# Методические указания по выполнению задания демонстрационного экзамена по модулю “A” компетенции №39

## 

## Схема сети

## 

## 

## 

## 

## Материалы курса

### Видео

Плейлист с выполнением основной части задания (пополняется по мере):

[**Выполнение задание на демо экзамен**](https://www.youtube.com/playlist?list=PLLkEABlJyUrcBNRqZdyhTaehIQCDq_1AS)

### Дополнительные материалы для саморазвития:

Файлы и директории

* [Основные операции с файлами и директориями](https://www.youtube.com/watch?v=mYyamfWx7IU&list=PLmxB7JSpraiep6kr802UDqiAIU-76nGfc&index=15)
* [Монтирование файловых систем](https://www.youtube.com/watch?v=wPfN3xT-vFQ&list=PLmxB7JSpraiep6kr802UDqiAIU-76nGfc&index=23)
* [Текстовый редактор Vi](https://www.youtube.com/watch?v=ocln2_AJ4vo&list=PLmxB7JSpraiep6kr802UDqiAIU-76nGfc&index=20)

Установка/удаление программ

* [Управление пакетами в Debian](https://www.youtube.com/watch?v=_zgRUYLJa4w&list=PLmxB7JSpraiep6kr802UDqiAIU-76nGfc&index=11)
* [Управление пакетами в RedHat](https://www.youtube.com/watch?v=AWc_9Cv6xnQ&list=PLmxB7JSpraiep6kr802UDqiAIU-76nGfc&index=12)

Сеть

* [Конфигурационные файлы сети](https://www.youtube.com/watch?v=rZsU6Tkgn4g&list=PLmxB7JSpraieH3aScuPs7h77cSEQvPi2A&index=46)

Службы

* [systemd](https://www.youtube.com/watch?v=0IQHR6g4zLA&list=PLmxB7JSpraiep6kr802UDqiAIU-76nGfc&index=6)

DHCP и DNS

* [Настройка DHCP сервера • DHCP Relay • IPv4 и IPv6](https://www.youtube.com/watch?v=S4UR6epGonM)
* [Настройка DNS сервера на BIND9 • IPv4 и IPv6 • Split DNS](https://www.youtube.com/watch?v=UKBDT56SrF0&t=2394s)
* [DNS+DHCP. Часть первая - настройка сети](https://www.youtube.com/watch?v=OzD70xRsltc)
* [DNS+DHCP. Часть вторая - установка сервера bind](https://www.youtube.com/watch?v=Asc47EUrCgg)
* [DNS+DHCP. Часть третья - настройка сервера bind](https://www.youtube.com/watch?v=YdmaZc2L8vE)
* [DNS+DHCP. Часть четвертая - настройка сервера dhcp](https://www.youtube.com/watch?v=3iSb-GR9Ylg)
* [DNS+DHCP. Часть пятая: связка DNS и DHCP](https://www.youtube.com/watch?v=9iz_jPeO9rs)

### Command reference

|  |  |
| --- | --- |
| Файлы и директории | |
| Пути:   * Полный путь - указывается начиная с корня /dir1/subdir1/subdir2/ * Относительный путь - точкой отсчета является - текущая директория Пример: Если мы находимся в директории “/root”, то указание пути “**./**dir1/dir2/” развернется в “/root/dir1/dir2/”.   Подстановки (globling/wildcards):   * \* - любое количество любых символов в пути * ? - один любой символ в пути | |
| **pwd** | Узнать текущую директорию |
| **cd** /change/current/directory | Изменить текущую директорию |
| **mkdir** /dir/i/want/to/create | Создать директорию “create” в директории “/dir/i/want/to”.  Родительская директория должна существовать. |
| **mv** /dir1/file1 /dir2/ | Переместить “file1” из директории “/dir1” в директорию “/dir2”. |
| **cp** /dir1/file1 /dir2/ | Копировать “file1” из директории “/dir1” в директорию “/dir2”. |
| **mount** /dev/device /mnt/directory | Смонтировать файловую систему с устройства “/dev/device” в директорию “/mnt/directory”. |
| **less** /text/file | less - пейджер, который постранично выводит текст из файла. Возможно двигаться вниз и вверх. Также есть возможность поиска, для этого нужно ввести “**/**” и набрать текст, нажать Enter.  Для выхода нужно нажать “**q”**. |
| **man** mkdir | man - система встроенной документации.  Будет активно использоваться во время выполнения задания. Навигация и поиск такие же как в less. |
| Текстовые редакторы | |
| **vi** /etc/file.txt | Великий и ужасный для новичков.  Имеет три режима:   1. Командный режим (дефолт) 2. Режим ввода текста 3. Режим ввода полнотекстовых команд.   Чтобы переключиться в режим ввода текста, нужно нажать “**i**” (от слова insert),в данном режиме могут работать не все клавиши.  Чтобы вернуться в командный режим, нужно нажать “**Esc**”.  Из командного режима, можно переместиться в режим ввода команд, нажав “**:**”.  Чтобы сохраниться, нужно ввести “**:wq**”. |
| **nano** /etc/file.txt | Более простой текстовый редактор, но установлен не на всех системах.  Никто не запрещает вам его установить, если он более удобен. Но для того чтобы его установить, нужно будет хотя бы раз воспользоваться **vi**. |
| Установка пакетов | |
| **apt-cdrom add** | Добавить информацию о дисках в базу системы. |
| **apt install -y** pkg1 pkg2 … pkgN | **apt** - пакетный менеджер Debian  **install** - установить  **-y** - Отвечать на все вопросы “Да”  далее через пробел перечисляются пакеты |
| **yum install -y** pkg1 pkg2 … pkgN | **yum** - пакетный менеджер CentOS  **install** - установить  **-y** - Отвечать на все вопросы “Да”  далее через пробел перечисляются пакеты |
| Сеть | |
| **ip a** | показать ip-адреса на интерфейсах |
| **nmtui** | Утилита, которая имеет интерфейс и позволяет настраивать адресацию, создавать виртуальные интерфейсы, туннели и тд.  Идёт в пакете с network-manager, который не установлен только на Debian без графического интерфейса. Поэтому, для того чтобы унифицировать настройку сети, мы его поставим. |
| **sysctl -a** | Вывести на экран параметры ядра |
| **sysctl -p** | Применить параметры из /etc/sysctl.conf |
| **Управление сервисами** | |
| **systemctl stop** firewalld.service | Остановить сервис “firewalld” |
| **systemctl disable** firewalld.service | Убрать сервис “firewalld” из автозагрузки |
| **systemctl start** frr.service | Запустить сервис frr“ |
| **systemctl restart** frr.service | Перезапустить сервис “frr” |
| **systemctl enable** frr.service | Добавить сервис “frr” в автозагрузку |
| **Netfilter and Firewalld** | |
| **firewalld-cmd --add-interface=**ens160 **--zone=**external **--permannet** | Добавить интерфейс в зону “external”. |
| **firewalld-cmd --add-service=**gre **--zone=**external **--permannet** | Разрешить GRE в зоне “external”. |
| iptable -t nat -A POSTROUTING -o ens160 -j MASQUERADE | Включить source NAT на интерфейсе ens160 |

## 

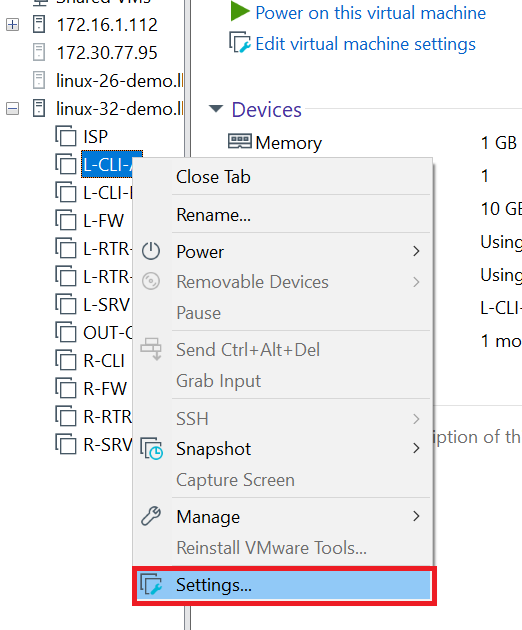
## Добавление устройств и запуск VM

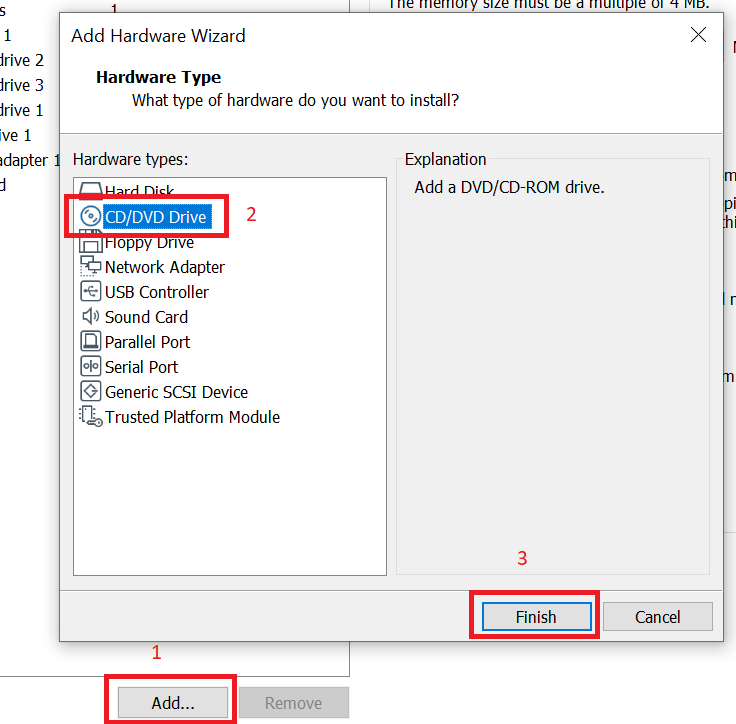
### Debian

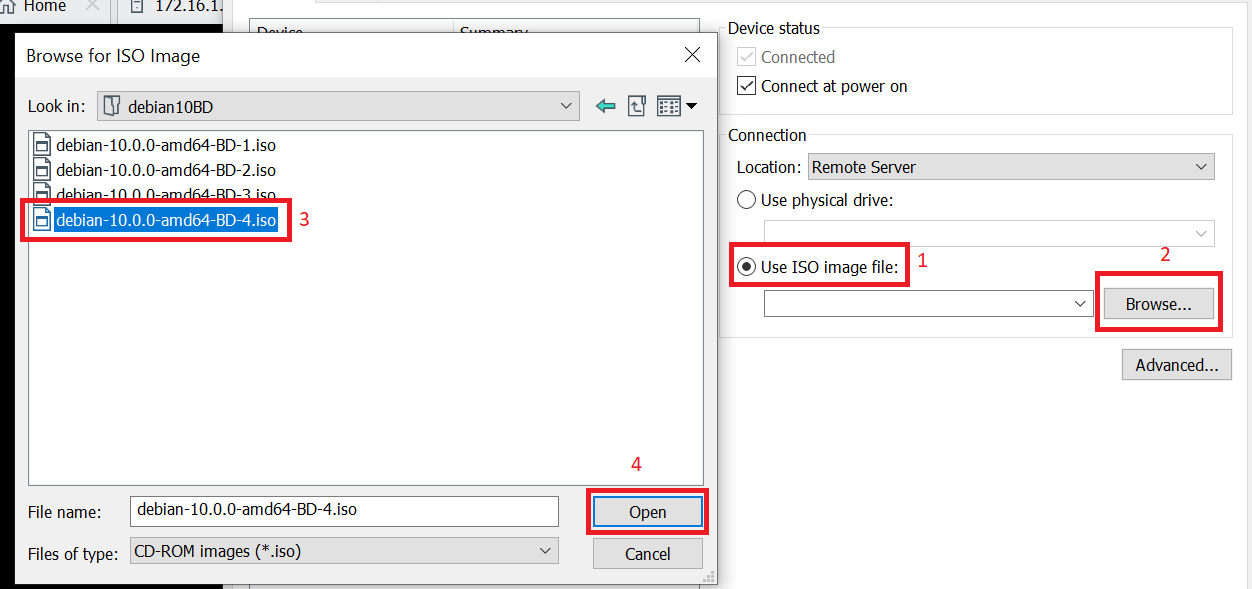
Напоминаю, что Debian установлен на: L-CLI-A, L-CLI-B, L-SRV, L-RTR-A, L-RTR-B, L-FW.

Всё программное обеспечение, которое вы будете устанавливать по ходу выполнения задания находится на дисках “debian-10.0.0-amd64-BD-1.iso”, “debian-10.0.0-amd64-BD-2.iso”, “debian-10.0.0-amd64-BD-3.iso” и “debian-10.0.0-amd64-BD-4.iso”.

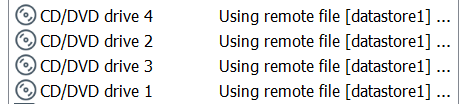
Система была установлена с диска “debian-10.0.0-amd64-BD-1.iso”, по этому изначально подключен только он. Вам необходимо добавить остальные три.







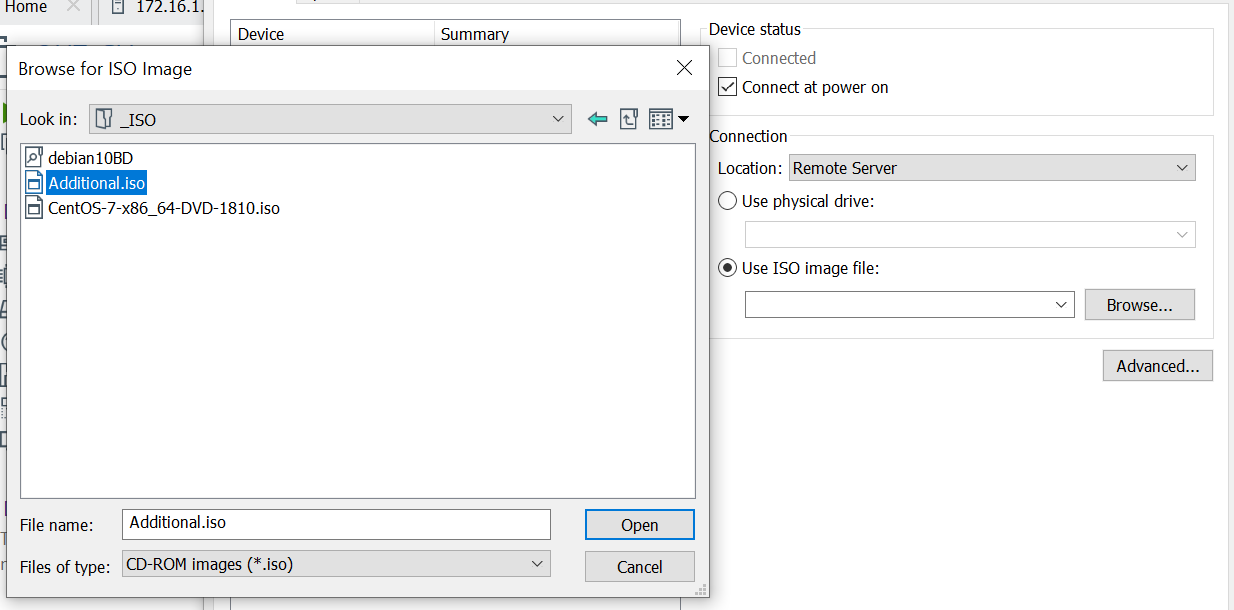
После подключения всех дисков, конфигурация виртуальной машины должна выглядеть следующим образом:



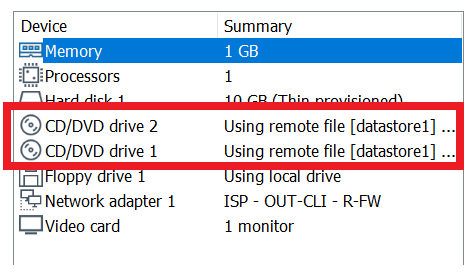
### CentOS

Напоминаю, что CentOS установлен на: OUT-CLI, R-FW, R-SRV, R-RTR, R-CLI.

Софт, который необходимо установить, на VM с CentOS находится на дисках “CentOS-7-x86\_64-Everything-1810.iso” и “Additional.iso”. Поэтому нам необходимо добавить только один диск.

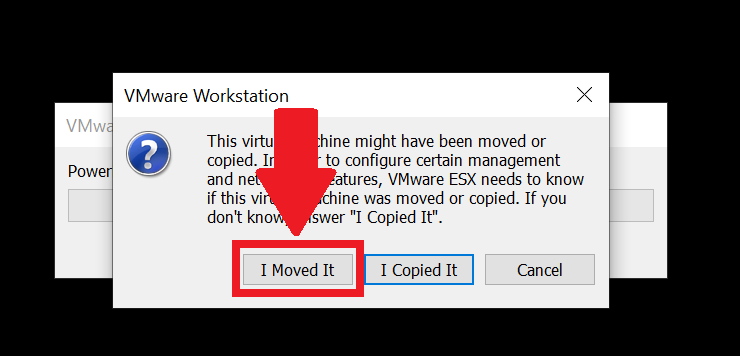


В результате получаем следующую конфигурацию



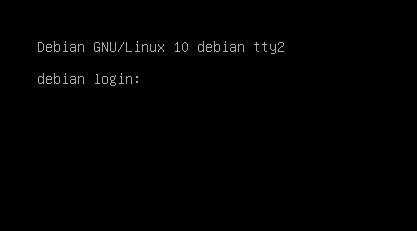
### Включение виртуальных машин

После того как вы добавили дополнительное оборудование, нужно включить виртуальные машины. После нажатия кнопки “Power On” вас попросят выбрать была эта VM клонирована или перемещена. Вам нужно ответить, что вы её переместили.



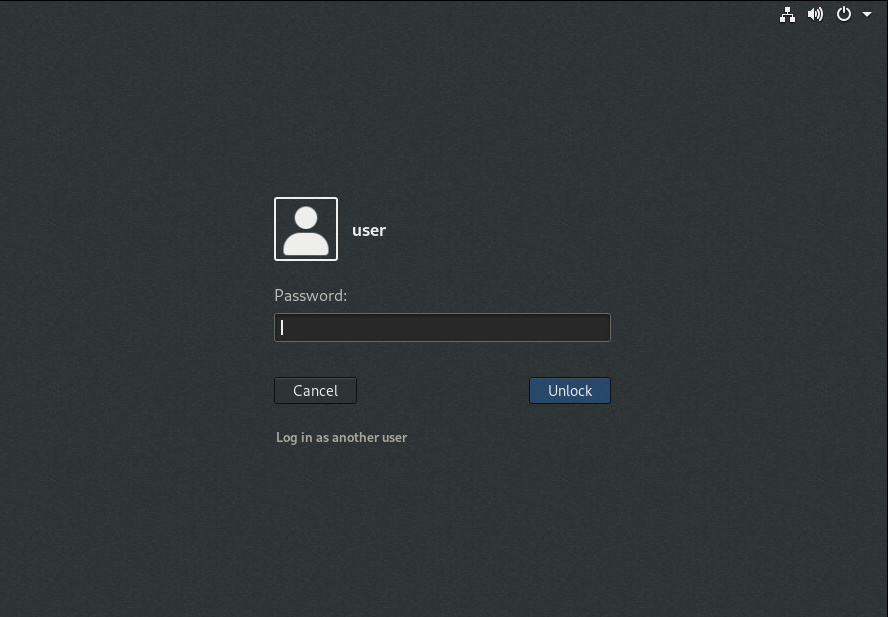
## Вход в систему

Для систем, которые установлены без графического интерфейса, нужно просто ввести логин и пароль.



Символы пароля не будут отображаться при вводе.

На системах с графическим интерфейсом вы увидите примерно следующее.



Вам нужно перейти на другую терминальную линию. Для этого нажмите Ctrl-Alt-F2 (или F3, F4 и тд). Последнее число - это номер терминальной линии, на которую вы хотите попасть. После чего вы увидите тоже самое приглашение входа как и на системах без графики.

## Packages (0.2)

### Debian

Пакетный менеджер должен узнать о новых дисках

|  |
| --- |
| [root@debian]# apt-cdrom add |

После проверяем, что на всех машинах подключены все четыре диска.

|  |
| --- |
| [root@debian]# less /etc/apt/source.list |

( Чтобы выйти из less нажмите **q** )

Устанавливаем требуемые пакеты на Debian

bind-utils = dnsutils

dhclient = isc-dhcp-common ( должен быть уже установлен )

nfs-utils = nfs-common

|  |
| --- |
| [root@debian]# apt install -y tcpdump net-tools curl vim lynx isc-dhcp-common dnsutils nfs-common cifs-utils sshpass openssh-server |

Пакеты для удобной работы с GNU/Linux. Поскольку для обычных людей слишком сложно запомнить разные варианты выполнения задания, мы будем использовать network-manager и на Debian, чтобы унифицировать выполнение задания.

|  |
| --- |
| [root@debian]# apt install -y bash-completion network-manager |

Для того чтобы автодополнение аргументов начало работать, нужно перезайти в систему!

### CentOS

Делаем бекап файлов yum.repos.d, возвращаем CentOS-Media.repo туда откуда взяли и правим.

|  |
| --- |
| [root@centos]# mkdir repos [root@centos]# mv /etc/yum.repos.d/CentOS-\* ./repos/ [root@centos]# cp ./repos/CentOS-Media.repo /etc/yum.repos.d/ [root@centos]# nano /etc/yum.repos.d/CentOS-Media.repo |

“\*” - означает любое количество любых символов. Перемещены будут все файлы, которые начинаются на “CentOS-”.

Включаем репозиторий, заменив значение enable на 1.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

[c7-media]

name=CentOS-$releasever - Media

baseurl=file:///media/CentOS/

file:///media/cdrom/

file:///media/cdrecorder/

gpgcheck=1

**enabled=1**

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| [root@centos]# mkdir /media/cdrom  [root@centos]# mkdir /media/CentOS [root@centos]# mount /dev/sr0 /media/cdrom  [root@centos]# mount /dev/sr1 /media/CentOS |

Устанавливаем необходимое ПО:

|  |
| --- |
| [root@centos]# yum -y install tcpdump net-tools curl vim lynx dhclient bind-utils nfs-utils cifs-utils |

Для проверки нужно ещё раз запустить эту команду, где напротив каждого пакета можно увидеть, что он "**already installed and latest version**".

|  |
| --- |
| [root@centos]# yum -y install /media/\*/sshpass\* |
|  |

Аналогично случаю выше “\*” будут развернуты и эта строка превратиться в “/media/cdrom/sshpass-1.0.6-1.el7.x86\_64.rpm” или “/media/CentOS/sshpass-1.0.6-1.el7.x86\_64.rpm” (зависит от того куда вставлен и смонтирован Additional.iso).

Также можно поставить:

|  |
| --- |
| [root@centos]# yum -y install nano bash-completion mc |

## Настройка имен хостов и достижение IP связанности

### Hosts and nsswitch

Сформируйте файл hosts.

|  |
| --- |
| [root@any]# nano /etc/hosts |

Добавляем в конец файла:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| 172.16.20.10 l-srv l-srv.skill39.wsr  10.10.10.1 l-fw l-fw.skill39.wsr  172.16.50.2 l-rtr-a l-rtr-a.skill39.wsr  172.16.55.2 l-rtr-b l-rtr-b.skill39.wsr  172.16.200.61 l-cli-b l-cli-b.skill39.wsr  20.20.20.5 out-cli out-cli.skill39.wsr  20.20.20.100 r-fw r-fw.skill39.wsr  192.168.20.10 r-srv r-srv.skill39.wsr  192.168.10.2 r-rtr r-rtr.skill39.wsr  192.168.100.100 r-cli r-cli.skill39.wsr  10.10.10.10 isp |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Формат: ip-address name1 name2 ... nameN

Ответы DNS сервера должны иметь более высокий приоритет.

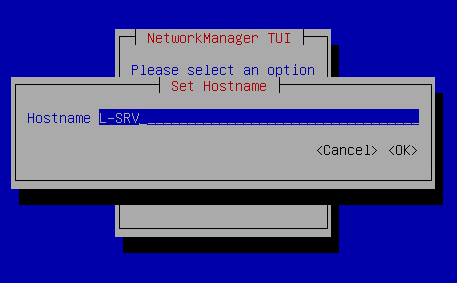
|  |
| --- |
| [root@any]# nano /etc/nsswitch.conf |

В строке, которая начинается с “hosts: ”, меняем местами слова files и dns.

### Адресация и firewalld

#### Basic configuration

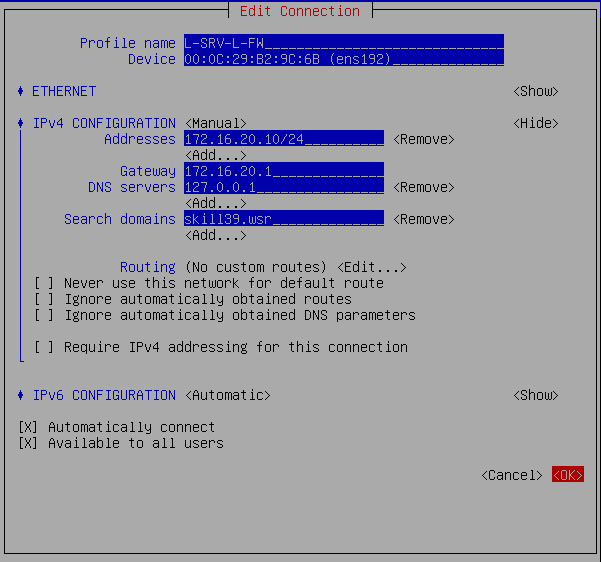
Задаем хостнейм. Внимание на регистр!!



Далее, настраиваем адресацию, гейтвей, dns-суффикс и т.д..

По поводу DNS:

* L-SRV и R-SRV будут сами серверами DNS, поэтому указываем 127.0.0.1.
* OUT-CLI должен резолвить через L-FW.
* Когда в задании R-SRV является secondary DNS сервером (это не всегда так), то обычно он должен отвечать на запросы R-CLI.



**Это пример!!! На L-SRV будет служба DNS, поэтому там в “DNS servers” 127.0.0.1, кто будет DNS сервером у остальных указано в задании.**

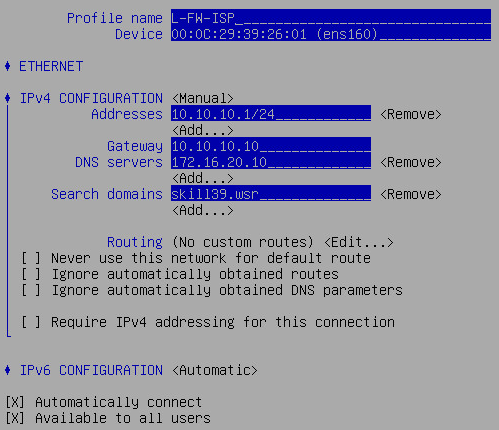
Перезапускаем профиль



После чего должны увидеть следующее:



Настраиваем L-FW



Для L-FW шлюзом будет ISP, поэтому на этом интерфейсе указываем его.

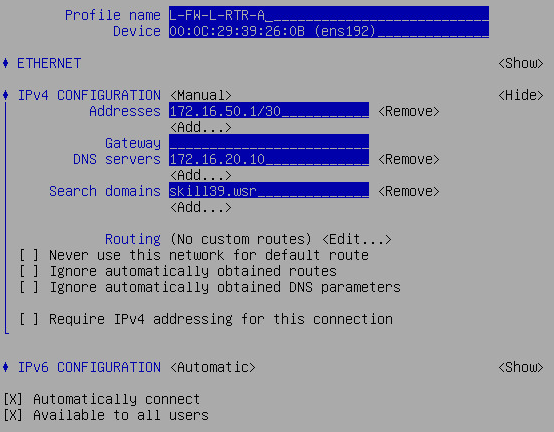
На остальных интерфейсах, шлюз указывать **НЕ НУЖНО**!

Необходимо убедиться, что опции:

* Automatically connect
* Available to all users

Установлены!

Пример Profile для L-FW.



Убеждаемся, что всё работает!



По возможности пингуем соседние устройства!

#### Forwarding

Далее необходимо включить возможность маршрутизации.

|  |
| --- |
| sysctl -a | grep forward  sysctl -w net.ipv4.ip\_forward=1 >> /etc/sysctl.conf |

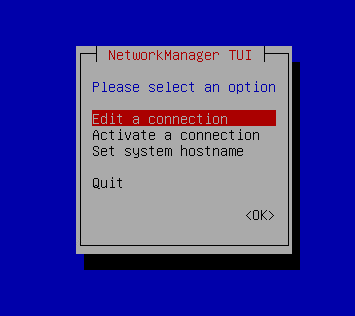
Данные действия необходимо проделать на всех устройствах, выступающих в роли маршрутизатора.

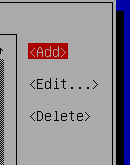
Подобным образом настраиваем: L-FW, L-RTR-A, L-RTR-B, R-FW, R-RTR-A.

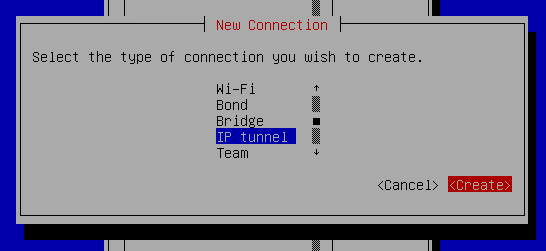
**Второй вариант** - это включить маршрутизацию при настройке FRR. Там в режиме конфигурации можно ввести “ip forwarding”.

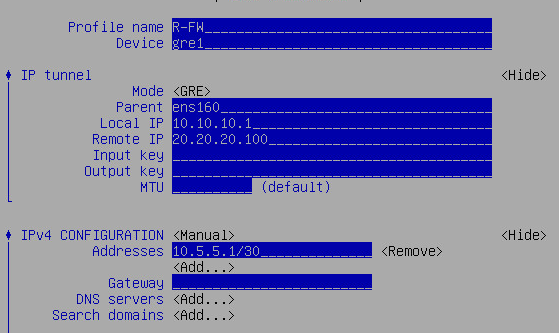
#### GRE

Между L-FW и R-FW создаём GRE туннель

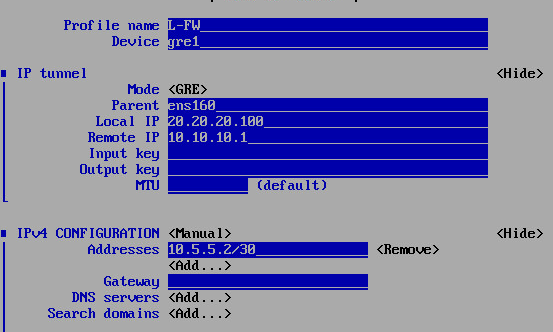




.

,

Profile name - имя профиля, а не имя виртуалки. Можно задать любое.



Необходимо поставить TTL отличный от 0, иначе не будет ospf соседства между L-FW и R-FW.

**Первый вариант:**

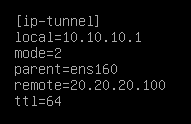
|  |
| --- |
| [root@ANY-FW]# nmcli connection modify CON-NAME ip-tunnel.ttl 64 |

CON-NAME - имя профиля

После этого нужно деактивировать и активировать соединение заново.

**Второй вариант -** можно вручную поправить файл конфигурации соединения.

|  |
| --- |
| [root@ANY-FW]# vim /etc/NetworkManager/system-connections/GRE.nmconnection |



**Перезагрузите VM!**

На R-FW, раскидываем интерфейсы по нужным зонам и добавляем протокол GRE в зону external.

#### Firewalld

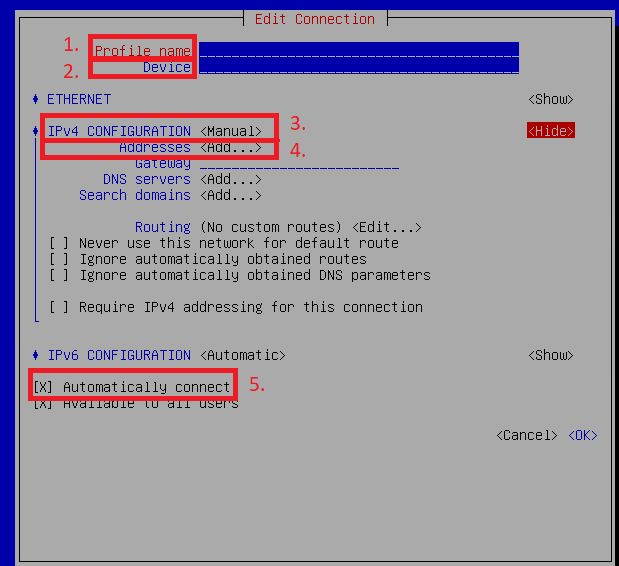
Настройте firewalld

|  |
| --- |
| [root@R-FW]# firewall-cmd --zone=external \  --add-interface=ens160 \  --permanent [root@R-FW]# firewall-cmd --zone=external \  **--add-service=gre** \  --permanent  [root@R-FW]# firewall-cmd --zone=trusted \  --add-interface=ens192 \  --permanent [root@R-FW]# firewall-cmd --zone=trusted \  --add-interface=ens224 \  --permanent [root@R-FW]# firewall-cmd --zone=trusted \  --add-interface=gre1 \  --permanent  [root@R-FW]# firewall-cmd --reload |

#### Dummy

**Первый вариант:**

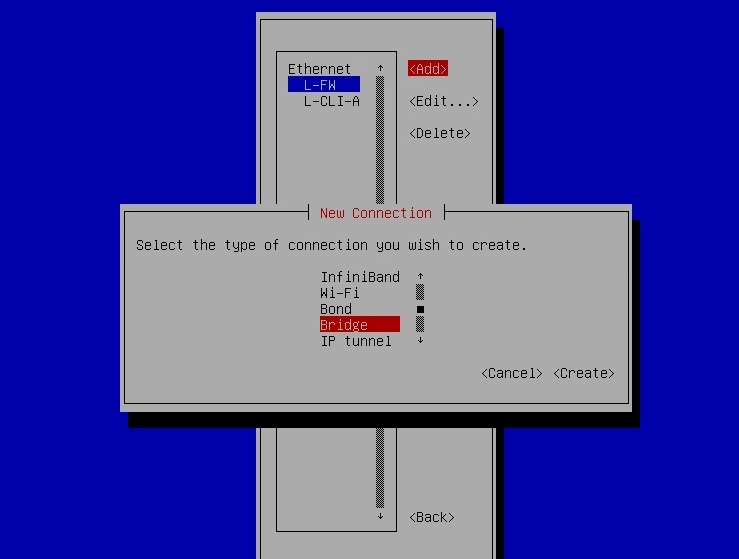
На L-RTR-A и L-RTR-B создаем dummy интерфейсы. В nmtui этого сделать нельзя, поэтому проще всего это будет сделать через nmcli. Если вы поставили bash-completion, то с вводом этой длинной команды проблем не будет. Проще всего запомнить эту команду можно посмотрев на стандартный интерфейс nmtui.

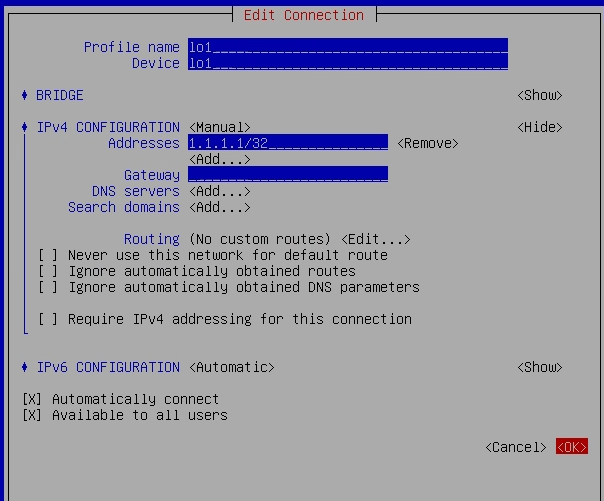


1. connection.id (или conn-name)
2. connection.interface-name
3. ipv4.method
4. ipv4.address
5. connection.autoconnect

|  |
| --- |
| [root@L-RTR-A]# nmcli connection add connection.id lo1 \  connection.interface-name lo1 \  connection.type dummy \  connection.autoconnect yes\  ipv4.method manual \  ipv4.address 1.1.1.1/32  [root@L-RTR-B]# nmcli connection add connection.id lo1 \  connection.interface-name lo1 \  connection.type dummy \  connection.autoconnect yes\  ipv4.method manual \  ipv4.address 2.2.2.2/32 |

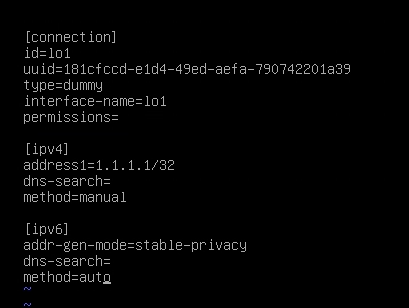
**Второй вариант -** создать bridge и изменить тип на dummy





Меняем тип:

|  |
| --- |
| [root@L-RTR-A]# cd /etc/NetworkManager/system-connections/ [root@L-RTR-B]# vim lo1.nmconnection |



После перезагрузки интерфейс станет типа dummy.

### Достижение IP связанности

Устанавливаем frr

|  |
| --- |
| [root@L-FW]# apt install frr |

Включаем поддержку ospf

|  |
| --- |
| [root@L-FW]# vim /etc/frr/daemons |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| ospfd=yes |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Перезапускаем демон

|  |
| --- |
| [root@L-FW]# systemctl restart frr |

и конфигурим ospf в cisco-like стиле на примере L-FW, подключившись к frr с помощью vtysh.

|  |
| --- |
| [root@L-FW]# vtysh |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| conf t  router ospf  network 172.16.20.0/24 area 0  network 172.16.50.0/30 area 0  network 172.16.55.0/30 area 0  network 10.5.5.0/30 area 0  network 5.5.5.0/27 area 0  passive-interface ens160  passive-interface ens256 **write** |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

На L-RTR-A и L-RTR-B не забудьте анонсировать dummy интерфейсы.

Устанавливаем FRR на CentOS

|  |
| --- |
| [root@R-FW]# yum install -y /media/\*/libyang\* [root@R-FW]# yum install -y /media/\*/frr\* |

Включаем поддержку ospf

|  |
| --- |
| [root@R-FW]# vim /etc/frr/daemons |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| zebra=yes ospfd=yes |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Добавляем демона в автозапуск и запускаем его

|  |
| --- |
| [root@R-FW]# systemctl enable frr [root@R-FW]# systemctl start frr |

**Не забывай добавлять интерфейсы внутри RIGHT в зону trusted!!!**

|  |
| --- |
| [root@R-RTR]# firewall-cmd --zone=trusted \  --add-interface=ens160 \  --permanent  [root@R-RTR]# firewall-cmd --zone=trusted \  --add-interface=ens192 \  --permanent [root@R-RTR]# firewall-cmd --reload |

## DHCP и Relay Agent (1.2)

### Basic DHCP

На L-RTR-A

Устанавливаем dhcp

|  |
| --- |
| [root@L-RTR-A]# apt install isc-dhcp-server |

Прописываем какие будем слушать интерфейсы

|  |
| --- |
| [root@L-RTR-A]# vim /etc/default/isc-dhcp-server |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| INTERFACESv4="ens192 ens224" INTERFACESv6="" |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Правим конфиг

|  |
| --- |
| [root@L-RTR-A]# vim /etc/dhcp/dhcpd.conf |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| option domain-name "skill39.wsr"; option domain-name-servers 172.16.20.10;  default-lease-time 600; max-lease-time 7200;  ddns-update-style none;  authoritative;  subnet 172.16.50.0 netmask 255.255.255.252 {}  subnet 172.16.100.0 netmask 255.255.255.0 {  range 172.16.100.65 172.16.100.75;  option routers 172.16.100.1; }  subnet 172.16.200.0 netmask 255.255.255.0 {  range 172.16.200.65 172.16.200.75;  option routers 172.16.200.1; }  host clib {  hardware ethernet 00:0c:29:a3:ad:bf;  fixed-address 172.16.200.61; } |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Запускаем DHCP

|  |
| --- |
| [root@L-RTR-A]# systemctl start isc-dhcp-server [root@L-RTR-A]# systemctl enable isc-dhcp-server |

Смотрим выданные IP

|  |
| --- |
| [root@L-RTR-A]# dhcp-lease-list |

### Relay Agent

На L-RTR-B

|  |
| --- |
| [root@L-RTR-B]# apt install isc-dhcp-relay |







## DNS

### (Dynamic NAT)-LEFT (0.3)

Для того чтобы L-SRV мог обращаться к ISP, необходимо настроить NAT.

|  |
| --- |
| [root@L-FW]# iptables -t nat -A POSTROUTING -o ens160 -j MASQUERADE |

После данной команды L-SRV должен получить возможность общаться с ISP. Теперь нужно сохранить правила netfilter, чтобы они не потерялись после ребута.

Установим соответствующий пакет.

|  |
| --- |
| [root@L-FW]# apt search netfilter | less [root@L-FW]# apt install iptables-persistent |

Во время установки было предложено сохранить правила в файл.

|  |
| --- |
| [root@L-FW]# cat /etc/iptables/rules.v4 |

В данный файл в дальнейшем мы будем дописывать необходимые правила.

### DNS: ISP Forwarders (0.3)

Теперь установим bind9.

|  |
| --- |
| [root@L-SRV]# apt install bind9 |

Далее необходимо сконфигурировать форвардеры и отключить DNSSEC.

|  |
| --- |
| [root@L-SRV]# vim /etc/bind/named.conf.options |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| options {  directory "/var/cache/bind";  forwarders {  10.10.10.10;  };  dnssec-validation no;  listen-on-v6 { any; }; }; |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### DNS: Lookup order

В случае корректной работы DNS-сервисов ответы DNS должны иметь более высокий приоритет.

На всех VM меняем местами files и dns.

|  |
| --- |
| [root@ANY]# vim /etc/nsswitch.conf |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| ...  hosts: dns files  ... |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### DNS: Forward and Reverse zones (0.6)

Далее необходимо создать файлы зон в /opt/dns и настроить права доступа.

|  |
| --- |
| [root@L-SRV]# mkdir /opt/dns [root@L-SRV]# cp /etc/bind/db.local /opt/dns/skill39.db [root@L-SRV]# cp /etc/bind/db.127 /opt/dns/db.172 [root@L-SRV]# cp /etc/bind/db.127 /opt/dns/db.192 [root@L-SRV]# chown -R bind:bind /opt/dns |

Далее необходимо добавить “/opt/dns” в соответствующий конфиг Apparmor

|  |
| --- |
| [root@L-SRV]# vim /etc/apparmor.d/usr.sbin.named |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| /opt/dns/\*\* rw, |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Перезапускаем Apparmor.

|  |
| --- |
| [root@L-SRV]# systemctl restart apparmor.service |

Теперь можно приступить к конфигурированию файлов зон. Но для начала скажем bind где их искать.

|  |
| --- |
| [root@L-SRV]# vim /etc/bind/named.conf.default-zones |

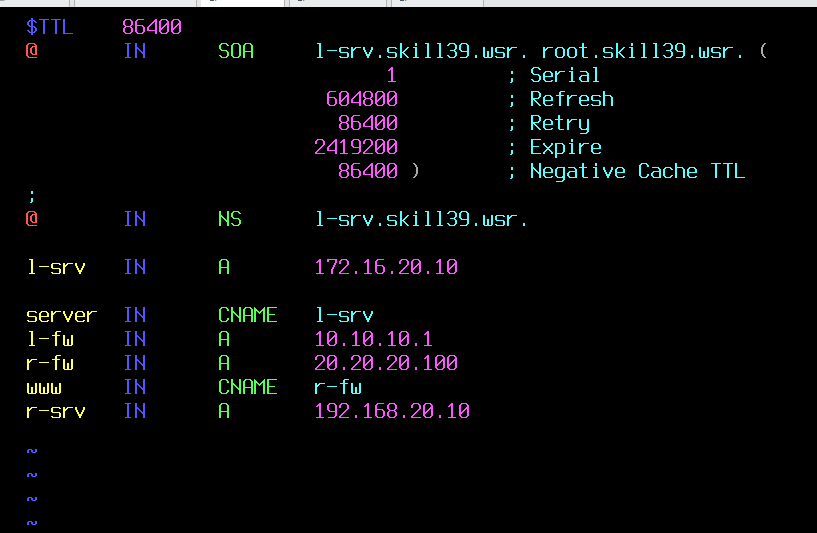
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| zone "skill39.wsr" {  type master;  allow-transfer { any; };  file "/opt/dns/skill39.db"; };  zone "16.172.in-addr.arpa" {  type master;  allow-transfer { any; };  file "/opt/dns/db.172"; };  zone "20.168.192.in-addr.arpa" {  type master;  allow-transfer { any; };  file "/opt/dns/db.192"; }; |
|  |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

После чего нужно внести необходимые изменения в соответствующие файлы.

Прямая зона:



### DNS-Proxy

OUT-CLI должен иметь доступ к DNS по внешнему адресу L-FW, поэтому на L-FW необходимо настроить DNAT.

|  |
| --- |
| [root@L-FW]# iptables -t nat -A PREROUTING -i ens160 -p udp --dport 53 -j DNAT --to 172.16.20.10:53 |

**iptables-save > /etc/iptables/rules.v4.**

На данный момент OUT-CLI должен получать ответы на DNS запросы.

### DNS: Dynamic DNS (0.5)

Читаем ман

|  |
| --- |
| [root@L-RTR-A]# man dhcpd.conf |

Меняем значение ddns-update-style и добавляем зоны, которые хотим апдейтить

|  |
| --- |
| [root@L-RTR-A]# vim /etc/dhcp/dhcpd.conf |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| ... ddns-update-style interim;  update-static-leases on; ...  zone skill39.wsr. {  primary 172.16.20.10; }  zone 16.172.in-addr.arpa. {  primary 172.16.20.10; } |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

В bind9 в нужных зонах разрешаем апдейты от dhcp сервера:

|  |
| --- |
| [root@L-SRV]# vim /etc/bind/named.conf.default-zones |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| ... allow-update { 172.16.50.2; }; ... |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### DNS: Secondary DNS (0.3)

Чтобы хранить файлы зон в “/opt/dns” необходимо отключить SELINUX.

Выключаем / проверяем

|  |
| --- |
| [root@R-SRV]# setenforce 0 [root@R-SRV]# getenforce Permissive |

Правим конфиг, чтобы SELINUX работал в режиме Permissive после перезагрузки.

|  |
| --- |
| [root@R-SRV]# vim /etc/selinux/config |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| # enforcing - SELinux security policy is enforced. # permissive - SELinux prints warnings instead of enforcing. # disabled - No SELinux policy is loaded.  SELINUX=permissive |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Смотрим какому пользователю принадлежит named.conf.

|  |
| --- |
| [root@R-SRV]# ls -l /etc/named.conf |

Создаём директорию для зон.

|  |
| --- |
| [root@R-SRV]# mkdir /opt/dns [root@R-SRV]# chown named:named /opt/dns |

Далее правим конфиг named.

|  |
| --- |
| [root@R-SRV]# vim /etc/named.conf |

В named.conf необходимо внести следующие изменения:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| listen-on port 53 { any; }; allow-query { any; };  dnssec-enable no; dnssec-validation no;  forwarders { 20.20.20.10; };  zone "skill39.wsr" {  type slave;  masters { 172.16.20.10; };  file "/opt/dns/skill39.db"; };  zone "16.172.in-addr.arpa" {  type slave;  masters { 172.16.20.10; };  file "/opt/dns/db.172"; };  zone "20.168.192.in-addr.arpa" {  type slave;  masters { 172.16.20.10; };  file "/opt/dns/db.192"; }; |

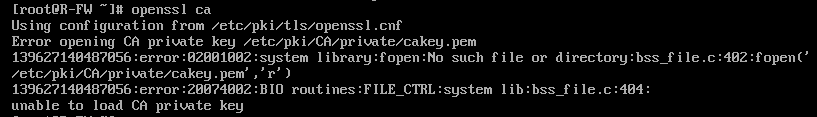
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Запускаем named и добавляем в автозагрузку.

|  |
| --- |
| [root@R-SRV]# systemctl start named [root@R-SRV]# systemctl enable named |

## OpenSSL CA ( 1 )

Ищем конфиг



Делаем копию конфигурационного файла и читаем

|  |
| --- |
| [root@R-FW]# cp /etc/pki/tls/openssl.cnf{,.bak}  [root@R-FW]# less /etc/pki/tls/openssl.cnf |

Нужно создать следующие директории и файлы

|  |
| --- |
| [root@R-FW]# mkdir /etc/ca  [root@R-FW]# cd /etc/ca  [root@R-FW]# mkdir certs newcerts private crl [root@R-FW]# touch index.txt [root@R-FW]# echo **-n** '00' > serial  [root@R-FW]#  [root@R-FW]# vim /etc/pki/tls/openssl.cnf |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ -\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| [ CA\_default ] # Меняем рабочую директорию, на /etc/ca dir = /etc/ca  ...  # Ставим политику ANYTHING policy = policy\_anything  ...  [ req\_distinguished\_name ] # Меняем умолчания, чтобы при генерации каждый раз не вводить одно и тоже countryName\_default = RU 0.organizationName\_default = WorldSkills Russia  ...  **# man x509\_config**  [ host\_cert ] # Для простоты будем использовать одинаковые x509 расширения для серверных и клиентских сертификатов. Важные поля - keyUsage и extendedKeyUsage. basicConstraints=CA:FALSE keyUsage = **digitalSignature, keyEncipherment** extendedKeyUsage = **serverAuth, clientAuth** subjectKeyIdentifier=hash authorityKeyIdentifier=keyid,issuer  [ ca\_ext ] subjectKeyIdentifier=hash authorityKeyIdentifier=keyid:always,issuer basicConstraints=CA:TRUE |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Вместе с OpenVPN в директории “/usr/share/doc/openvpn/sample-key” лежат уже сгенерированные ключи и пример конфига openssl. В самом конце можно найти секцию server, в которой можно подсмотреть значения x509.**

**Если прям совсем всё плохо, то оттуда можно взять все необходимые ключи.**

|  |
| --- |
| # Сгенерировать запрос на сертификат # - ключ не шифровать # - новый запрос # - сам запрос положить в текущую директорию # - приватный ключ должен быть в ./private/cakey.pem # - использовать x509\_extensions из секции v3\_ca [root@R-FW]# openssl req -nodes \  -new \  -out cacert.csr \  -keyout private/cakey.pem \  -extensions ca\_ext  # Подписать запрос на сертификат # - сертификат - само-подписан # - запрос на сертификат в файле cacert.csr # - подписанный сертификат положить в cacert.pem # - использовать x509\_extensions из секции v3\_ca [root@R-FW]# openssl ca -selfsign \  -in cacert.csr \  -out cacert.pem \  -extensions ca\_ext |

Пример генерации других ключей.

|  |
| --- |
| [root@R-FW]# openssl req -nodes \  -new \  -out custom.csr \  -keyout private/custom.key  [root@R-FW]# openssl ca -in custom.csr \  -out custom.crt \  -extensions host\_ext |

## OpenVPN (0.7)

### OpenVPN Basic (0.5)

Создаем директорию:

|  |
| --- |
| [root@ANY]# mkdir /opt/vpn |

Создаем файл с параметрами Диффи-Хеллмана, этот файл должен присутствовать только на сервере (эту команду можно найти в server.conf):

|  |
| --- |
| [root@L-FW]# openssl dhparam -out /opt/vpn/dh.pem 2048 |

Копируем сертификаты в /opt/vpn на L-FW с R-FW

Устанавливаем Openvpn

|  |
| --- |
| [root@L-FW]# apt install openvpn  [root@OUT-CLI]# yum install /media/\*/pkcs11-helper\* [root@OUT-CLI]# yum install /media/\*/openvpn\* |

Копируем дефолтный конфиг

|  |
| --- |
| [root@L-FW]# zcat /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz > /etc/openvpn/server.conf [root@L-FW]# vim /etc/openvpn/server.conf |

Правим конфиг согласно задания:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| local 10.10.10.1 port 1122 proto udp dev tun  ca /opt/vpn/cacert.pem cert /opt/vpn/fw.crt key /opt/vpn/fw.key dh /opt/vpn/dh.pem  server 5.5.5.0 255.255.255.224 push "route 192.168.0.0 255.255.0.0" push "route 172.16.0.0 255.255.0.0"  push "route 10.5.5.0 255.255.255.252"  push "route 1.1.1.1 255.255.255.255"  push "route 2.2.2.2 255.255.255.255"  duplicate-cn keepalive 10 120 cipher AES-256-CBC comp-lzo  user nobody group nobody persist-key persist-tun  status /var/log/openvpn/openvpn-status.log  verb 3 explicit-exit-notify 1 |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Добавляем в автозапуск и стартуем сервис

|  |
| --- |
| [root@L-FW]# systemctl enable openvpn@server [root@L-FW]# systemctl start openvpn@server |

Копируем дефолтную конфигурацию для клиента

|  |
| --- |
| [root@OUT-CLI]# cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf /etc/openvpn/client.conf [root@OUT-CLI]# vim /etc/openvpn/client.conf |

Задаем настройки согласно задания и отсекаем все ненужное

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| client dev tun proto udp remote 10.10.10.1 1122  resolv-retry infinite  nobind user nobody group nobody persist-key persist-tun  ca /opt/vpn/cacert.pem cert /opt/vpn/out.crt key /opt/vpn/out.key  cipher AES-256-CBC comp-lzo  verb 3 |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Добавляем “/opt/vpn” в PATH для всех пользователей

|  |
| --- |
| [root@OUT-CLI]# vim /etc/profile |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| # Добавляем в конец файла  PATH=$PATH:/opt/vpn |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Создаём скрипт подключения по OpenVPN и даём права на исполнение для всех.

|  |
| --- |
| [root@OUT-CLI]# vim /opt/vpn/start\_vpn.sh [root@OUT-CLI]# chmod a+x /opt/vpn/start\_vpn.sh |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| #!/bin/bash  systemctl start openvpn@client sed -ri 's/^(nameserver).\*/\1 172.16.20.10/' /etc/resolv.conf |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Создаём скрипт отключения OpenVPN и даём права на исполнение для всех.

|  |
| --- |
| [root@OUT-CLI]# vim /opt/vpn/stop\_vpn.sh [root@OUT-CLI]# chmod a+x /opt/vpn/stop\_vpn.sh |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| #!/bin/bash systemctl stop openvpn@client nmcli connection up ens160 &> /dev/null |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Если** будет необходимо, чтобы туннель мог поднять любой пользователь, то:

Разрешаем всем запускать и останавливать openvpn@client для всех.

|  |
| --- |
| [root@OUT-CLI]# visudo |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| ... ALL ALL=(ALL) NOPASSWD: /bin/systemctl start openvpn@client, /bin/systemctl stop openvpn@client ... |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

This line means:

* **ALL** (any) users can execute
* from **ALL** hosts,
* acting as **ALL** (any) users,
* and run "/bin/systemctl start/stop openvpn@client" without a password.

В скриптах нужно добавить "sudo" перед командой "systemctl".

Перемещаем замены DNS в юнит файл openvpn.

|  |
| --- |
| [root@OUT-CLI]# vim /lib/systemd/system/openvpn@.service |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| [Service] ... ExecStartPost=/bin/sed -ri 's/^(nameserver).\*/\1 172.16.20.10/' /etc/resolv.conf ExecStopPost=/bin/nmcli connection up ens160 ... |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Просим systemd перечитать конфиги

|  |
| --- |
| [root@OUT-CLI]# systemctl daemon-reload |

### VPN Clients have full access to LANs (0.2)

После того как клиент начал подключаться к серверу и необходимые маршруты появились в таблице маршрутизации, необходимо убедиться, что клиент может ходить в LEFT и RIGHT через данное VPN подключение, поэтому нужно сделать traceroute ко всем локальным машинам во внутренних сетях.

## IPSec (0.4)

|  |
| --- |
| # Разрешаем ESP и IKE [root@R-FW]# firewall-cmd --zone=external \  --add-service=ipsec \  --permanent  [root@R-FW]# firewall-cmd --reload |

Установка демона

|  |
| --- |
| [root@L-FW]# apt install libreswan [root@R-FW]# yum install libreswan |

Добавляем в автозагрузку

|  |
| --- |
| [root@ANY]# systemctl enable ipsec.service |

Читаем задание и вспоминаем что нужно писать в конфиг:

|  |
| --- |
| [root@ANY]# man ipsec.conf |

Задаем параметры первой и второй фазы:

|  |
| --- |
| [root@ANY]# vim /etc/ipsec.conf |

Добавляем ( НЕ ЗАМЕНЯЕМ ) в конец конфига:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| conn encrypted\_gre  auto=start  authby=secret  type=tunnel  ike=3des-sha1;modp2048  phase2=esp  phase2alg=aes-sha2;modp2048  left=10.10.10.1  leftprotoport=gre  right=20.20.20.100  rightprotoport=gre |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Создаём общий ключ для аутентификации:

|  |
| --- |
| [root@ANY]# vim /etc/ipsec.d/gre.secrets |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| # IP-1 IP-2 : PSK "Pre-shared key"  20.20.20.100 10.10.10.1 : PSK "P@$$w0rd" |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Запускаем сервис:

|  |
| --- |
| [root@ANY]# systemctl start ipsec.service |

Если что-то поправили то делаем reload, а затем restart:

|  |
| --- |
| [root@ANY]# systemctl reload ipsec.service [root@ANY]# systemctl restart ipsec.service |

Убеждаемся, что SA был создан и политики добавлены:

|  |
| --- |
| [root@ANY]# ip xfrm state [root@ANY]# ip xfrm policy |

## SSH (0.6)

Разрешаем подключаться руту на всех машинах:

|  |
| --- |
| [root@ANY]# vim /etc/ssh/sshd\_config |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| PermitRootLogin yes |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Или

|  |
| --- |
| [root@ANY]# sed -ie 's/#PermitRoot.\*/PermitRootLogin yes/' /etc/ssh/sshd\_config |

На L-FW доступ ограничен пользователями ssh\_p и ssh\_c

|  |
| --- |
| [root@L-FW]# vim /etc/ssh/sshd\_config |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| PermitRootLogin yes AllowUsers ssh\_p root ssh\_c |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дабы не забыть, сразу добавим пользователей в систему.

|  |
| --- |
| [root@L-FW]# adduser ssh\_p [root@L-FW]# adduser ssh\_c |

На OUT-CLI из под учетки рут должны заходить на L-FW пользователем ssh\_p без пароля.

|  |
| --- |
| [root@OUT-CLI]# ssh-keygen [root@OUT-CLI]# ssh-copy-id ssh\_p@10.10.10.1 |

Проверяем.

|  |
| --- |
| [root@OUT-CLI]# ssh ssh\_p@10.10.10.1 |

## Distribute files

Доверяем нашему CA.

|  |
| --- |
| [root@L-FW]# cp /opt/vpn/cacert.pem /usr/local/share/ca-certificates/ca.crt [root@L-FW]# update-ca-certificates |

Делаем чтобы другие клиенты тоже доверяли нашему CA

|  |
| --- |
| [root@L-FW:~]# cd /usr/local/share/ca-certificates/ [root@L-FW:/usr/local/share/ca-certificates]# scp ca.crt root@IP-ADDRESS:/usr/local/share/ca-certificates/ca.crt [root@L-FW:/usr/local/share/ca-certificates]# ssh root@IP-ADDRESS update-ca-certificates |

На CentOS проделываем аналогичные операции.

Отличие только в якорях.

|  |
| --- |
| [root@R-FW]# cp /opt/vpn/cacert.pem /etc/pki/ca-trust/source/anchors/ca.crt [root@R-FW]# update-ca-trust extract |

## Logging (0.8)

Создаём необходимые директории

|  |
| --- |
| [root@L-SRV]# mkdir -P /opt/logs/{L-FW,L-SRV} |

Читаем ман

|  |
| --- |
| [root@L-SRV]# man rsyslog.conf |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Available Properties:

* **hostname** - hostname from the message
* **syslogfacility-text** - the facility from the message - in text form
* **syslogseverity-text** - severity from the message - in text form

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нашли названия переменных, правим конфиг

|  |
| --- |
| [root@L-SRV]# vim /etc/rsyslog.conf |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| # # MODULES # Раскомментируем модуль получения логов по UDP  module(load="imudp") input(type="imudp" port="514")  # # RULES # Перед остальными правилами добавляем  if ($hostname == "L-SRV" and $syslogfacility-text == "auth")  then /opt/logs/L-SRV/auth.log if ($hostname == "l-fw" )  then /opt/logs/L-FW/err.log |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Проверяем на ошибки, перезапускаем, если все ОК.

|  |
| --- |
| [root@L-SRV]# rsyslog -N 1 [root@L-SRV]# systemctl restart rsyslog.service |

Шлем тестовые сообщения, проверяем, что они пишутся куда следует.

|  |
| --- |
| [root@L-SRV]# logger -p **auth.err** Test "AUTH.ERR" from L-SRV [root@L-SRV]# logger -p **auth.info** Test "AUTH.INFO" from L-SRV [root@L-SRV]# cat /opt/logs/L-SRV/auth.log |

Идём на L-FW

|  |
| --- |
| [root@L-FW]# vim /etc/rsyslog.conf |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| # # RULES #  \*.err @172.16.20.10 |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Проверяем на ошибки, перезапускаем

|  |
| --- |
| [root@L-FW]# rsyslog -N 1 [root@L-FW]# systemctl restart rsyslog.service |

Отправляем тестовые сообщения

|  |
| --- |
| [root@L-FW]# logger -p auth.err Test "AUTH.ERR" from L-FW [root@L-FW]# logger -p mail.err Test "MAIL.ERR" from L-FW |

Идём на L-SRV и проверяем, что всё пришло и записано куда нужно

|  |
| --- |
| [root@L-SRV]# cat /opt/logs/L-FW/err.log |

## Web (1.8)

### Apache (0.4)

yum install httpd php

mkdir -p **/var/www/html/out**

|  |
| --- |
| [root@R-SRV]# vim /etc/httpd/conf/httpd.conf |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| ... DocumentRoot /var/www/html/out ... |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| [root@R-SRV]# vim /var/www/html/out/index.html |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| Hello, www.skill39.wsr is here! |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| [root@R-SRV]# vim /var/www/html/out/date.php |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| <?php print(date('Y-m-d H:i:s')); ?> |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| [root@R-SRV]# php /var/www/html/out/date.php |

|  |
| --- |
| [root@R-SRV]# vim /etc/php.ini |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| ... date.timezone = “Europe/Moscow” ... |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| [root@R-SRV]# php /var/www/html/out/date.php |

|  |
| --- |
| [root@R-SRV]# systemctl start httpd [root@R-SRV]# systemctl enable httpd |

### NGINX (1.4)

#### Access (0.2)

|  |
| --- |
| [root@R-FW]# yum install /media/cdrom/nginx\*.rpm |

|  |
| --- |
| [root@R-FW]# cp /etc/nginx/conf.d/{default,task}.conf |

|  |
| --- |
| [root@R-FW]# systemctl start nginx [root@R-FW]# systemctl enable nginx  [root@R-FW]# firewall-cmd --zone=external \  --add-service=http \  --permanent [root@R-FW]# firewall-cmd --zone=external \  --add-service=https \  --permanent [root@R-FW]# firewall-cmd --reload |

#### Reverse SSL proxy (0.3)

1. На R-FW настройте реверс-прокси на основе NGINX:
   1. Сайт www.skill39.wsr должен быть доступен из внешней сети по внешнему адресу R-FW
   2. Все настройки, связанные с заданием, должны содержаться в отдельном конфигурационном файле в каталоге /etc/nginx/conf.d/task.conf
      1. Конфигурация основного файла должна быть минимальной и не влиять на работу NGINX в рамках выполнения задания.

|  |
| --- |
| [root@R-FW]# vim /etc/nginx/conf.d/task.conf |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| server {  listen 80;  server\_name www.skill39.wsr;  location / {  proxy\_pass http://192.168.20.10:8088;  } } |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| [root@R-FW]# nginx -t [root@R-FW]# systemctl start nginx [root@R-FW]# systemctl enable nginx |

#### Redirection (0.3)

c) Настройте SSL и автоматическое перенаправление незащищенных запросов на HTTPS-порт того же самого сервера.

|  |
| --- |
| [root@R-FW]# vim /etc/nginx/conf.d/task.conf |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| server {  listen 80;  server\_name www.skill39.wsr;  return 301 https://www.skill39.wsr$request\_uri; }  server {  listen 443 ssl;  server\_name www.skill39.wsr;   ssl\_certificate /etc/nginx/conf.d/server.crt;  ssl\_certificate\_key /etc/nginx/conf.d/server.key;   location / {  proxy\_pass http://192.168.20.10:8088;  } } |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| [root@R-FW]# nginx -t [root@R-FW]# systemctl restart nginx |

#### Passive check (0.3)

Реализуйте пассивную проверку работоспособности бекенда.

- Считать веб-сервер неработающим после 4 ошибок.

- Считать веб-сервер неработающим в течение 43 секунд.

|  |
| --- |
| [root@R-FW]# vim /etc/nginx/conf.d/task.conf |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| upstream backend {  server 192.168.20.10:8088 max\_fails=4 fail\_timeout=43s; }  server {  listen 80;  server\_name www.skill39.wsr;  return 301 https://www.skill39.wsr$request\_uri; }  server {  listen 443 ssl;  server\_name www.skill39.wsr;   ssl\_certificate /etc/nginx/conf.d/server.crt;  ssl\_certificate\_key /etc/nginx/conf.d/server.key;   location / {  proxy\_pass backend;  proxy\_set\_header Host $http\_host;  } } |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| [root@R-FW]# nginx -t [root@R-FW]# systemctl restart nginx |

#### Caching (0.3)

Реализуйте кэширование:

- Запросы к любым PHP-скриптам не должны кэшироваться.

- Кэширование успешных запросов к остальным типам данных должно выполняться в течение 40 секунд.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| upstream backend {  server 192.168.20.10:8088 max\_fails=4 fail\_timeout=43s; }   server {  listen 80;  server\_name www.skill39.wsr;  return 301 https://www.skill39.wsr$request\_uri; }  proxy\_cache\_path /var/cache/nginx keys\_zone=APP:10m inactive=40s; # key\_zone = ИМЯ\_ЗОНЫ:ОБЪЕМ\_ПАМЯТИ # inactive = удалять кеш через N времени, если не было обращений.  server {  listen 443 ssl;  server\_name www.skill39.wsr;   ssl\_certificate /etc/nginx/conf.d/server.crt;  ssl\_certificate\_key /etc/nginx/conf.d/server.key;   location / {  proxy\_pass http://backend;  proxy\_set\_header Host $http\_host;  proxy\_cache APP;  proxy\_cache\_valid 200 301 40s;  # Кешируем только валидные запросы и перенаправления.  }   location ~ \.php$ {  proxy\_pass http://backend;  proxy\_set\_header Host $http\_host;  **proxy\_cache\_bypass** *true*;  } } |
|  |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Запускаем

|  |
| --- |
| watch ls -la --full-time /var/cache/nginx |

Идём на L-FW и запускаем lynx

|  |
| --- |
| lynx https://www.skill39.wsr lynx https://www.skill39.wsr/date.php |

Должен появиться файл кеша для index.html

Убеждаемся, что php не кешируется.

|  |
| --- |
| for i in $(seq 10 ); do curl https://www.skill39.wsr/date.php; echo ""; sleep 1; done |

## RSYNC (0.7)

Ставим

|  |
| --- |
| [root@L-SRV]# apt install rsync |

Создаем каталог и юзера

|  |
| --- |
| [root@L-SRV]# mkdir /opt/sync [root@L-SRV]# useradd -d /opt/sync mrsync [root@L-SRV]# passwd mrsync [root@L-SRV]# chown mrsync:mrsync /opt/sync |

Читаем ман!

|  |
| --- |
| [root@L-SRV]# man rsyncd.conf /EXAMPLE |

Копипастим из мана и правим:

|  |
| --- |
| [root@L-SRV]# vim /etc/rsyncd.conf |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| uid = mrsync gid = mrsync use chroot = yes max connections = 4 syslog facility = local5 pid file = /var/run/rsyncd.pid  [wsr] path = /opt/sync comment = WSR hosts allow = l-cli-a l-cli-b auth users = sync read only = false secrets file = /etc/rsyncd.secrets |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Создаем юзера для аутентификации

|  |
| --- |
| [root@L-SRV]# vim /etc/rsyncd.secrets |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| sync:parol666 |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| [root@L-SRV]# chmod 600 /etc/rsyncd.secrets |

Запускаем и добавляем в автозапуск

|  |
| --- |
| [root@L-SRV]# systemctl start rsync.service [root@L-SRV]# systemctl enable rsync.service |

На клиенте ставим rsync

|  |
| --- |
| [root@L-CLI-B]# apt install rsync |

Создаем файл с паролем

|  |
| --- |
| [root@L-CLI-B]# vim /root/rp |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| parol666 |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оставляем права только руту

|  |
| --- |
| [root@L-CLI-B]# chmod 600 /root/rp |

Пробуем синхронизироваться

|  |
| --- |
| [root@L-CLI-B]# rsync -av --delete --password-file=/root/rp \  rsync://sync@l-srv.skill39.wsr/wsr/ \  /root/sync |

На L-CLI-A

|  |
| --- |
| [root@L-CLI-A]# apt install rsync [root@L-CLI-A]# mkdir /root/sync [root@L-CLI-A]# echo "parol666" > /root/rp [root@L-CLI-A]# chmod 600 /root/rp |

Тестим

|  |
| --- |
| [root@L-CLI-A]# rsync -av --delete --password-file=/root/rp \  /root/sync \  rsync://sync@l-srv.skill39.wsr/wsr/ |

Если, работает, автоматизируем клиентов

|  |
| --- |
| [root@L-CLI-A]# vim /root/sync.sh |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| #!/bin/bash  rsync -av --delete \  --password-file=/root/rp \  /root/sync \  rsync://sync@l-srv.skill39.wsr/wsr/ &> /dev/null |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| [root@L-CLI-A]# chmod a+x /root/sync.sh [root@L-CLI-A]# crontab -e |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| \* \* \* \* \* /root/sync.sh |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| [root@L-CLI-B]# vim /root/sync.sh |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| #!/bin/bash  rsync -av \  --password-file=/root/rp \  rsync://sync@l-srv.skill39.wsr/wsr/ \  /root/sync &> /dev/null |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| [root@L-CLI-B]# chmod a+x /root/sync.sh [root@L-CLI-B]# crontab -e |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

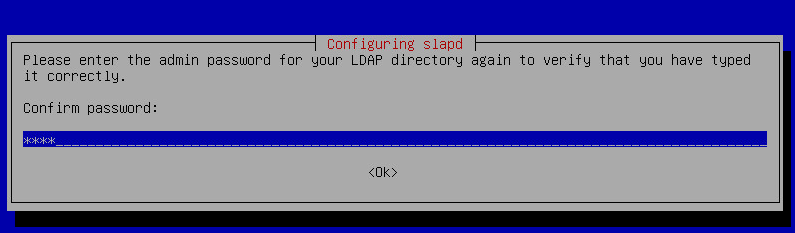
|  |
| --- |
| \* \* \* \* \* /root/sync.sh |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## OpenLDAP (1.5)

### LDAP: Users, Groups and OU (0.5)

|  |
| --- |
| [root@L-SRV]# apt install slapd ldap-utils  [root@L-SRV]# systemctl enable slapd |



Вызываем конфигуратор OpenLDAP

|  |
| --- |
| [root@L-SRV]# dpkg-reconfigure slapd |

Ставим скрипты для инициализации и добавления пользователей и групп

|  |
| --- |
| [root@L-SRV]# apt install ldapscripts |

Конфигурируем скрипты:

|  |
| --- |
| [root@L-SRV]# vim /etc/ldapscripts/ldapscripts.conf |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| SERVER="ldap://localhost"  SUFFIX="dc=skill39,dc=wsr" GSUFFIX="ou=Groups" USUFFIX="ou=Users" #MSUFFIX="ou=Machines" # < Комментим  BINDDN="cn=admin,dc=skill39,dc=wsr" BINDPWDFILE="/etc/ldapscripts/ldapscripts.passwd" |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Пароль от “cn=admin,dc=skill39,dc=wsr”:

|  |
| --- |
| [root@L-SRV]# echo **-n** "toor" > /etc/ldapscripts/ldapscripts.passwd |

Инициализируем (создаём OU и тд)

|  |
| --- |
| [root@L-SRV]# ldapinit |

Добавляем группы

|  |
| --- |
| [root@L-SRV]# ldapaddgroup Admin [root@L-SRV]# ldapaddgroup Guest |

Добавляем пользователя “tux” в группе Admin и задаем ему нужный пароль:

|  |
| --- |
| [root@L-SRV]# ldapadduser tux Admin [root@L-SRV]# ldapsetpasswd tux toor |

Создаем обычных пользователей:

|  |
| --- |
| [root@L-SRV]# for idx in $(seq 1 99); do ldapadduser user$idx Guest; done |

Меняем им пароли:

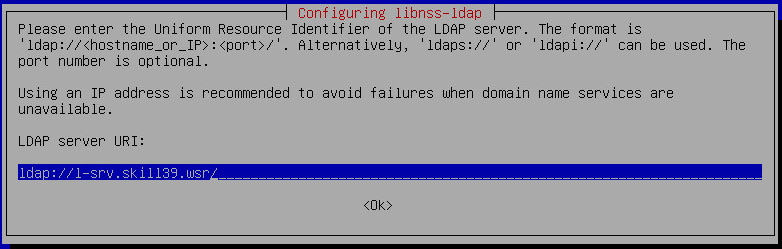
|  |
| --- |
| [root@L-SRV]# for idx in $(seq 1 99); do ldapsetpasswd user$idx P@ssw0rd; done |

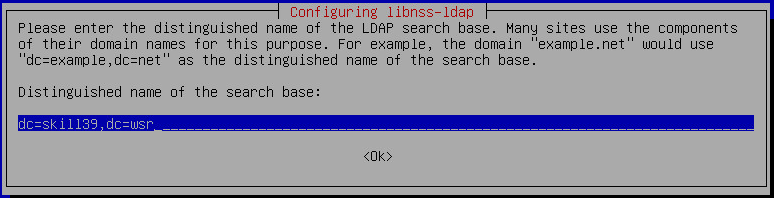
Убеждаемся что всё на месте:

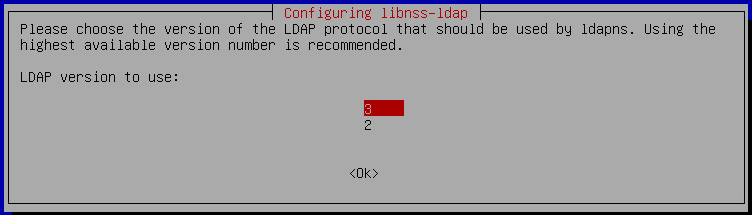
|  |
| --- |
| [root@L-SRV]# slapcat | less |

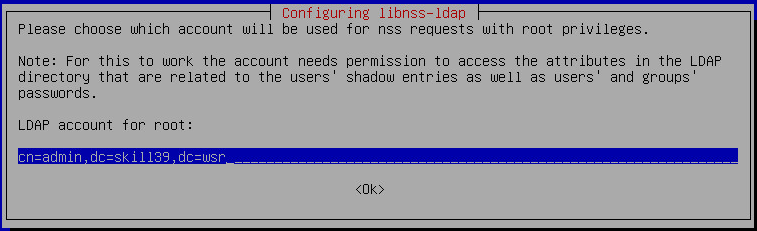
### LDAP: Clients authentication (0.5)

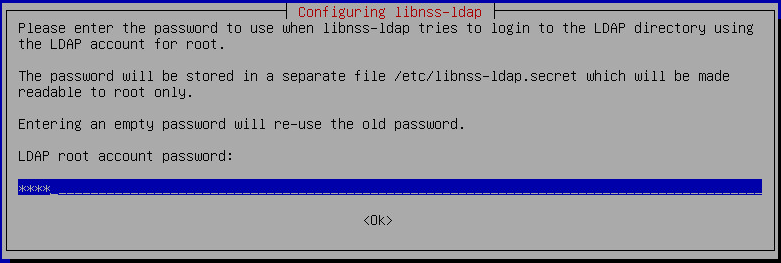
|  |
| --- |
| [root@L-CLI-ANY]# apt install libnss-ldap libpam-ldap |

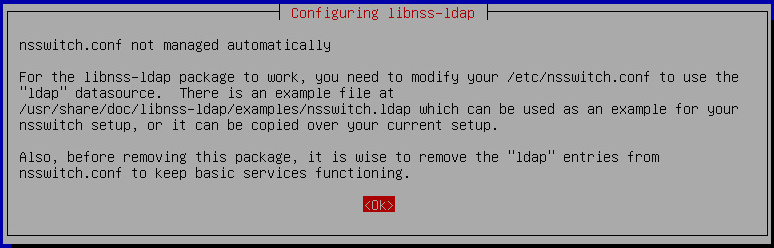


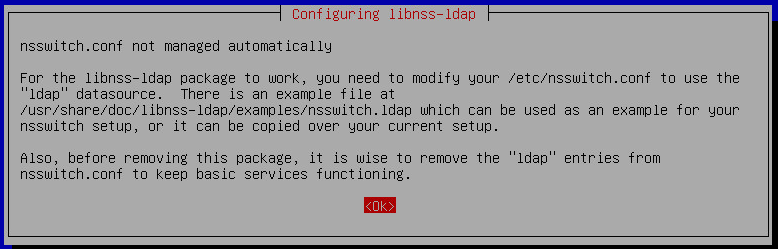


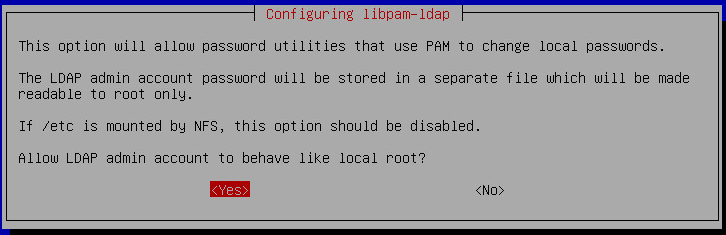


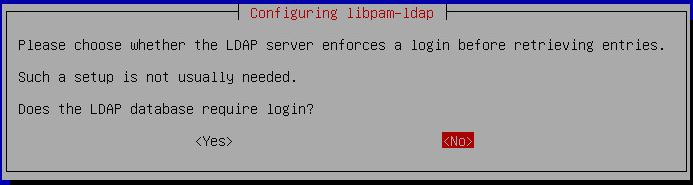


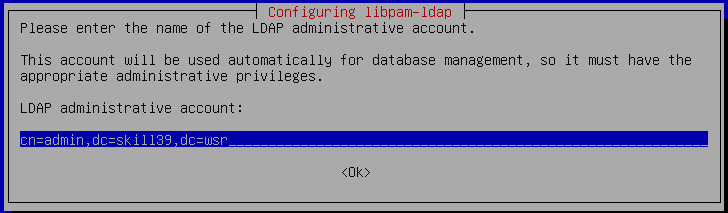


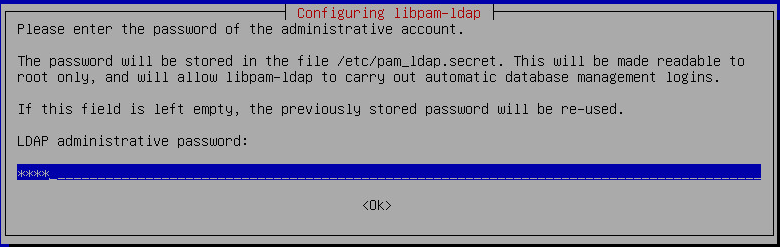




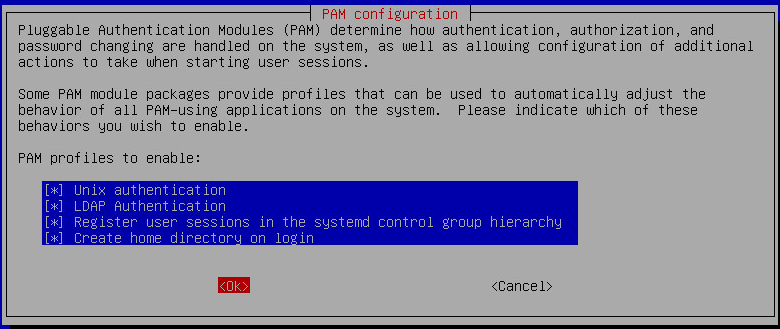








|  |
| --- |
| [root@L-CLI-ANY]# pam-auth-update |



Добавляем ldap в nss

|  |
| --- |
| [root@L-CLI-ANY]# vim /etc/nsswitch.conf |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

...

passwd: files systemd ldap

group: files systemd ldap

shadow: files ldap

gshadow: files ldap

...

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| [root@L-CLI-ANY]# reboot |

Вызвать TUI для данных пакетов повторно можно так:

|  |
| --- |
| [root@L-CLI-ANY]# dpkg-reconfigure libnss-ldap [root@L-CLI-ANY]# dpkg-reconfigure libpam-ldap [root@L-CLI-ANY]# pam-auth-update |

### LDAP: Logon restriction (0.5)

На L-SRV в “/etc/pam\_ldap.conf”:

Раскомментируем и поменяем значение pam\_filter, чтобы только пользователи из группы Admin могли заходить на сервер.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

...

pam\_filter gidNumber=10001

...

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

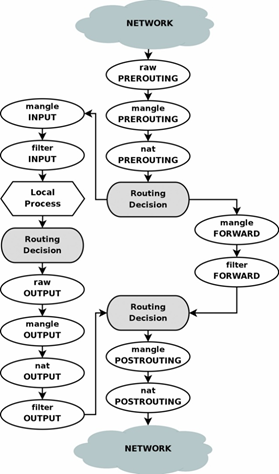
## Firewalls (1.8)

### IPtables

* [**Основные понятия iptables в Linux**](http://www.k-max.name/linux/netfilter-iptables-v-linux/)
* [**iptables в простейших примерах**](http://www.k-max.name/linux/iptables-v-primerax/)

**Netfilter** использует таблицы для организации правил. Эти **таблицы классифицируют правила согласно типу решения**, которое нужно принять. Каждая таблица разделяет правила на цепочки. Грубо говоря, **цепочки определяют, когда сработают данные правила**.

* Таблица Filter служит для принятия решения о том, стоит позволить пакету продолжить его движение, или отвергнуть его.
* Таблица Raw работает до подсистемы conntrack.
* Таблица NAT используется для реализации трансляции сетевых адресов.
* Таблица Mangle используется для изменения заголовков IP и маркировки пакетов.



Каждое правило содержит **критерий отбора** и **действие,** которое нужно выполнить.

* Критерии
  + Критерии срабатывания L3
    - -i/--in-interface - Входящий интерфейс
    - -o/--out-interface - Исходящий интерфейс
    - -s/--source - Адрес/сеть источника
    - -d/--destination - Адрес/сеть назначения
    - -p/--protocol - Протокол (tcp,udp,icmp, ...)
  + Критерии срабатывания L4
    - --sport port[:port] - порт или диапазон портов источника
    - --dport port[:port] - порт или диапазон портов назначения
* Действия
  + ACCEPT - принять пакет
  + DROP – молча отбросить пакет
  + REJECT- Отбросить пакет и сообщить источнику по icmp
  + MASQUERADE - Source NAT
* Дополнительные модули
  + conntrack/state - критерии срабатывания основанные на состоянии соединения

Подсистема CONNTRACK предоставляет важный механизм отслеживания соединений, на базе которого может происходить фильтрация трафика. Также, этот механизм жизненно необходим для NAT преобразований.

Соединения, отслеживаемые подсистемой CONNTRACK, могут находится в следующих состояниях:

* **NEW**: Новоприбывший пакет, не принадлежащий ни к одному из существующих соединений.
* **ESTABLISHED**: Соединение меняет своё состояние с NEW на ESTABLISHED, когда был получен валидный ответ в обратном направлении. Для TCP это значит, что был получен SYN/ACK, для UDP и ICMP, это значит, что был получен ответный пакет.
* **RELATED**: Пакеты, которые не принадлежат к текущим соединениям, но каким-то образом с ними связаны. Например, ICMP ответы на попытки соединения другими протоколами, или дополнительные соединения FTP.

Всё есть в мане!

|  |
| --- |
| [root@L-FW]# man -k iptables [root@L-FW]# man iptables [root@L-FW]# man iptables-extensions |

Смотрим добавленные правила в таблицу filter и таблицу nat

|  |
| --- |
| [root@L-FW]# iptables -S [root@L-FW]# iptables -t nat -S |

Смотрим расширенный вывод по правилам со статистикой

|  |
| --- |
| [root@L-FW]# iptables -L -v -n --line-numbers [root@L-FW]# iptables -t nat -L -v -n --line-numbers |

Есть возможность редактировать “/etc/iptables/rules.v4” напрямую и загружать правила в фаервол.

|  |
| --- |
| [root@L-FW]# vim /etc/iptables/rules.v4 [root@L-FW]# iptables-restore < /etc/iptables/rules.v4 |

Учтите, что сокращения, которые легко воспринимает команда iptables могут не восприниматься. Например, ‘-j DNAT --to’ разворачивается в ‘-j DNAT --to-destination’.

Если совсем отшибло память то используйте ман, а также у openvpn есть скрипт конфигурации фаервола "/usr/share/doc/openvpn/examples/sample-config-files/firewall.sh", из которого можно взять примеры команд.

**Внимание! Некоторые из правил уже были добавлены походу выполнения задания! Не нужно их добавлять повторно! Порядок - важен! Изменить его можно засейвившись в rules.v4 и перезагрузив правила с iptables-restore.**

**Если НЕ УВЕРЕН, то не делай REJECT, либо не сохраняй блокирующие правила! Иначе отрубишь себе всё что было сделано ДО.**

****

|  |
| --- |
| # Необходимо выполнить нат в момент,  # когда уже было принято решение о маршрутизации  # и пакет вот вот выходит из интерфейса ens160.  # Делаем PAT with overload.  [root@L-FW]# iptables **-t nat** \  **-A POSTROUTING** \  **-o ens160** \  **-j MASQUERADE** |

|  |
| --- |
| # Необходимо выполнить нат в момент, когда  # - еще не было принято решение о маршрутизации  # - и пакет только пришел на интерфейс ens160.  # Пакет должен содержать:  # - сегмент протокола UDP  # - идущий на порт 53 (DNS)  # Выполнить:  # - подмену адреса назначения на 172.16.20.10  [root@L-FW]# iptables **-t nat** \  **-A PREROUTING** \  **-i ens160 -p udp --dport 53** \  **-j DNAT --to 172.16.20.10:53** |

|  |
| --- |
| # Входящий **(предназначенный L-FW)** на интерфейс ens160  # трафик по протоколу TCP на порт 22 - Разрешить  [root@L-FW]# iptables **-A INPUT** \  **-i ens160 -p tcp --dport 22** \  **-j ACCEPT** |

|  |
| --- |
| # Входящий на интерфейс ens160  # трафик по протоколу UDP на порт 1122 - Разрешить  [root@L-FW]# iptables **-A INPUT** \  **-i ens160 -p udp --dport 1122** \  **-j ACCEPT** |

|  |
| --- |
| # Входящий на интерфейс ens160 трафик GRE - Разрешить  [root@L-FW]# iptables **-A INPUT** \  **-i ens160 -p gre** \  **-j ACCEPT**  # Входящий на интерфейс ens160 трафик ESP и ISAKMP - Разрешить  [root@L-FW]# iptables **-A INPUT** \  **-i ens160 -p esp** \  **-j ACCEPT** [root@L-FW]# iptables **-A INPUT** \  **-i ens160 -p udp --dport 500** \  **-j ACCEPT** [root@L-FW]# iptables **-A INPUT** \  **-i ens160 -p udp --dport 4500** \  **-j ACCEPT** |

|  |
| --- |
| [root@L-FW]# iptables **-A INPUT** \  **-i ens160** \  **-m conntrack --ctstate RELATED,ESTABLISHED** \  **-j ACCEPT**  # Запретить предназначенный L-FW трафик пришедший на интерфейс ens160  [root@L-FW]# iptables **-A INPUT** \  **-i ens160** \  **-j REJECT** |

Теперь запретим всё остальное!

|  |
| --- |
| # Запретить предназначенный L-FW трафик пришедший на интерфейс ens160  [root@L-FW]# iptables **-A INPUT** \  **-i ens160** \  **-j REJECT** |

|  |
| --- |
| # Если пакет предназначенный не L-FW принадлежит или относится  # к уже установленному соединению, значит его нужно пропустить.  [root@L-FW]# iptables **-A FORWARD** \  **-m conntrack --ctstate ESTABLISHED,RELATED \**  **-j ACCEPT**  [root@L-FW]# iptables **-A FORWARD** \  **-p udp --dport 53\**  **-j ACCEPT**  # Если пакет пришел на интерфейс ens160 и предназначен не L-FW,  # то его необходимо убить и отправить похоронку отправителю [root@L-FW]# iptables **-A FORWARD** \  **-i ens160** \  **-j REJECT**  # Если пакет не относится к уже установленному соединению  # и пришел с любого другого интерфейса, то его нужно пропустить  [root@L-FW]# iptables **-A FORWARD** **-j ACCEPT** |

Если всё, работает, то сохраняемся.

|  |
| --- |
| [root@L-FW]# iptables-save > /etc/iptables/rules.v4 |

### Firewalld

|  |
| --- |
| # Разрешаем сервисы ssh, http, https для зоны external  [root@R-FW]# firewall-cmd --zone=external \  --add-service=ssh \  --permanent [root@R-FW]# firewall-cmd --zone=external \  --add-service=http \  --permanent [root@R-FW]# firewall-cmd --zone=external \  --add-service=https \  --permanent |

|  |
| --- |
| # Добавляем icmp по вкусу  [root@R-FW]# firewall-cmd --zone=external \  --add-protocol=icmp \  --permanent |

|  |
| --- |
| # Разрешаем ESP и IKE [root@R-FW]# firewall-cmd --zone=external \  --add-service=ipsec \  --permanent |

|  |
| --- |
| Запрещаем всё остальное  [root@R-FW]# firewall-cmd --zone=external \  --set-target=REJECT \  --permanent |

|  |
| --- |
| [root@R-FW]# firewall-cmd --reload |

## Thats’ all!