based on the current number of half open IKE_SAs,
# see IKE_SA_INIT DROPPING in strongswan.conf(5).
# init_limit_half_open = 0

# Limit new connections based on the number of queued jobs.
# init_limit_job_load = 0

# Causes charon daemon to ignore IKE initiation requests.
# initiator_only = no

# Install routes into a separate routing table for established IPsec
# tunnels.
install_routes = no

# Install virtual IP addresses.
# install_virtual_ip = yes

# The name of the interface on which virtual IP addresses should be
# installed.
# install_virtual_ip_on =

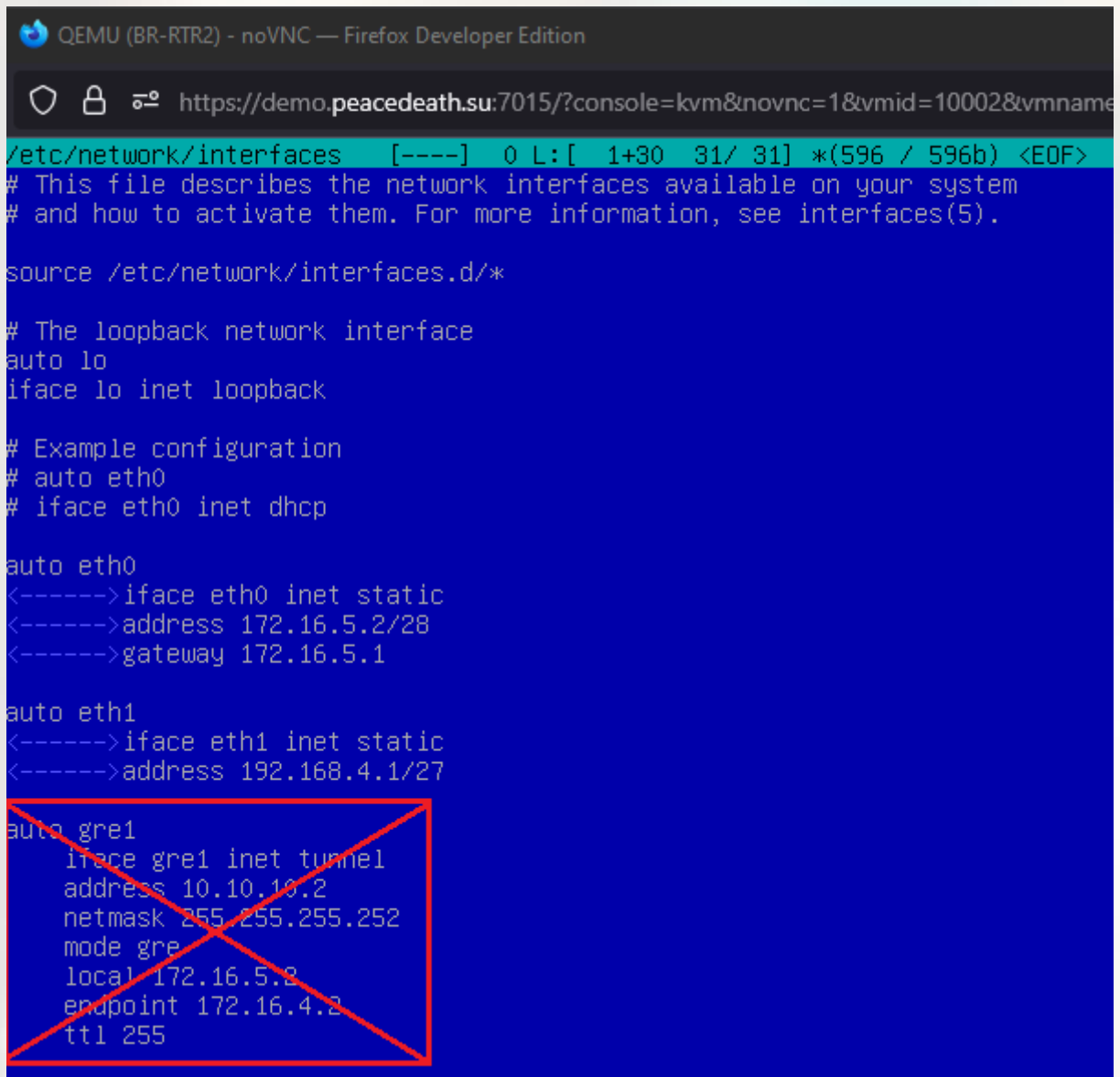
# Check daemon, libstrongswan and plugin integrity at startup.
# integrity_test = no

# A comma-separated list of network interfaces that should be ignored, if
```

Удалим теперь предыдущий туннель, который был создан в первом модуле, редактируем файл `interfaces` для этого:

**`mcedit /etc/network/interfaces`**

Удаляем там всё, что связано с **`gre1`**.



```
QEMU (BR-RTR2) - noVNC — Firefox Developer Edition
https://demo.peacedeath.su:7015/?console=kvm&novnc=1&vmid=10002&vmname=
/etc/network/interfaces [----] 0 L:[ 1+30 31/ 31] *(596 / 596b) <EOF>
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# Example configuration
# auto eth0
# iface eth0 inet dhcp

auto eth0
<----->iface eth0 inet static
<----->address 172.16.5.2/28
<----->gateway 172.16.5.1

auto eth1
<----->iface eth1 inet static
<----->address 192.168.4.1/27

auto gre1
  iface gre1 inet tunnel
  address 10.10.10.2
  netmask 255.255.255.252
  mode gre
  local 172.16.5.2
  endpoint 172.16.4.2
  ttl 255
```

И теперь удаляем существующий туннель в системе:

**ip tunnel del gre1**

И удаляем его ещё из **frr**:

**vtysh**

**conf t**

**no interface gre1**

AUTHORS:  
NECHAEV  
NAUMOV  
NAGORNOVA

```
QEMU (BR-RTR2) - noVNC — Firefox Developer Edition
https://demo.peacedeath.su:7015/?console=kvm&novnc=1
root@br-rtr:~# ip tunnel del gre1
root@br-rtr:~# vtysh

Hello, this is FRRouting (version 6.0.2).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

br-rtr.au-team.irpo# conf t
br-rtr.au-team.irpo(config)# no interface gre1
br-rtr.au-team.irpo(config)# end
br-rtr.au-team.irpo# exit
root@br-rtr:~#
```

И перезагружаем саму службу **ipsec**:

### **ipsec restart**

И можно проверить статус службы, чтобы узнать поднят ли туннель на обеих сторонах:

### **ipsec status**

```
QEMU (BR-RTR2) - noVNC — Firefox Developer Edition
https://demo.peacedeath.su:7015/?console=kvm&novnc=1&vmid=10002&vmname=BR-RTR2&node=altho
root@br-rtr:~# ipsec restart
Stopping strongSwan IPsec...
Starting strongSwan 5.5.1 IPsec [starter]...
root@br-rtr:~# ipsec status
Security Associations (1 up, 0 connecting):
    tunnel[1]: ESTABLISHED 5 seconds ago, 172.16.5.2[172.16.5.2]...172.16.4.2[172.16.4.2]
    tunnel{1}:  INSTALLED, TUNNEL, reqid 1, ESP SPIs: c54c28dc_i c68e8ab9_o
    tunnel{1}:   0.0.0.0/0 == 0.0.0.0/0
root@br-rtr:~#
```

Также можно проверить передаются ли зашифрованные пакеты по сети, для этого нам пригодится утилита **tcpdump**:

### **apt install tcpdump -y**

AUTHORS:  
NECHAEV  
NAUMOV  
NAGORNOVA

```
QEMU (BR-RTR2) - noVNC — Firefox Developer Edition
https://demo.peacedeath.su:7015/?console=kvm&novnc=1&vmid=10002&vmname=BR-RTR2&node=althome&resize=c
root@br-rtr:~# apt-get install tcpdump -y
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Следующий пакет устанавливался автоматически и больше не требуется:
  libports25
Для его удаления используйте «apt autoremove».
Будут установлены следующие дополнительные пакеты:
  librsar0.8
НОВЫЕ пакеты, которые будут установлены:
  librsar0.8 tcpdump
обновлено 0, установлено 2 новых пакетов, для удаления отмечено 0 пакетов, и 0 пакетов не обновлено.
Необходимо скачать 530 кБ архивов.
После данной операции, объём занятого дискового пространства возрастёт на 1 448 кБ.
0% [обработка]_
```

И теперь мы можем проверить это, пропишем на роутере **BR-RTR** команду:

**tcpdump -i eth0 -n -p esp**

А на роутере **HQ-RTR** отправим эхо-запрос:

**ping 10.10.10.2**

Как можно заметить, на правом роутере мы видим зашифрованные пакеты с меткой **ESP**.

```
QEMU (HQ-RTR2) - noVNC — Firefox Developer Edition
https://demo.peacedeath.su:7015/?console=kvm&novnc=1&vmid=10001&v
root@hq-rtr:~# ping 10.10.10.2
PING 10.10.10.2 (10.10.10.2) 56(84) bytes of data:
64 bytes from 10.10.10.2: icmp_seq=1 ttl=64 time=1.15 ms
64 bytes from 10.10.10.2: icmp_seq=2 ttl=64 time=1.06 ms
^C
--- 10.10.10.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.064/1.108/1.152/0.044 ms
root@hq-rtr:~# _

QEMU (BR-RTR2) - noVNC — Firefox Developer Edition
https://demo.peacedeath.su:7015/?console=kvm&novnc=1&vmid=10002&vmname=BR-RTR2&v
root@br-rtr:~# tcpdump -i eth0 -n -p esp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
09:10:42.575364 IP 172.16.4.2 > 172.16.5.2: ESP(spi=0xc2f24837,seq=0x1b), length 136
09:10:42.575545 IP 172.16.5.2 > 172.16.4.2: ESP(spi=0xc7af46ff,seq=0x1a), length 136
09:10:43.576955 IP 172.16.4.2 > 172.16.5.2: ESP(spi=0xc2f24837,seq=0x1c), length 136
09:10:43.577127 IP 172.16.5.2 > 172.16.4.2: ESP(spi=0xc7af46ff,seq=0x1b), length 136
09:10:44.004207 IP 172.16.5.2 > 172.16.4.2: ESP(spi=0xc2f24837,seq=0x1d), length 120
09:10:45.610565 IP 172.16.4.2 > 172.16.5.2: ESP(spi=0xc7af46ff,seq=0x1d), length 120
09:10:54.004521 IP 172.16.5.2 > 172.16.4.2: ESP(spi=0xc2f24837,seq=0x1e), length 120
09:10:55.610936 IP 172.16.4.2 > 172.16.5.2: ESP(spi=0xc7af46ff,seq=0x1e), length 120
^C
8 packets captured
8 packets received by filter
0 packets dropped by kernel
root@br-rtr:~# _
```

Задание выполнено!

#### 4. Настройте межсетевой экран на маршрутизаторах **HQ-RTR** и **BR-RTR** на сеть в сторону ISP

Для выполнения этого задания нам нужно обеспечить работу только нужных протоколов, а именно: HTTP, HTTPS, DNS, NTP, ICMP. А также запретить остальные подключения из сети Интернет во внутреннюю сеть.

#### **HQ-RTR**

Добавляем правила к уже существующим, которые были настроены в предыдущих модулях:

```
iptables -A INPUT -i lo -j ACCEPT
```

```
iptables -A INPUT -i eth0 -p tcp -m multiport --dports 22,53,80,443,2024 -j ACCEPT
```

```
iptables -A INPUT -i eth0 -p udp -m multiport --dports 53,123,500,4500 -j ACCEPT
```

```
iptables -A INPUT -i eth0 -p icmp -j ACCEPT
```

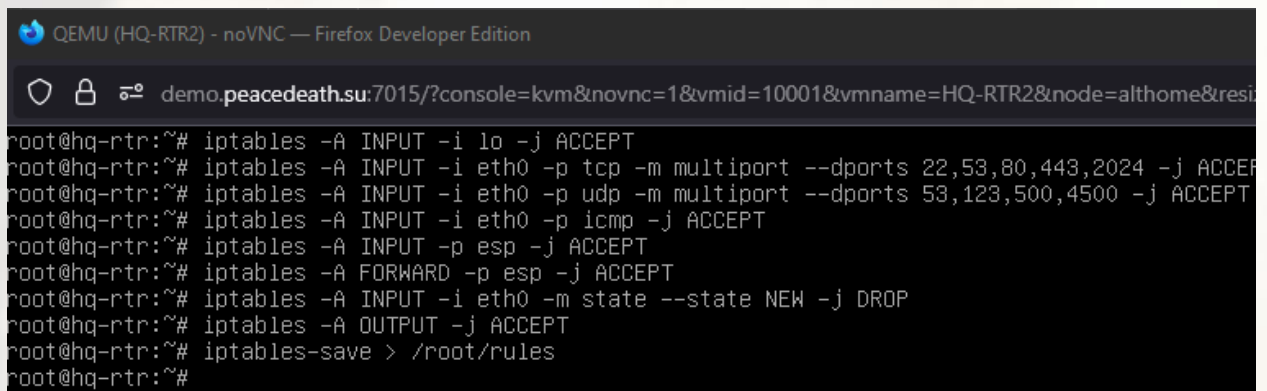
```
iptables -A INPUT -p esp -j ACCEPT
```

```
iptables -A FORWARD -p esp -j ACCEPT
```

```
iptables -A INPUT -i eth0 -m state --state NEW -j DROP
```

```
iptables -A OUTPUT -j ACCEPT
```

```
iptables-save > /root/rules
```



```
QEMU (HQ-RTR2) - noVNC — Firefox Developer Edition
demo.peacedeath.su:7015/?console=kvm&novnc=1&vmid=10001&vmname=HQ-RTR2&node=althome&resid=...
root@hq-rtr:~# iptables -A INPUT -i lo -j ACCEPT
root@hq-rtr:~# iptables -A INPUT -i eth0 -p tcp -m multiport --dports 22,53,80,443,2024 -j ACCEPT
root@hq-rtr:~# iptables -A INPUT -i eth0 -p udp -m multiport --dports 53,123,500,4500 -j ACCEPT
root@hq-rtr:~# iptables -A INPUT -i eth0 -p icmp -j ACCEPT
root@hq-rtr:~# iptables -A INPUT -p esp -j ACCEPT
root@hq-rtr:~# iptables -A FORWARD -p esp -j ACCEPT
root@hq-rtr:~# iptables -A INPUT -i eth0 -m state --state NEW -j DROP
root@hq-rtr:~# iptables -A OUTPUT -j ACCEPT
root@hq-rtr:~# iptables-save > /root/rules
root@hq-rtr:~#
```

В **crontab** изменения вносить не нужно, так как сохранение правил было в тот же файл.

Можно проверить наличие всех правил командой:

```
iptables -L -v
```

AUTHORS:  
NECHAEV  
NAUMOV  
NAGORNOVA



```

QEMU (HQ-RTR2) - noVNC — Firefox Developer Edition
demo.peacedeath.su:7015/?console=kvm&novnc=1&vmid=10001&vmname=HQ-RTR2&node=althome&resize=off

root@hq-rtr:~# iptables -L -v
Chain INPUT (policy ACCEPT 335 packets, 140K bytes)
 pkts bytes target    prot opt in     out     source                 destination
 20   3988 ACCEPT    all  --  lo      any     anywhere               anywhere
    0     0 ACCEPT    tcp  --  eth0    any     anywhere               anywhere          multiport dports ssh, domain, http, https, 2024
 26   4296 ACCEPT    udp  --  eth0    any     anywhere               anywhere          multiport dports domain, ntp, isakmp, ipsec-nat-t
    0     0 ACCEPT    icmp --  eth0    any     anywhere               anywhere
 315  152K ACCEPT    esp  --  any     any     anywhere               anywhere
    0     0 DROP     all  --  eth0    any     anywhere               anywhere          state NEW

Chain FORWARD (policy ACCEPT 1582 packets, 727K bytes)
 pkts bytes target    prot opt in     out     source                 destination
    0     0 ACCEPT    all  --  eth1    eth0    anywhere               anywhere
    0     0 ACCEPT    all  --  eth0    eth1    anywhere               anywhere
    0     0 ACCEPT    esp  --  any     any     anywhere               anywhere

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source                 destination
 967  205K ACCEPT    all  --  any     any     anywhere               anywhere
root@hq-rtr:~# _

```

## BR-RTR

Такие же правила добавляем на роутер справа и сохраняем их.

```

QEMU (BR-RTR2) - noVNC — Firefox Developer Edition
demo.peacedeath.su:7015/?console=kvm&novnc=1&vmid=10002&vmname=BR-RTR2&node=althome&resize=off

root@br-rtr:~# iptables -A INPUT -i lo -j ACCEPT
root@br-rtr:~# iptables -A INPUT -i eth0 -p tcp -m multiport --dport 22,53,80,443,2024 -j ACCEPT
root@br-rtr:~# iptables -A INPUT -i eth0 -p tcp -m multiport --dport 53,123,500,4500 -j ACCEPT
root@br-rtr:~# iptables -A INPUT -i eth0 -p icmp -j ACCEPT
root@br-rtr:~# iptables -A INPUT -p esp -j ACCEPT
root@br-rtr:~# iptables -A FORWARD -p esp -j ACCEPT
root@br-rtr:~# iptables -A INPUT -i eth0 -m state --state NEW -j DROP
root@br-rtr:~# iptables -A OUTPUT -j ACCEPT
root@br-rtr:~# iptables-save > /root/rules
root@br-rtr:~# _

```

И также проверить наличие всех правил командой:

## iptables -L -v

```

QEMU (BR-RTR2) - noVNC — Firefox Developer Edition
demo.peacedeath.su:7015/?console=kvm&novnc=1&vmid=10002&vmname=BR-RTR2&node=althome&resize=off&cmd=

root@br-rtr:~# iptables -L -v
Chain INPUT (policy ACCEPT 94 packets, 7765 bytes)
 pkts bytes target    prot opt in     out     source                 destination
    0     0 ACCEPT    all  --  lo      any     anywhere               anywhere
    6   460 ACCEPT    tcp  --  eth0    any     anywhere               anywhere          multiport dports ssh, domain, http, https, 2024
    0     0 ACCEPT    tcp  --  eth0    any     anywhere               anywhere          multiport dports domain, ntp, isakmp, 4500
    0     0 ACCEPT    icmp --  eth0    any     anywhere               anywhere
 170  31352 ACCEPT    esp  --  any     any     anywhere               anywhere
    0     0 DROP     all  --  eth0    any     anywhere               anywhere          state NEW

Chain FORWARD (policy ACCEPT 176 packets, 138K bytes)
 pkts bytes target    prot opt in     out     source                 destination
 77   4620 ACCEPT    all  --  eth1    eth0    anywhere               anywhere
 77  10395 ACCEPT    all  --  eth0    eth1    anywhere               anywhere
    0     0 ACCEPT    esp  --  any     any     anywhere               anywhere

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source                 destination
 384  152K ACCEPT    all  --  any     any     anywhere               anywhere
root@br-rtr:~# _

```

И проверим, не отвалился ли туннель ipsec после настройки правил на HQ-RTR:

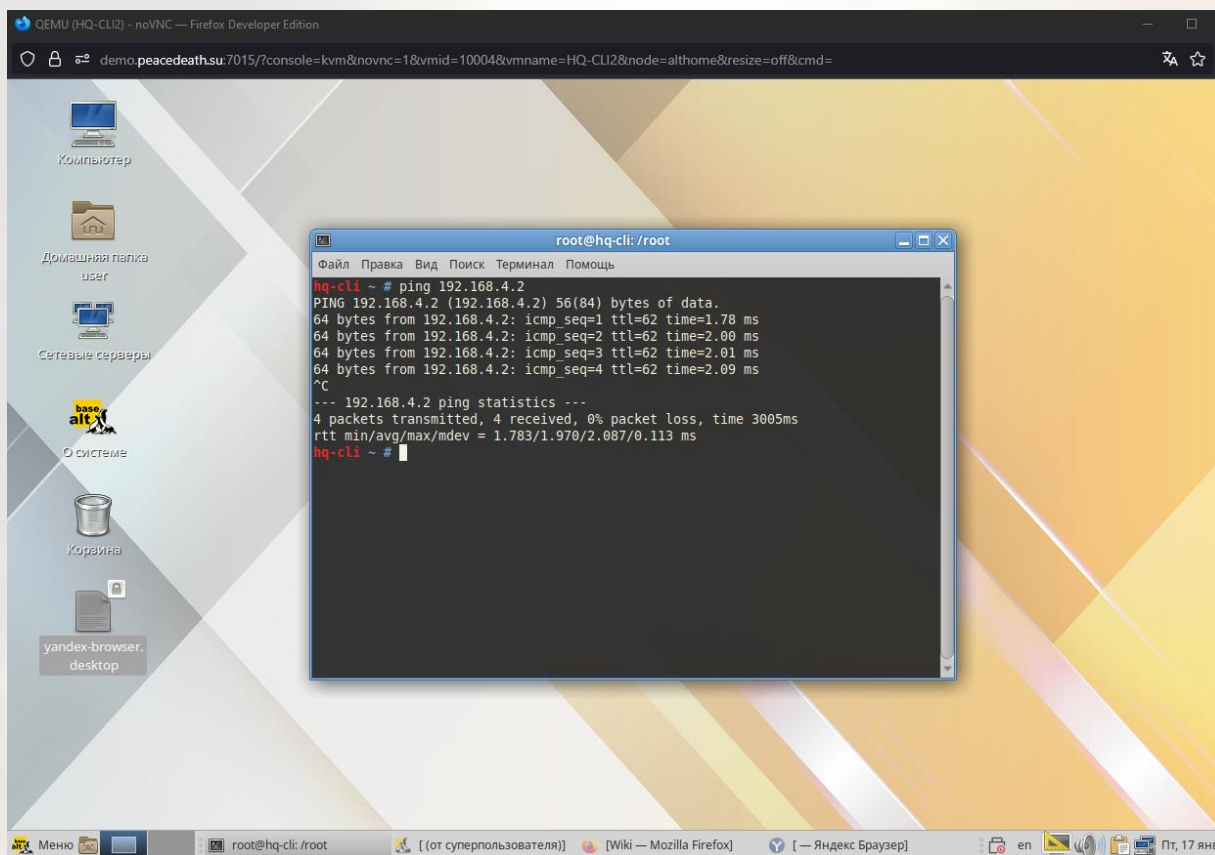
## ipsec status

```
root@hq-rtr:~# ipsec status
Security Associations (1 up, 0 connecting):
    tunnel[1]: ESTABLISHED 18 seconds ago, 172.16.4.2[172.16.4.2]...172.16.5.2[172.16.5.2]
    tunnel[2]: INSTALLED, TUNNEL, reqid 1, ESP SPIs: c4d6c29b_i cc88078a_o
    tunnel[2]: 0.0.0.0/0 == 0.0.0.0/0
root@hq-rtr:~# _
```

Видим, что соединение установлено и всё хорошо!

Проверим также наличие связи между конечными устройствами, отправим эхо-запрос с **HQ-CLI** на **BR-SRV**:

**ping 192.168.4.2**



Связь есть, всё отлично! Задание выполнено!

## 5. Настройте принт-сервер cups на сервере HQ-SRV.

Для начала необходимо установить пакеты cups и cups-pdf на **HQ-SRV**:

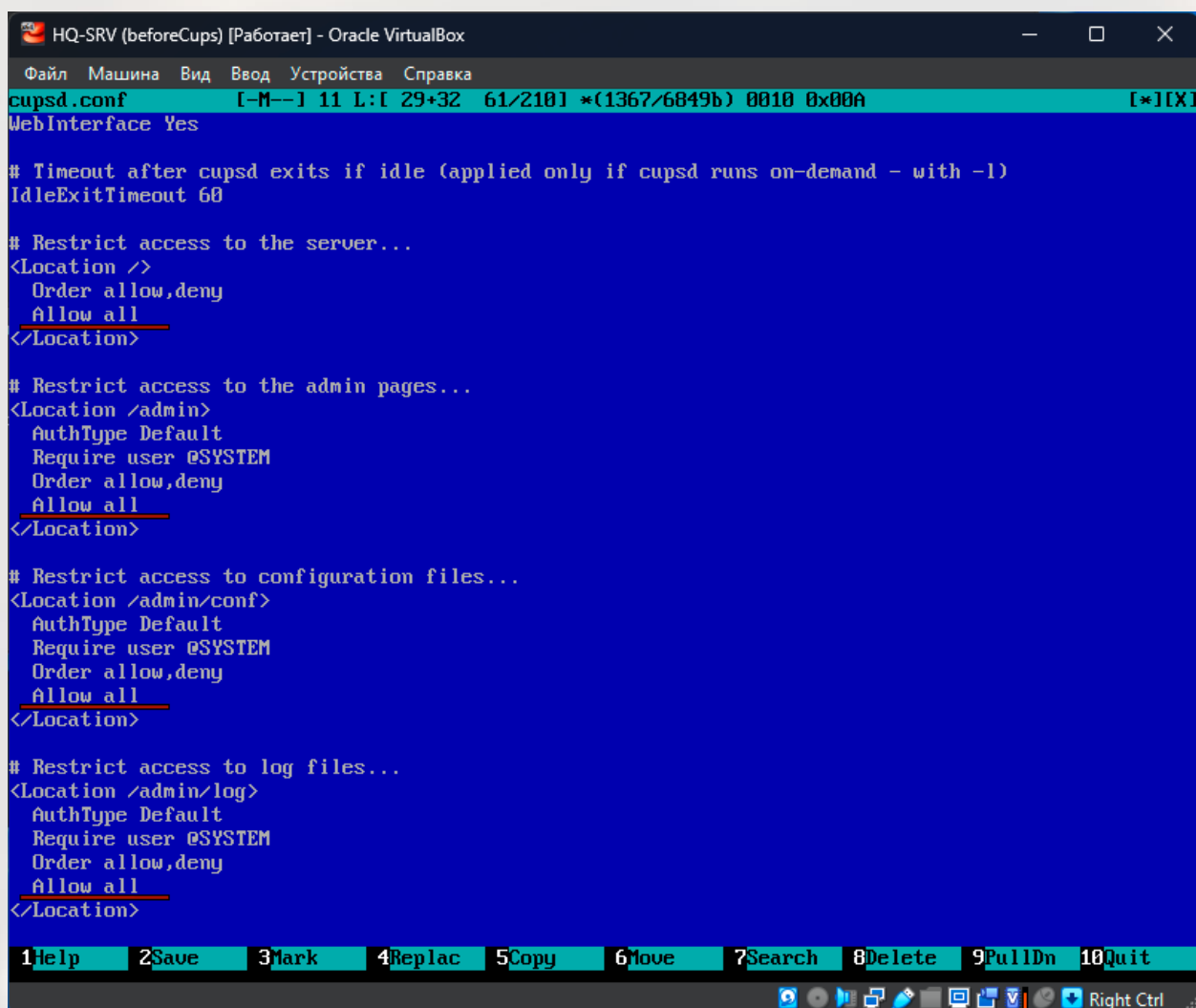
**apt-get install cups cups-pdf**

Теперь необходимо включить службу cups, чтобы она запускалась вместе с системой.

**systemctl enable --now cups**

Далее, необходимо отредактировать конфиг **/etc/cups/cupsd.conf**

Во всех блоках **Location** необходимо добавить строку **Allow all**, как на скриншоте:



```

HQ-SRV (beforeCups) [Работает] - Oracle VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
cupsd.conf  [-M--] 11 L:[ 29+32  61/210] *(1367/6849b) 0010 0x00A  [*][X]
WebInterface Yes

# Timeout after cupsd exits if idle (applied only if cupsd runs on-demand - with -l)
IdleExitTimeout 60

# Restrict access to the server...
<Location />
  Order allow,deny
  Allow all
</Location>

# Restrict access to the admin pages...
<Location /admin>
  AuthType Default
  Require user @SYSTEM
  Order allow,deny
  Allow all
</Location>

# Restrict access to configuration files...
<Location /admin/conf>
  AuthType Default
  Require user @SYSTEM
  Order allow,deny
  Allow all
</Location>

# Restrict access to log files...
<Location /admin/log>
  AuthType Default
  Require user @SYSTEM
  Order allow,deny
  Allow all
</Location>

1Help  2Save  3Mark  4Replac  5Copy  6Move  7Search  8Delete  9PullDn  10Quit
Right Ctrl
```

Перезапускаем службу **cups** для применения изменений:

**systemctl restart cups**

Переходим к подключению клиента **HQ-CLI**

На **HQ-CLI** выполняем следующую команду для подключения к принт-серверу:

**lpadm -p CUPS -E -v ipp://hq-srv.au-team.irpo:631/printers/Cups-PDF -m everywhere**

Установим принтер CUPS, как принтер по умолчанию:

**lptions -d CUPS**

```
root@hq-cli: /root
Файл Правка Вид Поиск Терминал Помощь
hq-cli ~ #
hq-cli ~ # lpadmin -p CUPS -E -v ipp://hq-srv.au-team.irpo:631/printers/Cups-PDF -m e
verywhere
hq-cli ~ # lptions -d CUPS
copies=1 device-uri=ipp://hq-srv.au-team.irpo:631/printers/Cups-PDF finishings=3 job-
cancel-after=10800 job-hold-until=no-hold job-priority=50 job-sheets=none,none marker
-change-time=0 number-up=1 printer-commands=none printer-info=Cups-PDF printer-is-acc
epting-jobs=true printer-is-shared=true printer-is-temporary=false printer-location p
rinter-make-and-model='CUPS-PDF Printer (w/ options) - IPP Everywhere' printer-state=
3 printer-state-change-time=1743213194 printer-state-reasons=none printer-type=61516
printer-uri-supported=ipp://localhost/printers/CUPS
hq-cli ~ #
```

Проверяем наличие принтера командой **lpstat -p**

```
hq-cli ~ # lpstat -p
принтер CUPS свободен. Включен с момента Сб 29 мар 2025 08:53:14
принтер Cups-PDF свободен. Включен с момента Пн 17 мар 2025 21:05:30
hq-cli ~ #
```

Как можно заметить, принтер **CUPS** успешно подключен. Из-за того, что на **HQ-CLI** также установлен принт-сервер, можно отключить локальный принтер “**Cups-PDF**”, чтобы он не мешал.

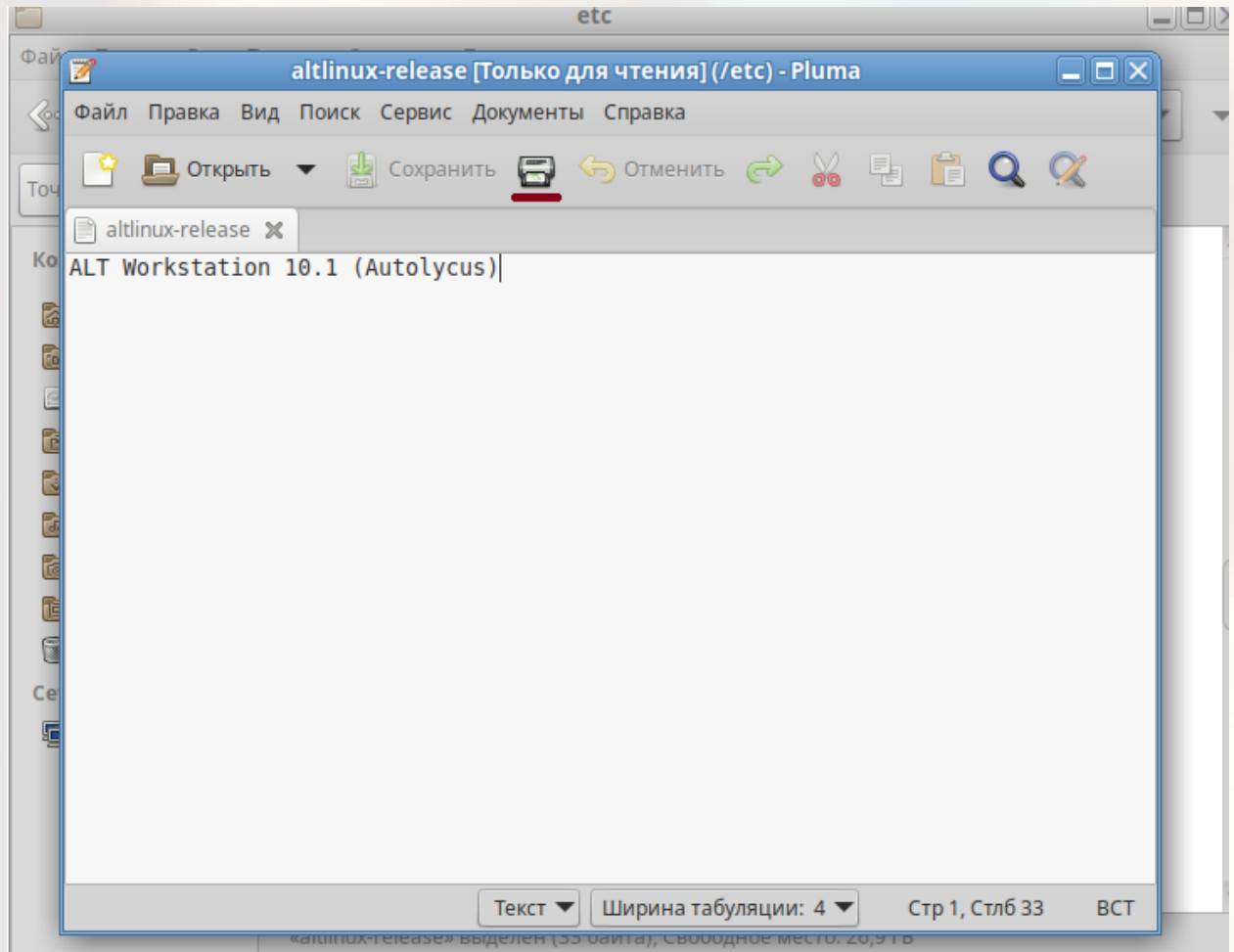
**lpadmin -x Cups-PDF**

```
hq-cli ~ # lpadmin -x Cups-PDF
hq-cli ~ # lpstat -p
принтер CUPS свободен. Включен с момента Сб 29 мар 2025 08:53:14
hq-cli ~ #
```

Теперь у нас остался один принтер. Проверим его работу. Откроем любой текстовый документ и попробуем его распечатать.

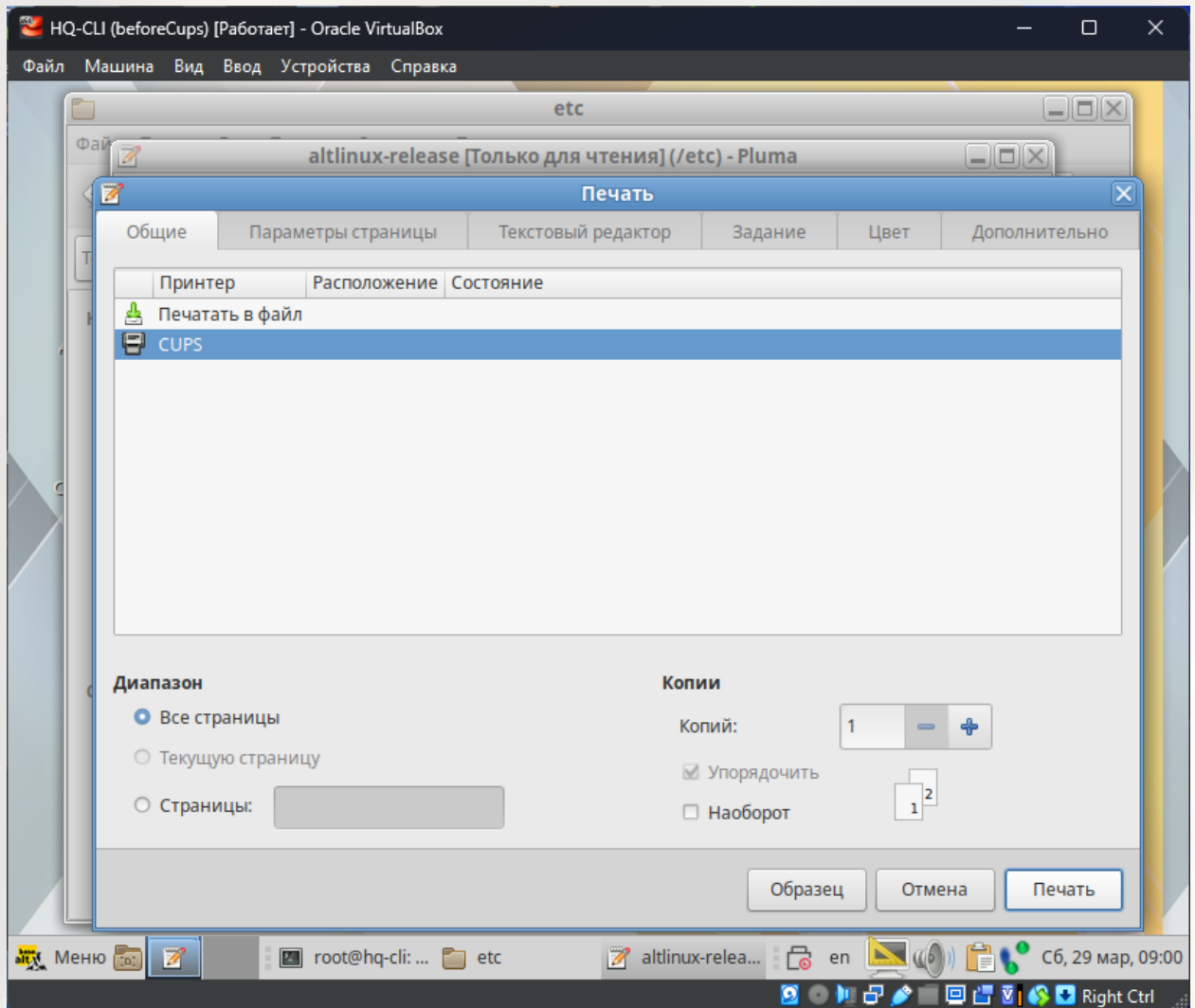
Допустим, откроем файл **/etc/altlinux-release**, нажимаем сверху значок печати:

AUTHORS:  
NECHAEV  
NAUMOV  
NAGORNOVA

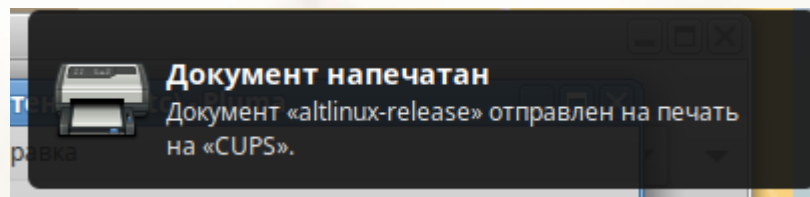


AUTHORS:  
NECHAEV  
NAUMOV  
NAGORNOVA

Выбираем наш принтер **CUPS** и жмем **Печать**



Сверху появится уведомление, что документ успешно напечатан.

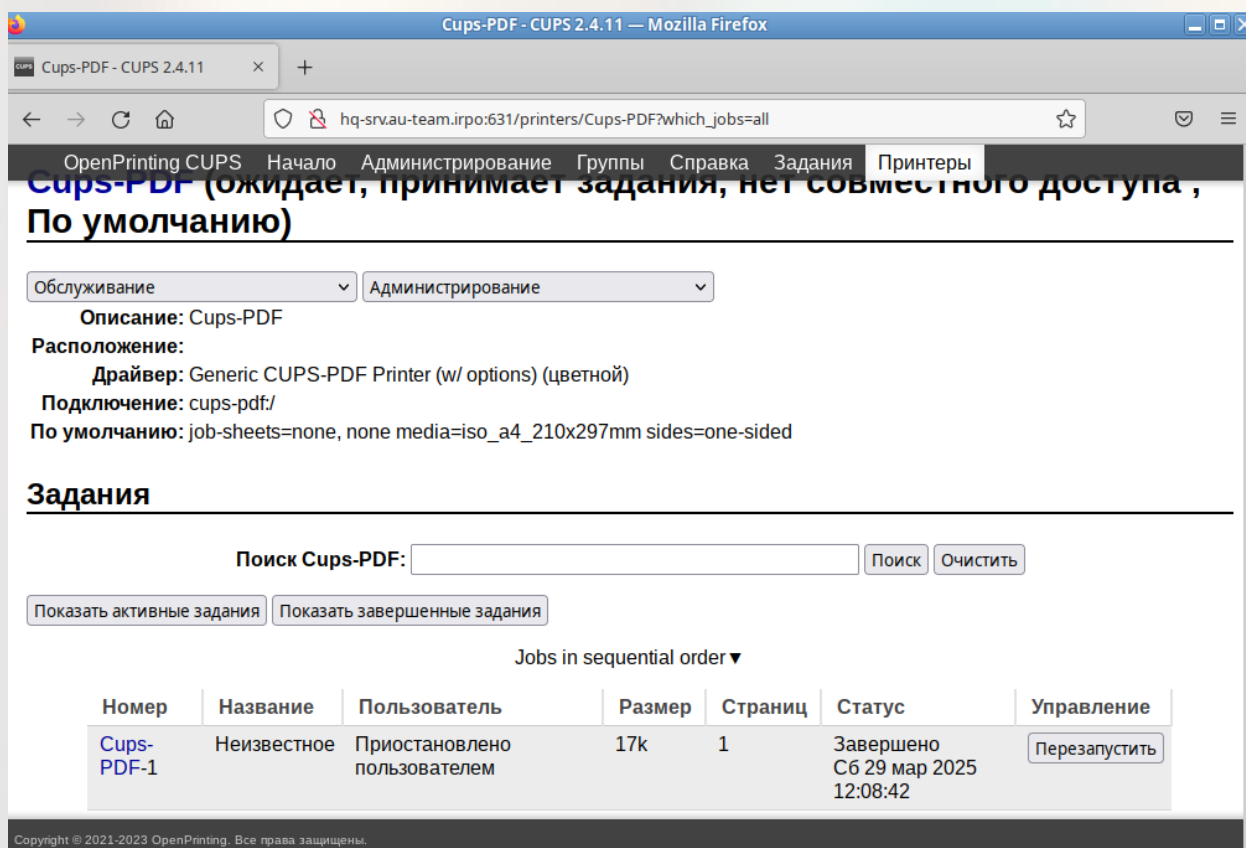


Перейдем в веб-интерфейс CUPS по адресу <https://hq-srv.au-team.irpo:631>

Вкладка **Принтеры**

Выбираем наш принтер.

Жмем кнопку **Показать все задания**



Как можно заметить, задание печати успешно завершилось и имеет статус “Завершено”.

Задание выполнено.

## 6. Реализуйте логирование при помощи rsyslog на устройствах HQ-RTR, BR-RTR, BR-SRV

Сперва необходимо настроить наш сервер для сбора логов.

Установим пакет **rsyslog** на **HQ-SRV**:

```
apt-get install rsyslog
```

Далее, отредактируем файл конфигурации, расположенный по пути **/etc/rsyslog.d/00\_common.conf**

Для передачи логов будем использовать протокол TCP, поэтому раскомментируем (уберем #) модуль **imtcp**, чтобы rsyslog мог получать логи с удаленных узлов.

Также необходимо в конец конфига добавить шаблон для сбора логов, чтобы rsyslog сохранял логи по пути, который указан в задании.



**\$template RemoteLogs, "/opt/%HOSTNAME%/rsyslog.txt"**

**.\* ?RemoteLogs**

**& stop**

Итоговый конфиг должен выглядеть так (измененные/добавленные строки отмечены ●):

```

##### MODULES #####

#module(load="imjournal") # provides support for systemd-journald logging
#module(load="imuxsock") # provides support for local system logging (e.g. via logger command)
#module(load="imklog") # provides kernel logging support (previously done by rklogd)
#module(load="immark") # provides --MARK-- message capability

# Provides UDP syslog reception
# for parameters see http://www.rsyslog.com/doc/imudp.html
#module(load="imudp") # needs to be done just once
#input(type="imudp" port="514")

# Provides TCP syslog reception
#for parameters see http://www.rsyslog.com/doc/imtcp.html
module(load="imtcp") # needs to be done just once ●
input(type="imtcp" port="514") ●

##### GLOBAL DIRECTIVES #####

# Use default timestamp format
module(load="builtin:omfile" Template="RSYSLOG_TraditionalFileFormat"
        DirCreateMode="0755"
        FileCreateMode="0640"
        fileOwner="root"
        fileGroup="adm")

# An "In-Memory Queue" is created for remote logging.
global(workDirectory="/var/spool/rsyslog") # where to place spool files

$template RemoteLogs, "/opt/%HOSTNAME%/rsyslog.txt" ●
.* ?RemoteLogs ●
& stop ●
  
```

Включаем службу **rsyslog**, чтобы она запускалась вместе с системой и перезапускаем ее для применения изменений:

**systemctl enable rsyslog**

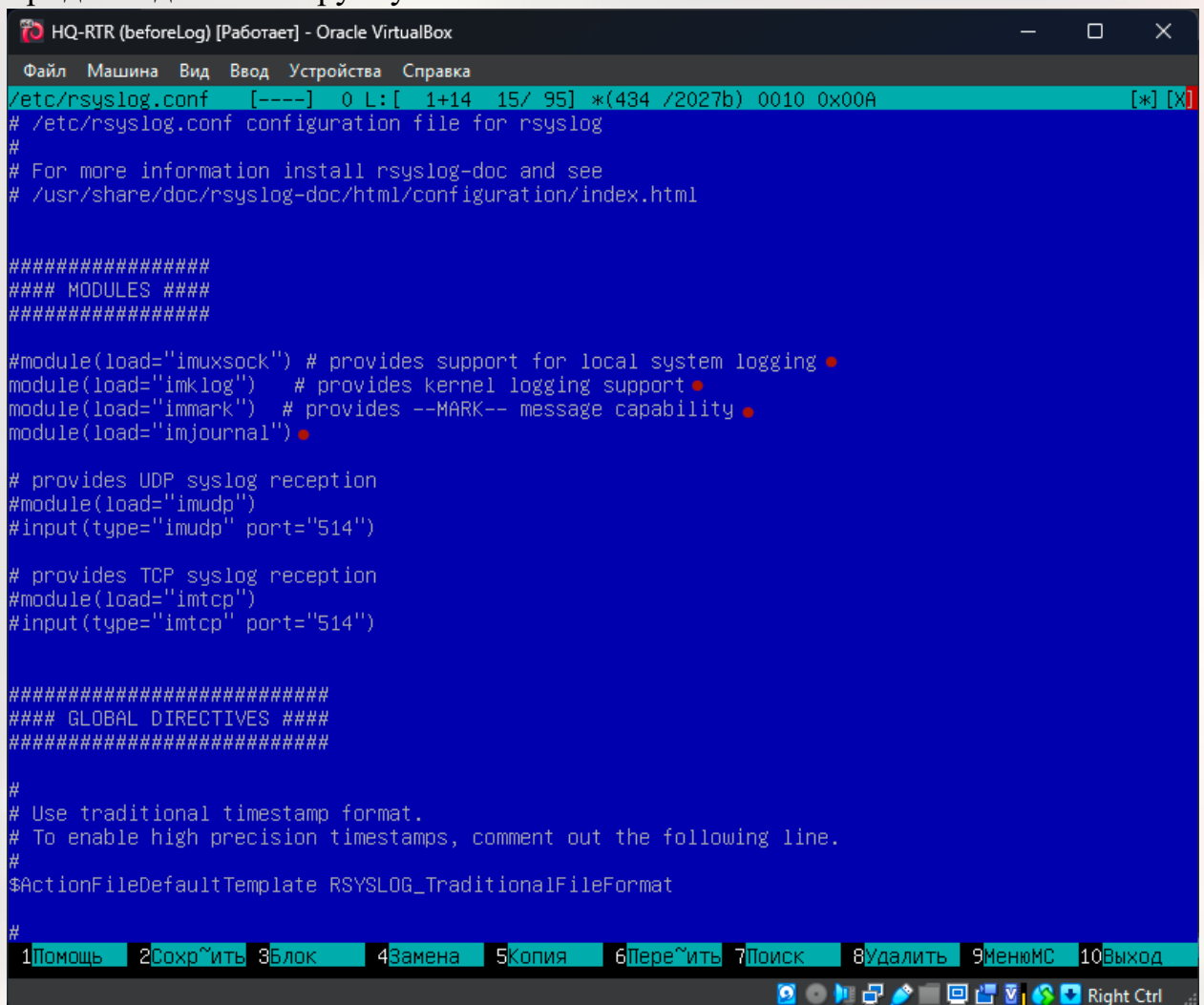
**systemctl restart rsyslog**

Сервер для приема логов настроен, переходим к настройке клиентов. Начнем с роутеров.



На **HQ-RTR** уже предустановлен пакет **rsyslog**, поэтому сразу перейдем к редактированию конфига **/etc/rsyslog.conf**

В блоке **MODULES** необходимо раскомментировать модули, которые обеспечивают поддержку логирования. (Все кроме модуля **imuxsock**, потому что вместо него будет использован модуль **imjournal**). Модуль **imjournal** придется дописать вручную.



```
HQ-RTR (beforeLog) [Работает] - Oracle VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
/etc/rsyslog.conf  [----]  0 L:[ 1+14 15/ 95] *(434 /2027b) 0010 0x00A  [X] [X]
# /etc/rsyslog.conf configuration file for rsyslog
#
# For more information install rsyslog-doc and see
# /usr/share/doc/rsyslog-doc/html/configuration/index.html
#####
#### MODULES ####
#####

#module(load="imuxsock") # provides support for local system logging •
module(load="imklog")    # provides kernel logging support •
module(load="immark")    # provides --MARK-- message capability •
module(load="imjournal") •

# provides UDP syslog reception
#module(load="imudp")
#input(type="imudp" port="514")

# provides TCP syslog reception
#module(load="imtcp")
#input(type="imtcp" port="514")

#####
#### GLOBAL DIRECTIVES ####
#####

#
# Use traditional timestamp format.
# To enable high precision timestamps, comment out the following line.
#
$ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat
#
1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перезагрузить 7Поиск 8Удалить 9Меню 10Выход
Right Ctrl
```

Теперь опускаемся в самый низ конфига, там расположены правила.

Добавляем в самый конец строку, которая отвечает за отправку логов уровня предупреждения (warning) и выше:

**\*.warning @@192.168.1.2:514**

AUTHORS:  
NECHAEV  
NAUMOV  
NAGORNOVA

```

HQ-RTR (beforeLog) [Работает] - Oracle VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
/etc/rsyslog.conf  [----] 27 L: [ 61+34  95/ 95] *(2026/2026b) <EOF>  [*] [X]
#
auth,authpriv.*<-----><----->/var/log/auth.log
*.*;auth,authpriv.none<----->-/var/log/syslog
#cron.*<-----><-----><----->/var/log/cron.log
daemon.*<-----><-----><----->-/var/log/daemon.log
kern.*<-----><-----><----->-/var/log/kern.log
lpr.*<-----><-----><----->-/var/log/lpr.log
mail.*<-----><-----><----->-/var/log/mail.log
user.*<-----><-----><----->-/var/log/user.log

#
# Logging for the mail system. Split it up so that
# it is easy to write scripts to parse these files.
#
mail.info<-----><-----><----->-/var/log/mail.info
mail.warn<-----><-----><----->-/var/log/mail.warn
mail.err<-----><-----><----->-/var/log/mail.err

#
# Some "catch-all" log files.
#
*.debug;\
<----->auth,authpriv.none;\
<----->mail.none<-----><----->-/var/log/debug
*.info;*.notice;*.warn;\
<----->auth,authpriv.none;\
<----->cron,daemon.none;\
<----->mail.none<-----><----->-/var/log/messages

#
# Emergencies are sent to everybody logged in.
#
*.emerg<-----><-----><----->:omusrmsg:*

*.warning @@192.168.1.2:514
1Помощь  2Сохранить  3Блок  4Замена  5Копия  6Пере~ить  7Поиск  8Удалить  9МенюМС  10Выход

```

Теперь перезапускаем службу **rsyslog**, чтобы применить изменения.

**systemctl restart rsyslog**

На **BR-RTR** нужно повторить аналогично.

Переходим к **BR-SRV**, здесь настройка почти такая же.

Установим на **BR-SRV** пакет **rsyslog**:

**apt-get install rsyslog rsyslog-journal**

Далее, отредактируем файл конфигурации, расположенный по пути **/etc/rsyslog.d/00\_common.conf**

Здесь также необходимо раскомментировать модули **imjournal**, **imklog**, **immark**. И добавить строку в конец конфига для того, чтобы логи отправлялись на сервер.

```

BR-SRV (beforeLog) [Работает] - Oracle VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
00_common.conf  [-M--] 27 L:[ 1+29 30/ 30] *(1080/1080b) <EOF>  [*][X]
#### MODULES ####

module(load="imjournal") # provides support for systemd-journald logging
module(load="imuxsock") # provides support for local system logging (e.g. via logger command)
module(load="imklog") # provides kernel logging support (previously done by rklogd)
module(load="immark") # provides --MARK-- message capability

# Provides UDP syslog reception
# for parameters see http://www.rsyslog.com/doc/imudp.html
module(load="imudp") # needs to be done just once
input(type="imudp" port="514")

# Provides TCP syslog reception
# for parameters see http://www.rsyslog.com/doc/imtcp.html
module(load="imtcp") # needs to be done just once
input(type="imtcp" port="514")

#### GLOBAL DIRECTIVES ####

# Use default timestamp format
module(load="builtin:omfile" Template="RSYSLOG_TraditionalFileFormat"
        DirCreateMode="0755"
        FileCreateMode="0640"
        fileOwner="root"
        fileGroup="adm")

# An "In-Memory Queue" is created for remote logging.
global(workDirectory="/var/spool/rsyslog") # where to place spool files

*.warning @@192.168.1.2:514_

1Help  2Save  3Mark  4Replac  5Copy  6Move  7Search  8Delete  9PullDn  10Quit

```

Включаем службу **rsyslog**, чтобы она запускалась вместе с системой и перезапускаем ее для применения изменений:

**systemctl enable rsyslog**

**systemctl restart rsyslog**

За время пока выполнялась настройка клиентов уже должны появиться логи, проверим каталог **/opt HQ-SRV**

```

[root@hq-srv opt]# ls
br-rtr  br-srv  hq-rtr
[root@hq-srv opt]# _

```

AUTHORS:

NECHAEV

NAUMOV

NAGORNOV

Как можно заметить, были автоматически созданы каталоги с именами клиентов. В каждом из них есть файл **rsyslog.txt**

Проверим, что логируются только сообщения уровня warning и выше.

Добавим несколько записей различного уровня в лог на любом из клиентов, например на **BR-SRV**, командами:

**logger -p user.info "Test info"**

Также добавим сообщения уровня **warning**:

**logger -p user.warning "Test warning"**

Также добавим сообщения уровня **error**:

**logger -p user.error "Test error"**

```
[root@br-srv ~]# logger -p user.info "Test info"
[root@br-srv ~]# logger -p user.warning "Test warning"
[root@br-srv ~]# logger -p user.error "Test error"
[root@br-srv ~]#
```

Теперь проверим на **HQ-SRV** содержимое файла **/opt/br-srv/rsyslog.txt**

```
[root@hq-srv opt]# cat /opt/br-srv/rsyslog.txt
Mar 29 18:10:34 br-srv kernel: [Firmware Bug]: TSC doesn't count with P0 frequency!
Mar 29 18:10:34 br-srv kernel: -----
Mar 29 18:10:34 br-srv kernel: | NMI testsuite:
Mar 29 18:10:34 br-srv kernel: -----
Mar 29 18:10:34 br-srv kernel:      remote IPI: ok |
Mar 29 18:10:34 br-srv kernel:      local IPI:  ok |
Mar 29 18:10:34 br-srv kernel: -----
Mar 29 18:10:34 br-srv kernel: Good, all    2 testcases passed! |
Mar 29 18:10:34 br-srv kernel: -----
Mar 29 18:10:34 br-srv kernel: pmd_set_huge: Cannot satisfy [mem 0xf0000000-0xf0200000] with a huge-
page mapping due to MTRR override.
Mar 29 18:10:34 br-srv kernel: [drm:umw_host_printf [umwgfx]] *ERROR* Failed to send host log messag
e.
Mar 29 18:10:34 br-srv kernel: device-mapper: core: CONFIG_IMA_DISABLE_HTABLE is disabled. Duplicate
IMA measurements will not be recorded in the IMA log.
Mar 29 18:10:34 br-srv kernel: clocksource: Long readout interval, skipping watchdog check: cs_nsec:
1689036074 wd_nsec: 1689035304
Mar 29 18:10:34 br-srv kernel: clocksource: Long readout interval, skipping watchdog check: cs_nsec:
15367950705 wd_nsec: 15367943690
Mar 29 18:10:34 br-srv kernel: clocksource: Long readout interval, skipping watchdog check: cs_nsec:
40938221208 wd_nsec: 40938202554
Mar 29 18:10:34 br-srv kernel: clocksource: Long readout interval, skipping watchdog check: cs_nsec:
92744393523 wd_nsec: 92744351242
Mar 29 18:23:32 br-srv root: Test warning
Mar 29 18:23:37 br-srv root: Test error
[root@hq-srv opt]#
```

Как можно заметить, здесь появились только сообщения уровня **warning** и **error**.

Перейдем к настройке ротации логов. На **HQ-SRV** создадим файл **/etc/logrotate.d/rsyslog**

AUTHORS:

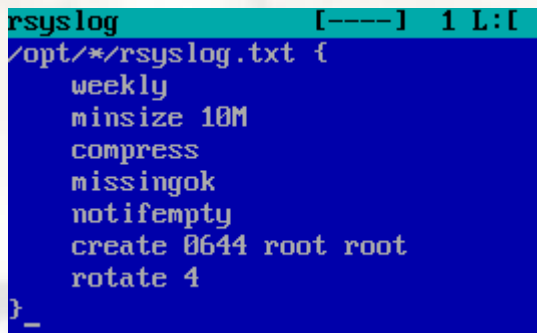
NECHAEV

NAUMOV

NAGORNOVA

Запишем в него следующее содержимое:

```
/opt/*/rsyslog.txt {  
    weekly  
    minsize 10M  
    compress  
    missingok  
    notifempty  
    create 0644 root root  
    rotate 4  
    dateext  
}
```



```
rsyslog [----] 1 L:  
/opt/*/rsyslog.txt {  
    weekly  
    minsize 10M  
    compress  
    missingok  
    notifempty  
    create 0644 root root  
    rotate 4  
}_
```

Настройка ротации на этом закончена, каждую неделю будут проверяться логи и если какие-то из них больше 10МБ, они будут сжаты в архив.

**7. На сервере HQ-SRV реализуйте мониторинг устройств с помощью открытого программного обеспечения.**

**8. Реализуйте механизм инвентаризации машин HQ-SRV и HQ-CLI через Ansible на BR-SRV**

Для начала необходимо создать каталог, в котором будут размещены отчеты о рабочих местах:

```
mkdir /etc/ansible/PC_INFO
```

Далее, создадим плейбук `/etc/ansible/inventory.yml` со следующим содержимым:

---

- name: Инвентаризация машин HQ-SRV и HQ-CLI

hosts:

- HQ-SRV

- HQ-CLI

gather\_facts: yes

tasks:

- name: Создать отчёт с информацией рабочем месте

delegate\_to: localhost

copy:

dest: "/etc/ansible/PC\_INFO/{{ ansible\_hostname }}.yaml"

content: |

---

Имя компьютера: "{{ ansible\_hostname }}"

IP-адрес компьютера: "{{ ansible\_default\_ipv4.address }}"

```
inventory.yml  [----]  0 L:1  1+15  16/ 16]  *(509 / 509b) <EOF>
---
- name: Инвентаризация машин HQ-SRV и HQ-CLI
  hosts:
    - HQ-SRV
    - HQ-CLI
  gather_facts: yes
  tasks:
    - name: Создать отчёт с информацией рабочем месте
      delegate_to: localhost
      copy:
        dest: "/etc/ansible/PC_INFO/{{ ansible_hostname }}.yaml"
        content: |
          ---
          Имя компьютера: "{{ ansible_hostname }}"
          IP-адрес компьютера: "{{ ansible_default_ipv4.address }}"
```

AUTHORS  
NECHAEV  
NAUMOV  
NAGORNOVA

Проверим работу, командой:

**ansible-playbook /etc/ansible/inventory.yml**

```
[root@br-srv ~]# ansible-playbook /etc/ansible/inventory.yml

PLAY [Инвентаризация машин HQ-SRV и HQ-CLI] *****

TASK [Gathering Facts] *****
ok: [HQ-SRV]
ok: [HQ-CLI]

TASK [Создать отчёт с информацией рабочем месте] *****
changed: [HQ-CLI -> localhost]
changed: [HQ-SRV -> localhost]

PLAY RECAP *****
HQ-CLI                : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=
0 ignored=0
HQ-SRV                : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=
0 ignored=0

[root@br-srv ~]#
```

Ansible помечает результат как changed, так как фактическое состояние системы меняется. При первом запуске плейбука это ожидаемое поведение.

Если запустить плейбук ещё раз, то Ansible покажет для тех же задач статус ok, потому что требуемое состояние уже достигнуто и ничего менять не нужно.

Проверим наличие и содержимое, созданных отчетов:

**ls -la /etc/ansible/PC\_INFO**

**cat /etc/ansible/PC\_INFO/hq-cli.yml**

**cat /etc/ansible/PC\_INFO/hq-srv.yml**

```
[root@br-srv ~]# ls -la /etc/ansible/PC_INFO/
total 16
drwxr-xr-x 2 root root 4096 Mar 29 15:04 .
drwxr-xr-x 3 root root 4096 Mar 29 15:02 ..
-rw-r--r-- 1 root root  93 Mar 29 15:04 hq-cli.yml
-rw-r--r-- 1 root root  92 Mar 29 15:04 hq-srv.yml
[root@br-srv ~]# cat /etc/ansible/PC_INFO/hq-cli.yml
-----
Имя компьютера: "hq-cli"
IP-адрес компьютера: "192.168.2.10"
[root@br-srv ~]# cat /etc/ansible/PC_INFO/hq-srv.yml
-----
Имя компьютера: "hq-srv"
IP-адрес компьютера: "192.168.1.2"
[root@br-srv ~]# _
```

AUTHORS:  
NECHAEV  
NAUMOV  
NAGORNOVA

Как можно заметить, отчеты созданы и содержат необходимую информацию.

Задание выполнено.



## 9. Реализуйте механизм резервного копирования конфигурации для машин HQ-RTR и BR-RTR, через Ansible на BR-SRV

Спасибо за содействие подписчикам нашего канала, что предоставили плейбук для этого задания!

Создадим также каталог, в котором будут размещены резервные копии конфигураций маршрутизаторов:

```
mkdir /etc/ansible/NETWORK_INFO
```

И создаём сам плейбук **/etc/ansible/backup.yml** со следующим содержимым:

---

**- name: Резервное копирование конфигурации маршрутизаторов HQ-RTR и BR-RTR**

**hosts:**

**- HQ-RTR**

**- BR-RTR**

**gather\_facts: no**

**tasks:**

**- name: Создание локальных папок для резервных копий**

**ansible.builtin.file:**

**path: "/etc/ansible/NETWORK\_INFO/{{ inventory\_hostname }}/{{ item }}**

**state: directory**

**loop:**

**- "frr"**

**delegate\_to: localhost**

**- name: Копирование конфигурации FRR**

**ansible.builtin.fetch:**

**src: "/etc/frr/{{ item }}**

AUTHORS:  
NECHAEV  
NAUMOV  
NAGORNOVA



**dest: "/etc/ansible/NETWORK\_INFO/{{ inventory\_hostname }}/frr/"**

**flat: yes**

**loop:**

- "daemons"**
- "frr.conf"**
- "frr.conf.sav"**
- "vtysh.conf"**

**become: yes**

**- name: Копирование сохранённых правил iptables**

**ansible.builtin.fetch:**

**src: /etc/iptablesRules**

**dest: /etc/ansible/NETWORK\_INFO/{{ inventory\_hostname  
}}/iptablesRules**

**flat: yes**

**become: yes**

**- name: Копирование конфигурации сетевых интерфейсов**

**ansible.builtin.fetch:**

**src: /etc/network/interfaces**

**dest: /etc/ansible/NETWORK\_INFO/{{ inventory\_hostname }}/interfaces**

**flat: yes**

**become: yes**

AUTHORS:  
NECHAEV  
NAUMOV  
NAGORNOVA

```
backup.yml [-----] 15 L:[ 1+13 14/ 42] *(441 /1343b) 0010 0x00A [*IXI]
-----
- name: Резервное копирование конфигурации маршрутизаторов HQ-RTR и BR-RTR
  hosts:
    - HQ-RTR
    - BR-RTR
  gather_facts: no
  tasks:
    - name: Создание локальных папок для резервных копий
      ansible.builtin.file:
        path: "/etc/ansible/NETWORK_INFO/{{ inventory_hostname }}/{{ item }}"
        state: directory
      loop:
        - "frr"
      delegate_to: localhost

    - name: Копирование конфигурации FRR
      ansible.builtin.fetch:
        src: "/etc/frr/{{ item }}"
        dest: "/etc/ansible/NETWORK_INFO/{{ inventory_hostname }}/frr/"
        flat: yes
      loop:
        - "daemons"
        - "frr.conf"
        - "frr.conf.sav"
        - "vtysh.conf"
      become: yes

    - name: Копирование сохранённых правил iptables
      ansible.builtin.fetch:
        src: /etc/iptablesRules
        dest: /etc/ansible/NETWORK_INFO/{{ inventory_hostname }}/iptablesRules
        flat: yes
      become: yes

    - name: Копирование конфигурации сетевых интерфейсов
      ansible.builtin.fetch:
        src: /etc/network/interfaces
        dest: /etc/ansible/NETWORK_INFO/{{ inventory_hostname }}/interfaces
        flat: yes
      become: yes
```

1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn 10Quit

Абсолютно также, как и в предыдущем задании, проверяем его работу, командой:

**ansible-playbook /etc/ansible/backup.yml**

AUTHORS:  
NECHAEV  
NAUMOV  
NAGORNOVA

```

[root@br-srv ansible]# ansible-playbook /etc/ansible/backup.yml
PLAY [Резервное копирование конфигурации маршрутизаторов HQ-RTR и BR-RTR] *****
TASK [Создание локальных папок для резервных копий] *****
changed: [BR-RTR -> localhost] => (item=frr)
changed: [HQ-RTR -> localhost] => (item=frr)

TASK [Копирование конфигурации FRR] *****
changed: [BR-RTR] => (item=daemons)
changed: [HQ-RTR] => (item=daemons)
changed: [BR-RTR] => (item=frr.conf)
changed: [HQ-RTR] => (item=frr.conf)
changed: [BR-RTR] => (item=frr.conf.sav)
changed: [HQ-RTR] => (item=frr.conf.sav)
changed: [BR-RTR] => (item=vttysh.conf)
changed: [HQ-RTR] => (item=vttysh.conf)

TASK [Копирование сохранённых правил iptables] *****
changed: [BR-RTR]
changed: [HQ-RTR]

TASK [Копирование конфигурации сетевых интерфейсов] *****
changed: [HQ-RTR]
changed: [BR-RTR]

PLAY RECAP *****
BR-RTR                : ok=4    changed=4    unreachable=0    failed=0    skipped=0    rescued=0
0 ignored=0
HQ-RTR                : ok=4    changed=4    unreachable=0    failed=0    skipped=0    rescued=0
0 ignored=0

[root@br-srv ansible]#

```

Как и в прошлом задании, Ansible помечает результат как **changed**, так как фактическое состояние системы меняется. При первом запуске плейбука так и должно быть.

И если запустить его ещё раз, то Ansible покажет для тех же задач статус **ok**, потому что требуемое состояние уже достигнуто и ничего менять не нужно.

Проверим наличие созданных отчетов:

```
ls -la /etc/ansible/NETWORK_INFO
```

```
ls -la /etc/ansible/NETWORK_INFO/HQ-RTR
```

```
ls -la /etc/ansible/NETWORK_INFO/BR-RTR
```

AUTHORS:  
NECHAEV  
NAUMOV  
NAGORNOVA

```
[root@br-srv ansible]# ls -la /etc/ansible/NETWORK_INFO/
total 16
drwxr-xr-x 4 root root 4096 Apr 13 21:00 .
drwxr-xr-x 4 root root 4096 Apr 13 21:00 ..
drwxr-xr-x 3 root root 4096 Apr 13 21:00 BR-RTR
drwxr-xr-x 3 root root 4096 Apr 13 21:00 HQ-RTR
[root@br-srv ansible]# ls -la /etc/ansible/NETWORK_INFO/HQ-RTR
total 20
drwxr-xr-x 3 root root 4096 Apr 13 21:00 .
drwxr-xr-x 4 root root 4096 Apr 13 21:00 ..
drwxr-xr-x 2 root root 4096 Apr 13 21:00 frr
-rw-r--r-- 1 root root 678 Apr 13 21:00 interfaces
-rw-r--r-- 1 root root 1305 Apr 13 21:00 iptablesRules
[root@br-srv ansible]# ls -la /etc/ansible/NETWORK_INFO/BR-RTR
total 20
drwxr-xr-x 3 root root 4096 Apr 13 21:00 .
drwxr-xr-x 4 root root 4096 Apr 13 21:00 ..
drwxr-xr-x 2 root root 4096 Apr 13 21:00 frr
-rw-r--r-- 1 root root 443 Apr 13 21:00 interfaces
-rw-r--r-- 1 root root 1300 Apr 13 21:00 iptablesRules
[root@br-srv ansible]#
```

А также их содержимое, если хотите убедиться, что действительно скопировалось, для примера покажем файл **interfaces** с маршрутизатора **HQ-RTR**, остальные можете сами:

```
cat /etc/ansible/NETWORK_INFO/HQ-RTR/interfaces
```

AUTHORS:  
NECHAEV  
NAUMOV  
NAGORNOVA

```
# and how to activate them. For more information, see interfaces(5).
```

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

```
# The loopback network interface
```

```
auto lo
```

```
iface lo inet loopback
```

```
# Example configuration
```

```
# auto eth0
```

```
# iface eth0 inet dhcp
```

```
auto eth0
```

```
iface eth0 inet static
```

```
address 172.16.4.2/28
```

```
gateway 172.16.4.1
```

```
auto eth1
```

```
iface eth1 inet manual
```

```
auto eth1.100
```

```
iface eth1.100 inet static
```

```
address 192.168.1.1/26
```

```
vlan-raw-device eth1
```

```
auto eth1.200
```

```
iface eth1.200 inet static
```

```
address 192.168.2.1/28
```

```
vlan-raw-device eth1
```

```
auto eth1.999
```

```
iface eth1.999 inet static
```

```
address 192.168.3.1/29
```

```
vlan-raw-device eth1
```

```
auto gre1
```

```
[root@br-srv ansible]# _
```

По итогу все резервные копии конфигураций созданы и содержат необходимую информацию.

Задание выполнено.

AUTHORS:  
NECHAEV  
NAUMOV  
NAGORNOVA