

Выполните настройку центра сертификации на базе HQ-SRV

Выполните настройку центра сертификации на базе HQ-SRV

Требования:

- Необходимо использовать отечественные алгоритмы шифрования (мы этого делать не будем)
- Сертификаты выдаются на 365 дней
- Обеспечьте доверие сертификату для HQ-CLI
- Выдайте сертификаты веб серверам
- Перенастройте ранее настроенные веб сервера, moodle, wiki, реверсивный прокси nginx на протокол https (мы установим только на reverse-проху nginx)
- При обращении к веб серверам по их доменным именам у браузера клиента не должно возникать предупреждений

Создание центра сертификации (теория)

Центр сертификации (Certificate authority) состоит всего из 2 компонентов:

1. **Закрытый ключ**, имеет расширение .key, его мы должны хранить в тайне;
2. **Сертификат**, содержит открытый ключ, имеет расширение .crt.

Этапы создания СА:

1. Создание закрытого ключа (ca.key)
`openssl genrsa -out ca.key 2048`
2. Создание запроса на подпись (ca.csr)
`openssl req -key ca.key -new -out ca.csr`
3. Создание самоподписанного сертификата из запроса (ca.crt)
`openssl x509 -signkey ca.key -in ca.csr -req -days 365 -out ca.crt`

Как видите ничего особо сложного, но можно сделать намного проще, эти 3 действия можно выполнить всего 1 командой, именно так мы и поступим.

Создание центра сертификации (практика)

Для начала перейдите в домашнюю директорию пользователя:

```
cd
```

Сору

Организацию СА начнем со следующей команды:

```
openssl req -newkey rsa:4096 -nodes -keyout ca.key -x509 -days 365 -out ca.crt
```

Сору

Далее начнется интерактивный ввод параметров, заполните его следующим образом:

```
Generating a RSA private key
.....+++++
.....+++++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [RU]:
State or Province Name (full name) []: RU-BA
Locality Name (eg, city) []: UFA
Organization Name (eg, company) []: URKTB
Organizational Unit Name (eg, section) []: 308
Common Name (e.g., your name or your server's hostname) []: AU-Team CA
Email Address []:
```

Сору

Теперь мы имеем закрытый ключ, и сам сертификат (файла csr при этом не будет)

```
[root@hq-srv ~]# ls -l
total 12
-rw-r--r-- 1 root root 1159 Feb 28 15:57 ca.crt
-rw----- 1 root root 1708 Feb 28 15:56 ca.key
```

Сору

Создание сертификата для веб-сервера

Создаем закрытый ключ:

```
openssl genrsa -out web.key 4096
```

Сору

Запрос на подпись:

```
openssl req -key web.key -new -out web.csr
```

Сору

Здесь так же начнется ввод параметров:

```
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]: RU
State or Province Name (full name) [Some-State]: RU-BA
Locality Name (eg, city) []: UFA
Organization Name (eg, company) [Internet Widgits Pty Ltd]: UKRTB
Organizational Unit Name (eg, section) []:308
Common Name (e.g. server FQDN or YOUR name) []:*.au-team.irpo
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Сору

Мы использовали wildcard домен *.au-team.irpo, чтобы не генерировать сертификаты отдельно под wiki и moodle.

Создайте файл openssl.cnf:

```
[req]
req_extensions = req_ext

[req_ext]
#Эту строку указывать необязательно, так как Common Name сертификата wildcard
subjectAltName = DNS:wiki.au-team.irpo, DNS:moodle.au-team.irpo
extendedKeyUsage = serverAuth
keyUsage = digitalSignature
```

Сору

Этот файл указывает, что сертификат можно будет использовать для создания TLS на веб-сервере

Теперь подписываем этот запрос через наш CA:

```
openssl x509 -req -in web.csr -CA ca.crt -CAkey ca.key -CAcreateserial \
-out web.crt -days 365 -sha256 -extfile openssl.cnf -extensions req_ext
```

Сору

И того получаем:

```
[root@hq-srv ~]# ls -l
total 24
-rw-r--r-- 1 root root 1159 Feb 28 15:57 ca.crt
-rw----- 1 root root 1159 Feb 28 15:57 ca.key
-rw-r--r-- 1 root root 1159 Feb 28 15:57 ca.srl
-rw-r--r-- 1 root root 1159 Feb 28 15:57 web.crt
-rw-r--r-- 1 root root 1159 Feb 28 15:57 web.csr
-rw----- 1 root root 1159 Feb 28 15:57 web.key
```

Сору

Теперь осталось 2 задачи:

1. Настроить веб-сервер Nginx на работу с нашим сертификатом;
2. Установить доверие HQ-CLI к нашему CA (иначе он не будет доверять сертификату web.crt, который подписан нашим CA, а как следствие будет ошибка - "Подключение не защищено").

Установка сертификата на веб-сервер

Нам нужно закинуть web.crt и web.key на BR-SRV, где у нас находится Nginx, для этого выполняем:

```
scp -P 2024 web.crt sshuser@192.168.20.2:/home/sshuser
scp -P 2024 web.key sshuser@192.168.20.2:/home/sshuser
```

Copy

Затем на самом BR-SRV перенесем эти файлы в директорию /etc/nginx

```
mv /home/sshuser/web.crt /etc/nginx
mv /home/sshuser/web.key /etc/nginx
```

Copy

Теперь изменим конфиг /etc/nginx/sites-available.d/default.conf:

```
upstream moodle.au-team.irpo {
    server 192.168.10.2;
}

upstream wiki.au-team.irpo {
    server 192.168.20.2:8080;
}

#MOODLE
server {
    listen 80;
    server_name moodle.au-team.irpo;

    return 301 https://$host;
}

server {
    listen 443 ssl;
    server_name moodle.au-team.irpo;

    ssl_certificate /etc/nginx/web.crt;
    ssl_certificate_key /etc/nginx/web.key;

    location / {
        proxy_pass http://moodle.au-team.irpo;
    }
}

#MEDIAWIKI
server {
    listen 80;
    server_name wiki.au-team.irpo;

    return 301 https://$host;
}

server {
    listen 443 ssl;
    server_name wiki.au-team.irpo;

    ssl_certificate /etc/nginx/web.crt;
    ssl_certificate_key /etc/nginx/web.key;

    location / {
        proxy_pass http://wiki.au-team.irpo;
    }
}
```

Copy

После чего проверьте конфигурацию на ошибки:

```
nginx -t
```

Copy

Если все в порядке перезагрузите веб-сервер:

```
systemctl restart nginx
```

Copy

Установка доверительных отношений

Чтобы при заходе на наши веб-сервисы был HTTPS, нужно чтобы HQ-CLI доверял нашему CA. Для этого нужно установить корневой сертификат CA ca.crt в качестве доверенного на HQ-CLI.

С HQ-SRV скопируйте ca.crt на HQ-CLI:

```
scp ca.crt user@192.168.10.66:/home/user
```

Сору

На самом HQ-CLI переместите сертификат в директорию /etc/pki/ca-trust/source/anchors:

```
mv /home/user/ca.crt /etc/pki/ca-trust/source/anchors/
```

Сору

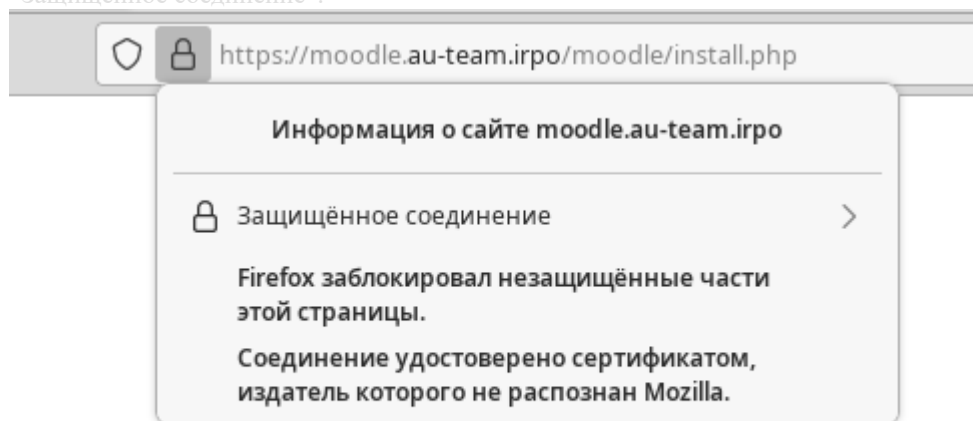
А потом выполните команду:

```
update-ca-trust
```

Сору

Проверка успешности выполнения задания

Полностью перезапустите браузер и зайдите на веб-ресурсы по доменным именам. У вас должен висеть замочек “Защищенное соединение”.



Если возникли проблемы проверьте доверяет ли HQ-CLI вашему CA:

```
trust list | grep 'AU-Team CA'
```

Сору

Так же можете проверить сам сертификат:

```
openssl x509 -in web.crt -text -noout | less
```

Сору

Там должны быть расширения (X509v3 extensions):

- X509v3 Subject Alternative Name
- X509v3 Extended Key Usage
- X509v3 Key Usage

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

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

- Обеспечьте доступность по URL - <https://mon.au-team.irpo>
 - Мониторить нужно устройства HQ-RTR, HQ-SRV, BR-RTR и BR-SRV
 - В мониторинге должны визуально отображаться нагрузка на ЦП, объем занятой ОП и основного накопителя
 - Логин и пароль для службы мониторинга admin P@ssw0rd
 - Выбор программного обеспечения, основание выбора и основные параметры с указанием порта, на котором работает мониторинг, отметьте в отчёте
- В качестве системы мониторинга мы будем использовать zabbix, но так как на HQ-SRV у нас размещен еще и moodle, который так же написан на php и использует Apache, то возможны конфликты зависимостей. Поэтому мы будем развертывать Zabbix с помощью Docker Compose.

Установка контейнерной инфраструктуры

```
apt-get install docker-engine docker-compose-v2 -y
systemctl enable --now docker
```

Copy

Проверьте имеющиеся образы:

```
docker images
```

Copy

Должны быть образы для реализации Zabbix с базами данных mariadb и postgres.

Создание Docker Compose

В домашней директории создайте файл zabbix.yml и заполните его:

```
services:
  zabbix-postgres:
    container_name: zabbix-postgres
    image: postgres
    volumes:
      - postgres-zabbix:/var/lib/postgresql/data
    environment:
      POSTGRES_DB: zabbix
      POSTGRES_USER: zabbix
      POSTGRES_PASSWORD: zabbix
    restart: unless-stopped

  zabbix-server:
    container_name: zabbix-server
    image: zabbix/zabbix-server-pgsql
    environment:
      DB_SERVER_HOST: zabbix-postgres
      DB_SERVER_PORT: 5432
      POSTGRES_DB: zabbix
      POSTGRES_USER: zabbix
      POSTGRES_PASSWORD: zabbix
    ports:
      - 10051:10051
    restart: unless-stopped
    depends_on:
      - zabbix-postgres

  zabbix-web:
    container_name: zabbix-web
    image: zabbix/zabbix-web-nginx-pgsql
    environment:
      DB_SERVER_HOST: zabbix-postgres
      DB_SERVER_PORT: 5432
      POSTGRES_DB: zabbix
      POSTGRES_USER: zabbix
      POSTGRES_PASSWORD: zabbix
      ZBX_SERVER_HOST: zabbix-server
      ZBX_SERVER_PORT: 10051
      PHP_TZ: Europe/Yekaterinburg
    ports:
```

```
- 8080:8080
restart: unless-stopped
depends_on:
  - zabbix-postgres
```

volumes:
postgres-zabbix:

Copy

Запустите стек контейнеров:

```
docker compose -f zabbix.yml up -d
```

Copy

Убедитесь что все контейнеры запустились и исправны:

```
docker ps -a
```

Copy

Веб-конфигурирование

На HQ-SRV создайте DNS запись для сервиса:

```
mon IN A 192.168.10.2
```

Copy

Перейдите на веб-интерфейс по адресу <https://mon.au-team.irpo:8080>

Зайдите под стандартной учеткой:

- логин: Admin
- пароль: zabbix

Первым делом нам нужно сменить пароль на P@ssw0rd, но в zabbix по умолчанию стоит проверка пароля на сложность, ее надо отключить.

Перейдите по пути **Users** → **Authentication** и снимите галочку на пункте **Avoid easy-to-guess password**, после чего можно менять пароль **User settings** → **Profile** и смените пароль.

Настройка zabbix-агентов

На Linux:

На узлах HQ-SRV и BR-SRV откройте /etc/zabbix/zabbix_agentd.conf и измените в нем параметры:

```
Server=0.0.0.0/0
ServerActive=192.168.10.2
```

Copy

После чего запустите агент:

```
systemctl restart zabbix_agentd
```

Copy

На Eltex (здесь пример на HQ-RTR):

```
hq-rtr(config)# zabbix-agent
hq-rtr(config-zabbix-agent)# server 192.168.10.2
hq-rtr(config-zabbix-agent)# active-server 192.168.10.2
hq-rtr(config-zabbix-agent)# port 10050
hq-rtr(config-zabbix-agent)# hostname hq-rtr.au-team.irpo
hq-rtr(config-zabbix-agent)# enable
hq-rtr(config-zabbix-agent)# end

hq-rtr# commit
hq-rtr# confirm
```

Copy

Подключение zabbix-агентов к серверу

Вернитесь к веб-интерфейсу <https://mon.au-team.irpo:8080>, и перейдите по пути **Monitorin** → **Hosts**, далее нажмите **Create host**

Zabbix server (сам HQ-SRV, который мы мониторим из контейнера, он уже создан его нужно только изменить):

Host

Host IPMI Tags Macros Inventory Encryption Value mapping

* Host name Zabbix server

Visible name Zabbix server

Templates	Name	Actions
	Linux by Zabbix agent	Unlink Unlink and clear
	Zabbix server health	Unlink Unlink and clear
	type here to search	

Select

* Host groups Zabbix servers X type here to search

Select

Interfaces	Type	IP address	DNS name	Connect to	Port	Default
Agent		192.168.10.2		IP DNS	10050	<input checked="" type="radio"/> Remove

[Add](#)

Description

Monitored by Server Proxy Proxy group

Enabled ☒

Update Clone

BR-SRV:

Host

Host IPMI Tags Macros Inventory Encryption Value mapping

* Host name

Visible name

Templates

Name

Linux by Zabbix agent

Actions

Unlink Unlink and clear

Select

* Host groups

Linux servers X

Select

Interfaces	Type	IP address	DNS name	Connect to	Port	Default
Agent		<input type="text" value="192.168.20.2"/>	<input type="text"/>	<div>IP DNS</div>	<input type="text" value="10050"/>	<div><input checked="" type="radio"/> Remove</div>

[Add](#)

Description

Monitored by

Server Proxy Proxy group

Enabled ☒

Update Clone

IP: 192.168.20.2

Host

Host IPMI Tags Macros Inventory Encryption Value mapping

* Host name

Visible name

Templates

Name

Linux by Zabbix agent

type here to search

Actions

[Unlink](#) [Unlink and clear](#)

Select

* Host groups

Linux servers X

type here to search

Select

Interfaces	Type	IP address	DNS name	Connect to	Port	Default
Agent		<input type="text" value="192.168.10.1"/>	<input type="text"/>	<div><div>IP</div><div>DNS</div></div>	<input type="text" value="10050"/>	<div><div><input checked="" type="radio"/></div><div>Remove</div></div>

[Add](#)

Description

Monitored by

Server

Proxy

Proxy group

Enabled ☒

Update

Clone

BR-RTR:

Host

Host IPMI Tags Macros Inventory Encryption Value mapping

* Host name

Visible name

Templates

Name

Linux by Zabbix agent

Actions

Unlink Unlink and clear

Select

* Host groups

Linux servers X

Select

Interfaces	Type	IP address	DNS name	Connect to	Port	Default
Agent		<input type="text" value="192.168.20.2"/>	<input type="text"/>	<div>IP DNS</div>	<input type="text" value="10050"/>	<div><input checked="" type="radio"/> Remove</div>

[Add](#)

Description

Monitored by

Server Proxy Proxy group

Enabled ☒

Update

Clone

У вас должно быть так:

Save as

Apply

Reset

Name ▲	Interface	Availability	Tags	
br-rtr.au-team.irpo	192.168.20.1:10050	ZBX	class: os target: linux	E
br-srv.au-team.irpo	192.168.20.2:10050	ZBX	class: os target: linux	E
hq-cli.au-team.irpo	192.168.10.66:10050	ZBX	class: os target: linux	E
hq-rtr.au-team.irpo	192.168.10.1:10050	ZBX	class: os target: linux	E
Zabbix server	192.168.10.2:10050	ZBX	class: os class: software target: linux ...	E

Настройте принт-сервер cups на сервере HQ-SRV

Опубликуйте виртуальный pdf-принтер

Скачайте CUPS на HQ-SRV (пакет cups-pdf на 2 стенде пока не предустановлен, так что качайте через внешку):

```
apt-get install cups cups-pdf
```

Сору

Откройте /etc/cups/cupsd.conf и везде где есть строка Location вставьте **Allow all**, вот пример:

```
<Location />
  Allow all
</Location>
```

Сору

Далее измените в этом же файле строку **Listen localhost** на:

```
Listen hq-srv.au-team.irpo:631
```

Сору

Затем включите CUPS:

```
systemctl enable --now cups
```

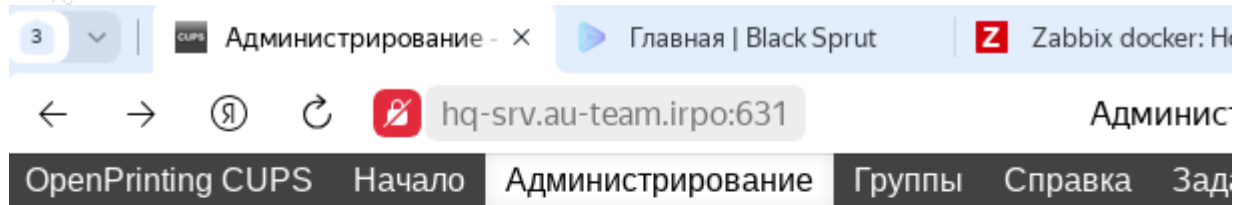
Сору

Веб-конфигурирование

Зайдите с HQ-CLI на <https://hq-srv.au-team.irpo:631>

Зайдите на вкладку Администрирование, залогиньтесь под root P@ssw0rd.

Следуйте шагам:



Администрирование

Принтеры

Добавить принтер

Найти новый принтер

Управление принтерами

Группы

Добавить группу

Управление группами

Задания

Управление заданиями

Добавить принтер

Добавление принтера

Установленные принтеры: ☒ CUPS-PDF (Virtual PDF Printer)

Найденные сетевые принтеры:

- Другие сетевые принтеры:
- ☐ Backend Error Handler
 - ☐ AppSocket/HP JetDirect
 - ☐ Протокол интернет-печати (ipp)
 - ☐ Протокол интернет-печати (http)
 - ☐ Хост или принтер LPD/LPR
 - ☐ Протокол интернет-печати (ipp)
 - ☐ Протокол интернет-печати (https)

Продолжить

Добавить принтер

Добавление принтера

Название:

(может содержать любые символы, кроме "/", "# и пробела)

Описание:

(расширенное описание, например, "HP LaserJet с дуплексной печатью")

Расположение:

(месторасположение принтера, например, "Кабинет 55")

Подключение: cups-pdf/

Совместный доступ: ☒ Разрешить совместный доступ к этому принтеру

Добавить принтер

Добавление принтера

Название: Virtual_PDF_Printer

Описание: Virtual PDF Printer

Расположение:

Подключение: cups-pdf/

Совместный доступ: Разрешить совместный доступ к этому принтеру

Создать:

DYMO

Epson

Fuji Xerox

Generic

HP

Index

Intellitech

Oki

Raw

Ricoh

Продолжить

или использовать файл PPD: Выберите файл Файл не выбран

Добавить принтер

Добавить принтер

Добавление принтера

Название: Virtual_PDF_Printer

Описание: Virtual PDF Printer

Расположение:

Подключение: cups-pdf/

Совместный доступ: Разрешить совместный доступ к этому принтеру

Создать: HP

Модель:

HP Color LaserJet CM3530 MFP PDF (en)

HP Color LaserJet Series PCL 6 CUPS (en)

HP DesignJet 600 pcl, 1.0 (en)

HP DesignJet 750c pcl, 1.0 (en)

HP DesignJet 1050c pcl, 1.0 (en)

HP DesignJet 4000 pcl, 1.0 (en)

HP DesignJet T790 pcl, 1.0 (en)

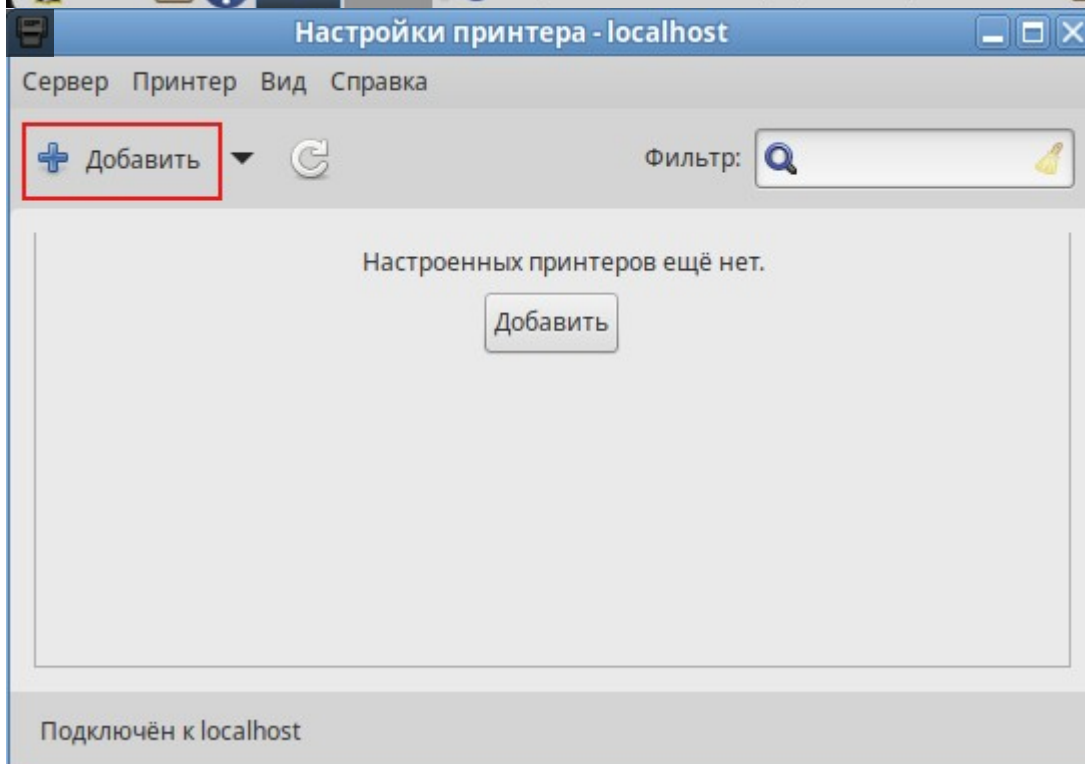
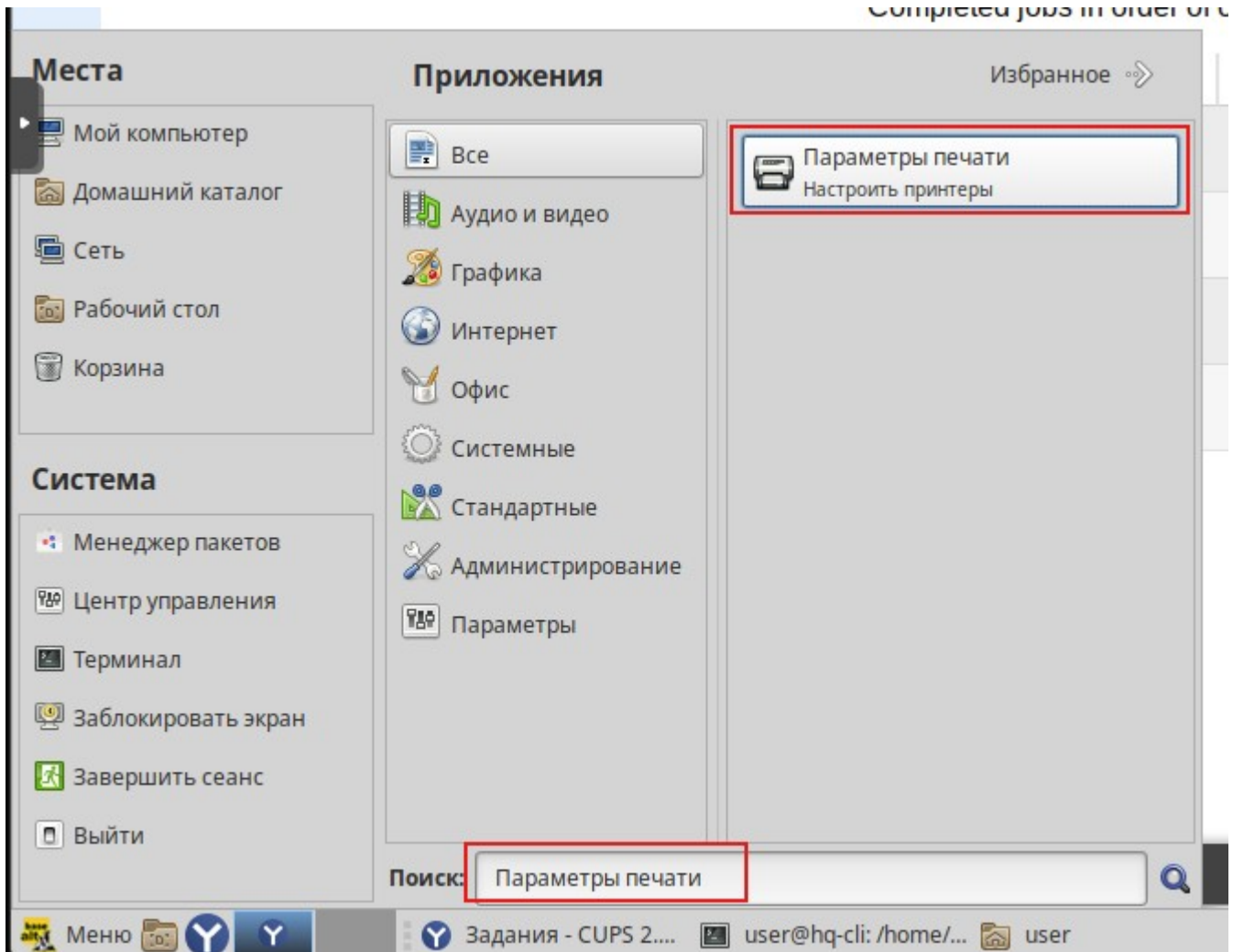
HP DesignJet T1100 pcl, 1.0 (en)

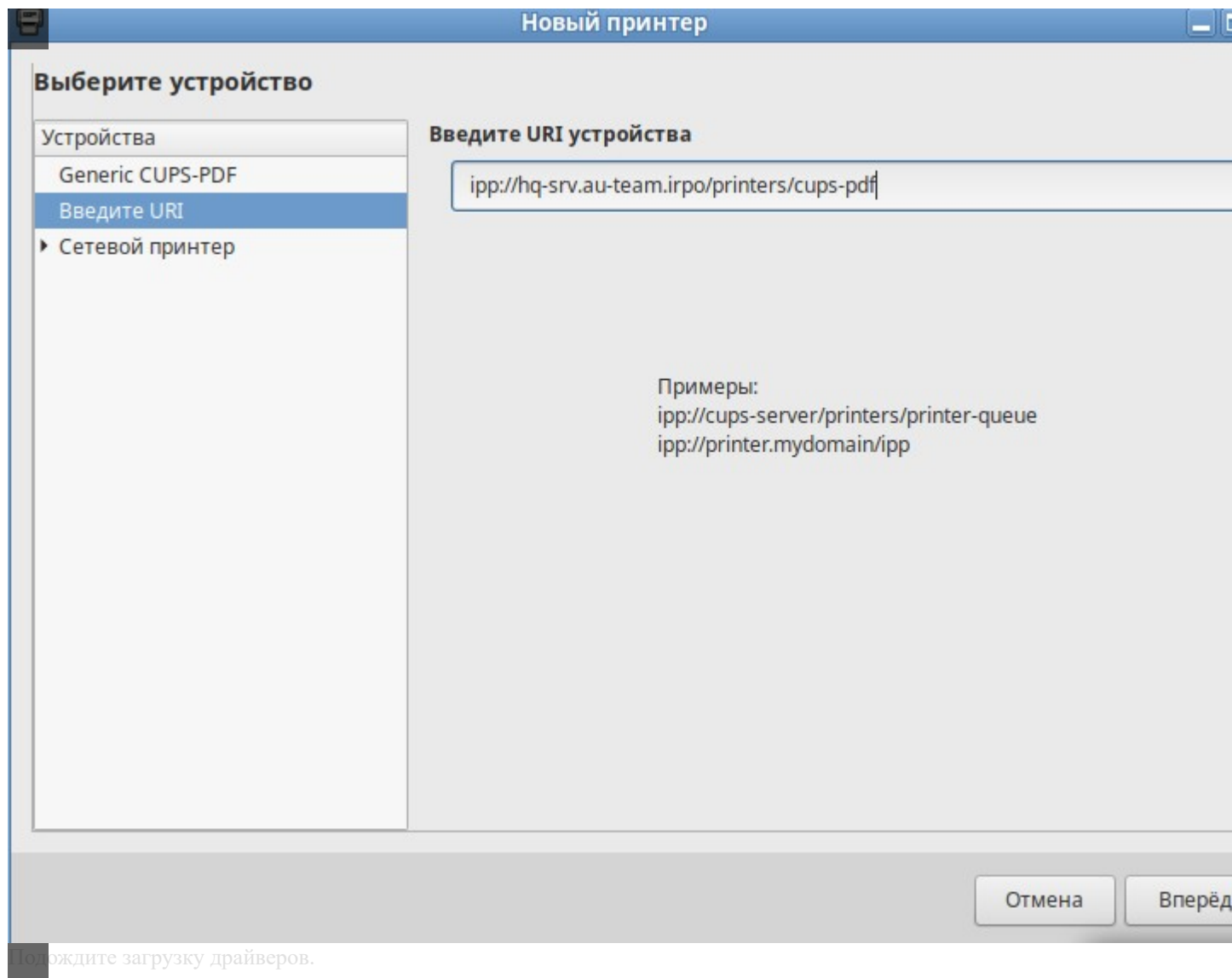
HP DeskJet Series (en)

HP LaserJet Series PCL 4/5 (en)

или использовать файл PPD: Файл не выбран

На клиенте HQ-CLI подключите виртуальный принтер как принтер по умолчанию
Нужно добавить принтер в систему в качестве принтера по умолчанию:







Новый принтер



Опишите принтер

Имя принтера

Краткое имя этого принтера, например «laserjet»

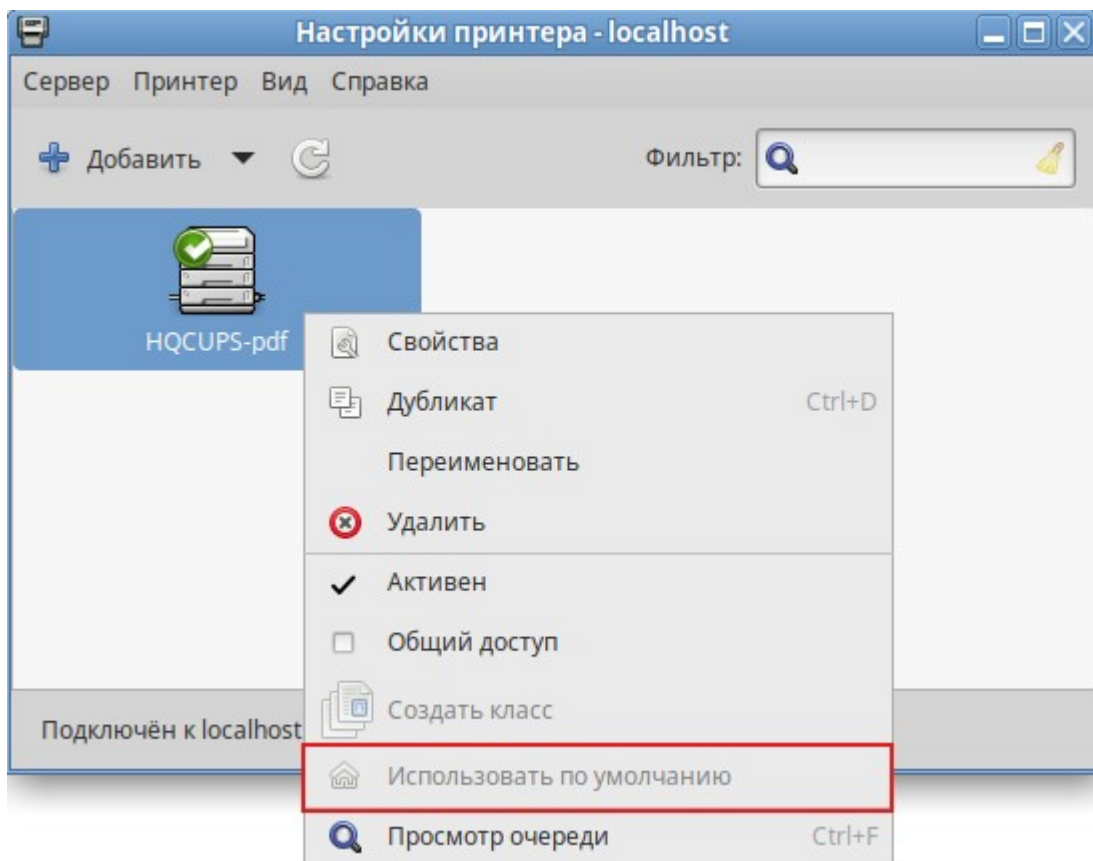
Описание (необязательно)

Удобное для восприятия описание, например «HP LaserJet с устройством двусторонней печати»

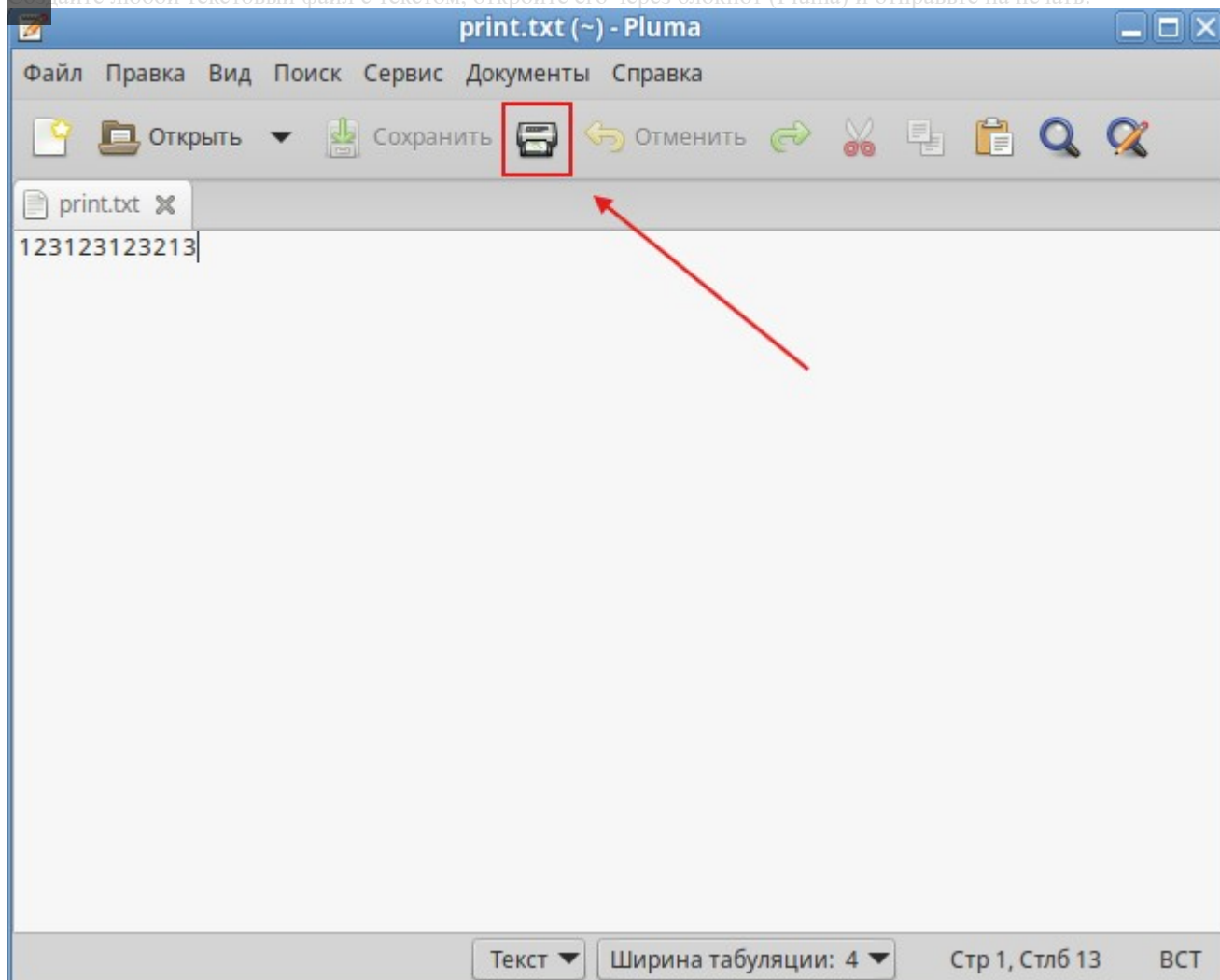
Расположение (необязательно)

Описание места расположения принтера, например «Lab 1»

У вас в системе будет 2 принтера, по этому нужно выбрать принтер по умолчанию:



Создайте любой текстовый файл с текстом, откройте его через блокнот (Pluma) и отправьте на печать:



Печать



Общие

Параметры страницы

Текстовый редактор

Задание

Качество изображения

Принтер	Расположение	Состояние
 Печатать в файл		
 HQCUPS-pdf		

Диапазон

☒ Все страницы

☐ Текущую страницу

☐ Страницы:

Копии

Копий:

☐ Упорядочить

☐ Наоборот

Образец

Далее проверьте сработала ли печать, чтобы это сделать перейдите на веб-морде CUPS на вкладку Задания и нажмите “Показать все задания”:

Задания

Поиск задания:

Показать активные задания

Показать все задания

Completed jobs in order of completion or cancellation

Номер	Название	Пользователь	Размер	Состояние
Cups-PDF-5	Неизвестное	Приостановлено пользователем	16k	1
Cups-PDF-4	Неизвестное	Приостановлено пользователем	16k	1
Cups-PDF-3	Неизвестное	Приостановлено пользователем	1k	1
Cups-PDF-2	Неизвестное	Приостановлено пользователем	1k	1
Cups-PDF-1	Неизвестное	Приостановлено пользователем	1k	1

Если у вас появились задания, то все все хорошо

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

Настройте защищенный туннель между HQ-RTR и BR-RTR

GRE туннель у нас уже [настроен](#), но не обеспечивает защищенное соединение, а IPSec туннель не способен передавать мультикастовый трафик – нужно использовать GRE over IPSec туннель (GRE находится внутри IPSec туннеля). Только этот вид туннеля обеспечит защищенное соединение и возможность передачи мультикастового трафика для протоколов динамической маршрутизации.

Так же при использовании этого вида туннеля желательно перевести IPSec в транспортный режим, это сэкономит 20 байтов в пакете, но для упрощения конфигурации этого здесь не будет.

HQ-RTR:

```
# Создание профиля определяющего конфигурацию подключения для служебного туннеля
hq-rtr(config)# security ike proposal ike_prop1
hq-rtr(config-ike-proposal)# authentication algorithm md5
hq-rtr(config-ike-proposal)# encryption algorithm aes128
hq-rtr(config-ike-proposal)# dh-group 2
hq-rtr(config-ike-proposal)# exit

# Создание политики определяющей профиль и пароль для служебного туннеля
hq-rtr(config)# security ike policy ike_pol1
hq-rtr(config-ike-policy)# pre-shared-key ascii-text cisco_forever
hq-rtr(config-ike-policy)# proposal ike_prop1
hq-rtr(config-ike-policy)# exit

# Создание шлюза для протокола IKE
hq-rtr(config)# security ike gateway ike_gw1
hq-rtr(config-ike-gw)# ike-policy ike_pol1
hq-rtr(config-ike-gw)# local address 172.16.4.2
hq-rtr(config-ike-gw)# local network 172.16.4.2/32 protocol gre
hq-rtr(config-ike-gw)# remote address 172.16.5.2
hq-rtr(config-ike-gw)# remote network 172.16.5.2/32 protocol gre
hq-rtr(config-ike-gw)# mode policy-based
hq-rtr(config-ike-gw)# exit

# Создание профиля определяющего конфигурацию подключения для ipsec туннеля
hq-rtr(config)# security ipsec proposal ipsec_prop1
hq-rtr(config-ipsec-proposal)# authentication algorithm md5
hq-rtr(config-ipsec-proposal)# encryption algorithm aes128
hq-rtr(config-ipsec-proposal)# pfs dh-group 2
hq-rtr(config-ipsec-proposal)# exit

# Создание политики для ipsec туннеля
hq-rtr(config)# security ipsec policy ipsec_pol1
hq-rtr(config-ipsec-policy)# proposal ipsec_prop1
hq-rtr(config-ipsec-policy)# exit

# Создание самого IPSec туннеля
hq-rtr(config)# security ipsec vpn ipsec1
hq-rtr(config-ipsec-vpn)# ike establish-tunnel immediate
hq-rtr(config-ipsec-vpn)# ike gateway ike_gw1
hq-rtr(config-ipsec-vpn)# ike ipsec-policy ipsec_pol1
hq-rtr(config-ipsec-vpn)# enable
hq-rtr(config-ipsec-vpn)# exit
```

Copy

BR-RTR:

```
br-rtr(config)# security ike proposal ike_prop1
br-rtr(config-ike-proposal)# authentication algorithm md5
br-rtr(config-ike-proposal)# encryption algorithm aes128
br-rtr(config-ike-proposal)# dh-group 2
br-rtr(config-ike-proposal)# exit

br-rtr(config)# security ike policy ike_pol1
br-rtr(config-ike-policy)# pre-shared-key ascii-text cisco_forever
br-rtr(config-ike-policy)# proposal ike_prop1
```

```
br-rtr(config-ike-policy)# exit

br-rtr(config)# security ike gateway ike_gw1
br-rtr(config-ike-gw)# ike-policy ike_pol1
br-rtr(config-ike-gw)# local address 172.16.5.2
br-rtr(config-ike-gw)# local network 172.16.5.2/32 protocol gre
br-rtr(config-ike-gw)# remote address 172.16.4.2
br-rtr(config-ike-gw)# remote network 172.16.4.2/32 protocol gre
br-rtr(config-ike-gw)# mode policy-based
br-rtr(config-ike-gw)# exit

br-rtr(config)# security ipsec proposal ipsec_prop1
br-rtr(config-ipsec-proposal)# authentication algorithm md5
br-rtr(config-ipsec-proposal)# encryption algorithm aes128
br-rtr(config-ipsec-proposal)# pfs dh-group 2
br-rtr(config-ipsec-proposal)# exit

br-rtr(config)# security ipsec policy ipsec_pol1
br-rtr(config-ipsec-policy)# proposal ipsec_prop1
br-rtr(config-ipsec-policy)# exit

br-rtr(config)# security ipsec vpn ipsec1
br-rtr(config-ipsec-vpn)# ike establish-tunnel immediate
br-rtr(config-ipsec-vpn)# ike gateway ike_gw1
br-rtr(config-ipsec-vpn)# ike ipsec-policy ipsec_pol1
br-rtr(config-ipsec-vpn)# enable
br-rtr(config-ipsec-vpn)# exit
```

Copy

Проверка работоспособности туннеля:

```
ESR# show tunnels counters gre
ESR# show security ipsec vpn status
```

Настройка GRE-туннеля с шифрованием на Ubuntu с использованием Netplan

```
sudo apt update
```

```
sudo apt install strongswan strongswan-swanctl
```

2. Настройка StrongSwan для шифрования (IPSec)

На **обеих** машинах создайте конфигурацию IPSec:

```
sudo nano /etc/ipsec.conf
```

Добавьте конфигурацию (адаптируйте под ваши IP-адреса):

```
config setup
```

```
    charondebug="ike 2, knl 2, cfg 2, net 2, esp 2, dmh 2, mgr 2"
```

```
conn %default
```

```
    keyexchange=ikev2
```

```
    ike=aes256-sha256-modp2048!
```

```
    esp=aes256-sha256-modp2048!
```

```
    keyingtries=0
```

```
    ikelifetime=1h
```

```
    lifetime=8h
```

```
conn gre-tunnel
```

```
    left=<LOCAL_IP>
```

```
    leftsubnet=<LOCAL_NETWORK>
```

```
    right=<REMOTE_IP>
```

```
    rightsubnet=<REMOTE_NETWORK>
```

```
    auto=start
```

```
    type=transport
```

```
    authby=secret
```

```
    leftprotoport=gre
```

```
    rightprotoport=gre
```

Создайте файл с общим ключом:

```
sudo nano /etc/ipsec.secrets
```

Добавьте строку (используйте свой ключ):

```
<LOCAL_IP> <REMOTE_IP> : PSK "your_strong_pre_shared_key_here"
```

3. Настройка Netplan для GRE-туннеля

Создайте/отредактируйте конфигурационный файл Netplan:

```
sudo nano /etc/netplan/99-gre-tunnel.yaml
```

Пример конфигурации (для сервера):

```
network:
```

```
  version: 2
```

```
  renderer: networkd
```

```
  tunnels:
```

```
    gre1:
```

```
      mode: gre
```

```
      remote: <REMOTE_IP>
```

```
      local: <LOCAL_IP>
```

```
      addresses: [10.0.0.1/30]
```

```
      mtu: 1400
```

```
      parameters:
```

```
        ikey: 12345
```

```
        okey: 12345
```

Для клиента:

```
network:
```

```
  version: 2
```

```
  renderer: networkd
```

```
  tunnels:
```

```
    gre1:
```



```
mode: gre
remote: <SERVER_IP>
local: <LOCAL_IP>
addresses: [10.0.0.2/30]
mtu: 1400
parameters:
  ikey: 12345
  okey: 12345
```

4. Применение изменений

1.Перезапустите StrongSwan:

```
sudo systemctl restart strongswan
```

2.Примените конфигурацию Netplan:

```
sudo netplan apply
```

5. Проверка работы

Проверьте состояние туннеля:

```
ip tunnel show
```

```
ip addr show gre1
```

Проверьте IPSec соединение:

```
sudo ipsec status
```

Проверьте связность:

```
ping 10.0.0.2 # с сервера на клиент
```

Дополнительные настройки

Включение маршрутизации (если нужно)

На сервере:

```
sudo nano /etc/sysctl.conf
```

Раскомментируйте строку:

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

Примените изменения:

```
sudo sysctl -p
```

Firewall правила (UFW)

```
sudo ufw allow 500/udp # для IKE
```

```
sudo ufw allow 4500/udp # для NAT-T
```

```
sudo ufw allow proto gre
```

Автозапуск сервисов

```
sudo systemctl enable strongswan
```

Возможные проблемы и решения

1. Туннель не поднимается:

- Проверьте journalctl -xe и sudo ipsec status
- Убедитесь, что порты 500/udp и 4500/udp открыты на фаерволе

2. Нет связи через туннель:

- Проверьте MTU (может потребоваться уменьшить до 1400 или меньше)
- Убедитесь, что маршруты настроены правильно

3. Проблемы с шифрованием:

- Проверьте файлы /etc/ipsec.conf и /etc/ipsec.secrets на обеих машинах
- Убедитесь, что временные зоны синхронизированы (sudo apt install ntp)