[**Pre-requirements**](#_iroxirgyeig) **2**

[**Конфигурация хостов**](#_3yes53wucxbw) **5**

[Установка ПО на ВСЕ виртуальные машины](#_mllg3gbxoyxq) 7

[Если все диски в комплекте](#_s9morgiq6o4u) 7

[Настройка имен хостов](#_kj7seexwxlzr) 8

[Формирование файла hosts](#_pdqiebx4pnp9) 8

[**Конфигурация сетевой инфраструктуры**](#_athbp375lmll) **9**

[Конфигурация сетевых интерфейсов](#_89kv1zwyibvq) 9

[Форвардинг пакетов](#_4sr88dtqwwup) 14

[NTP](#_r046i0m00iis) 15

[DHCP Server](#_76x7fk9zkleu) 15

[DNS Server](#_quubt386pbop) 17

[DHCP + DNS](#_2m9myxuh5m8) 20

[Трансляция сетевых адресов](#_1nk3dxce0amt) 22

[**Конфигурация служб удаленного доступа**](#_a79pwlyb2bam) **29**

[GRE](#_unmrrw3y8s2c) 29

[OpenVPN](#_55f2idhnrbug) 31

[Plaintext tunnel](#_4cppa12zi8ab) 31

[Static Key/Keys](#_vlvj5bq4w09) 32

[С использованием OpenSSL сертификатов](#_j26a435wl3qz) 33

[Настройка SSH](#_5z7hea1iu63u) 37

[**Конфигурация веб служб**](#_hqwcorv1ej9e) **39**

[**Конфигурация служб хранения данных**](#_gcyx6w9sby1u) **40**

[NFS](#_aqdaqzf147nl) 40

[FTP](#_qd0f6top4zco) 41

[**Службы централизованного управления и журналирования**](#_ht0ysj4hqzn6) **42**

[Logging](#_uz8ciex8k3jy) 42

[OpenLDAP](#_cfs7mzjnc459) 46

[Установка OpenLDAP](#_9ednu7ftsc7b) 46

[Создаем пользователей и группы](#_7pzqnx3vl7fq) 48

[Подключаем клиентские машины](#_5x1ozzrya19c) 50

[Debian (на примере L-FW)](#_g5pntlmfu6sh) 50

[CentOS (на примере R-FW) (Если вдруг понадобится, мало ли)](#_5wog5xt03zbb) 54

[**Конфигурация параметров безопасности и служб аутентификации**](#_h5j5iocdijp7) **55**

[Настройка CA](#_heqmv2ecsy5n) 55

[Настройка межсетевых экранов на L-FW и R-FW](#_e6ribcxsxa39) 56

[**Критерии и Оценка задания**](#_5dgxhqpowxso) **58**

[**На всякий случай**](#_a5vyovvh2lkz) **61**

[Если вам предоставлен только Debian DVD-1](#_so5vqilr4q66) 61

[Если на диске вам предоставили RPM пакет easy-rsa версии 3](#_20ylwsyif9np) 62

[Если использовать firewalld на R-FW](#_m1mlnvavpvu1) 64

## Pre-requirements

**Умение пользоваться man!!!!** Базовые навыки работы с простыми командами типа: ls, mv, cp, rm и т.д. Умение пользоваться редактором vi ( vim ).

Научится можно воспользовавшись программой vimtutor и

<https://www.shortcutfoo.com/app/dojos/vim/learn>

Мой стенд и AdditionalPackages.iso доступны тут: <https://yadi.sk/d/SFqhMISfJzTfJg>

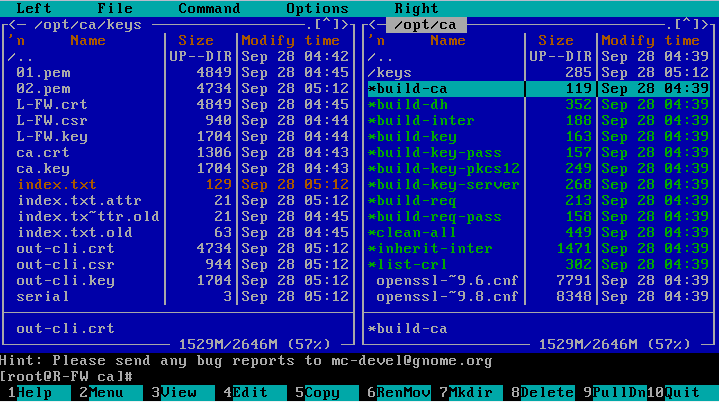
**Внимание! Был предоставлен стенд и диски от разработчиков задания! Используйте его! Там есть всё что необходимо. Задание будет переработано с учетом последних обновлений.**

Для копирования файлов между машинами нужно, чтобы был установлен sshd.

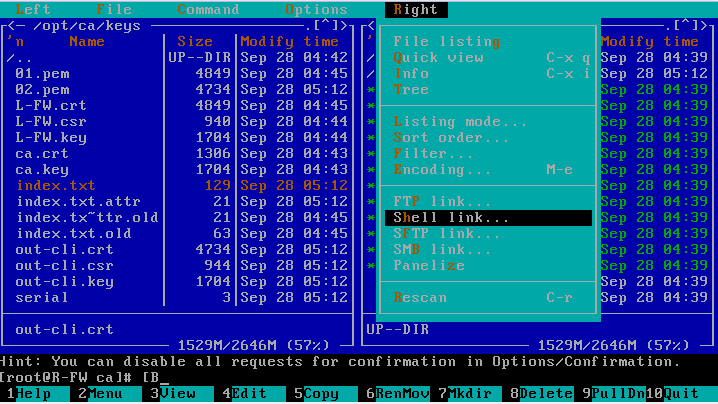
|  |
| --- |
| [root@host]# apt install openssh-server [root@host]# vim /etc/ssh/sshd\_config |

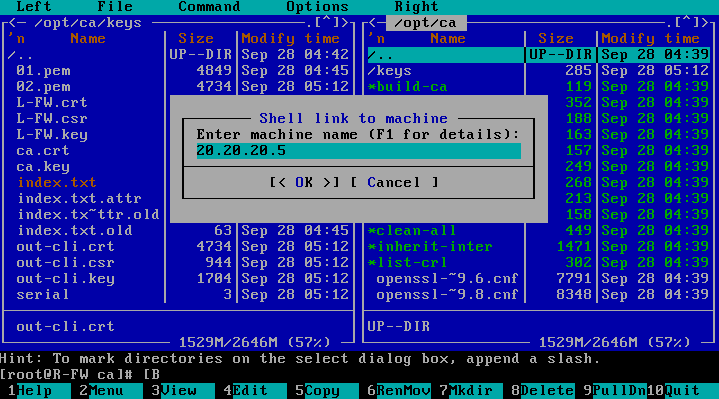
Должна быть установлена опция “PermitRootLogin yes”.

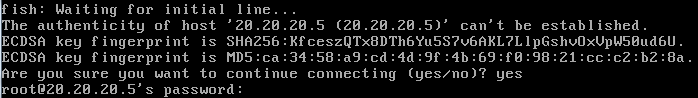
Для удобного копирования файлов можно воспользоваться утилитой **mc**. В прохождении я использую только scp, но для вашего удобства можно рассмотреть этот вариант. Если не настроен DNS и он имеет приоритет в резолвинге, то SSH будет безбожно лагать!

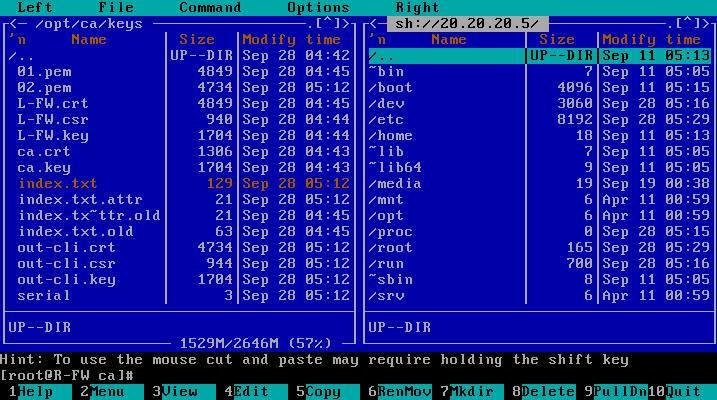


F9

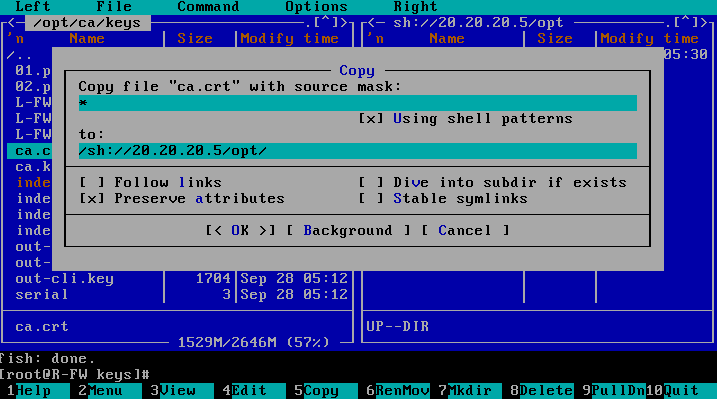






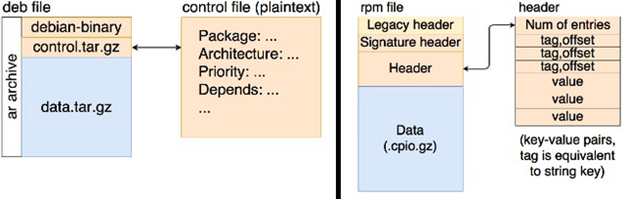


Выбираем файл, жмём F5.



## Конфигурация хостов

Программное обеспечение в мире Linux представляется в виде пакетов, представляющих собой обычный архив, содержащих само программное обеспечение, набор определённых метаданных, которые могут включать в себя: полное имя пакета, номер версии, описание пакета, контрольную сумму, зависимости от других пакетов, список пакетов с которыми конфликтует данный пакет.



Установкой/удалением пакетов и поддержанием базы данных о них занимаются утилиты dpkg (в Debian-like системах) и rpm (в RedHat-like системах). Важно: Данные утилиты не разрешают зависимости и конфликты.

Команды, которые нужно знать для выполнения задания:

|  |  |  |
| --- | --- | --- |
|  | **Debian** | **RedHat** |
| Установить пакет: | dpkg -i ./pkg.deb | rpm -ivh ./pkg.rpm |
| Список установленных пакетов: | dpkg -l | rpm -qa |
| Список файлов, установленных пакетом: | dpkg -L base-passwd | rpm -ql base-passwd |
| Какому пакету принадлежит файл: | dpkg -S /bin/date | rpm -qf /bin/date |

Пакеты собирают в "репозитории", которые хранят метаинформацию о пакетах и сами пакеты. Репозитории могут располагаться на CD/DVD дисках, сетевых шарах, и специальных серверах-хранилищах.

Для работы с репозиториями существуют такие утилиты как apt (в Debian-like системах) и yum (в RedHat-like системах). Данные утилиты позволяют синхронизировать данные о доступном программном обеспечении, выполнять разрешение зависимостей и установку пакетов через соответствующие backend утилиты.

Команды, которые нужно выучить:

|  |  |  |
| --- | --- | --- |
|  | **Debian** | **RedHat** |
| Обновить информацию о доступных пакетах | apt update | yum check-update |
| Обновить доступные пакеты: | apt upgrade | yum update |
| Найти пакет в репозиториях: | apt search pkg\_name | yum search pkg\_name |
| Установить пакет (и его зависимости): | apt install pkg\_name | yum install pkg\_name |
| Удалить пакет из системы: | apt purge pkg\_name | yum erase pkg\_name |

Установка ПО с DVD в Debian

Посмотреть список подключенных репозиториев “/etc/apt/sources.list”. Разъяснения о формате файла sources.list можно найти в мануале “man sources.list”.

Подключить диск и добавить как репозиторий можно командой “apt-cdrom add”. Debian предоставляется в виде набора из нескольких DVD дисков, поэтому “apt-cdrom add” нужно проделать над каждым диском. Утилита apt-cdrom сама монтирует диск и добавляет запись о нём в sources.list.

Установка ПО с DVD в CentOS

Посмотреть список подключенных репозиториев и их состояние можно командой:

yum repolist all

Файлы конфигурации репозиториев хранятся в директории “/etc/yum.repo.d/”. Репозиторий будет являться “disabled”, если у него в конфигурации есть строка “enabled=0”. Чтобы не возится со всеми файлами и записями в них, предлагаю, перенести все файлы в резервную директорию и оставить в “yum.repo.d”, только один файл “CentOS-Media.repo”.

1. Создаём директорию для резервного хранения конфигурации по умолчанию   
   # mkdir ~/repo
2. Переносим все файлы из “yum.repo.d” в новую директорию  
   # mv /etc/yum.repo.d/CentOS-\* repos/
3. Копируем файл отвечающий за поиск репозиториев на внешних носителях обратно  
   # cp repos/CentOS-Media.repo /etc/yum.repos.d/
4. Включаем данный репозиторий изменив “enabled=0” на “enabled=1”  
   # vi /etc/yum.repos.d/CentOS-Media.repo
5. В файле “CentOS-Media.repo” было указано где должны находится точки монтирования, поэтому создаём директорию, куда будет смонтирован DVD  
   # mkdir /media/cdrom
6. Вставляем DVD диск и монтируем его в созданную директорию  
   # mount /dev/cdrom /media/cdrom
7. Проверяем доступность репозитория  
   # yum check-update

### Установка ПО на ВСЕ виртуальные машины

#### Если все диски в комплекте

Debian:

|  |
| --- |
| cat /etc/apt/sources.list apt-cdrom add  apt install -y tcpdump net-tools vim lynx dnsutils ftp lftp apt install bash-completion mc |

CentOS:

|  |
| --- |
| mkdir ~/repos mv /etc/yum.repos.d/Cent\* ~/repos cp ~/repos/CentOS-Media.repo /etc/yum.repos.d/ vi /etc/yum.repos.d/CentOS-Media.repo # enabled=1 mkdir /media/cdrom mount /dev/cdrom /media/cdrom  yum install -y tcpdump net-tools vim lynx bind-utils ftp lftp yum install -y bash-completion mc |

Пакеты bash-completion и mc - необязательны, но весьма удобны. Для того что бы авто дополнение заработало, нужно выйти “exit” и залогиниться снова.

### Настройка имен хостов

Следующее, что нам необходимо - сменить имя машин. Это можно сделать, записав его в файл “/etc/hostname” и перезагрузив машину. Обратите внимание на регистр. “L-SRV” и “l-srv” - это разные имена.

|  |
| --- |
| vim /etc/hostname reboot |

### Формирование файла hosts

Про резолвинг в Linux:

* <https://habr.com/post/352300/>
* <https://searchitchannel.techtarget.com/feature/Using-nsswitchconf-to-find-Linux-system-information>

Файл hosts используется как база данных для хранения доменных имен и трансляции их в сетевые адреса. Историю файла hosts должны знать все, кто хоть что-то слышал о DNS.

Формат файла: {ip-address} [fqdn] {hostname}. Если забыли, то можно всегда найти в man.

|  |
| --- |
| man hosts vim /etc/hosts |

Пример hosts для l-srv:

127.0.0.1 l-srv.skill39.wsr l-srv

10.10.10.10 isp.skill39.wsr isp

172.16.20.1 l-fw.skill39.wsr l-fw

20.20.20.5 out-cli.skill39.wsr out-cli

20.20.20.100 r-fw.skill39.wsr r-fw

Для разрешения доменных имен стандартным делом для OS будет посмотреть сначала в hosts, и если соответствий не было найдено послать запрос dns серверу. Порядок разрешения доменных имен, портов, сервисов определяется в файле “/etc/nsswitch”.

|  |
| --- |
| vim /etc/nsswitch.conf |

Меняем порядок с “files dns” на “dns files”

...

hosts dns files

...

## Конфигурация сетевой инфраструктуры

### Конфигурация сетевых интерфейсов

В современных дистрибутивах для манипуляции сетевыми устройствами присутствует пакет iproute, в состав которого входят такие утилиты как: ip, ss, tc, и т.д. Минимально необходимые команды, которые нужно знать:

* Посмотреть список физических интерфейсов  
  # ip link ( или ip l)
* Посмотреть список интерфейсов с IP-адресами  
  # ip address show (или ip a s)
* Добавить IP адрес на интерфейс  
  # ip address add {ip-address/mask} dev {interface}
* Удалить IP адрес с интерфейса  
  # ip address del {ip-address/mask} dev {interface}
* Посмотреть список маршрутов  
  # ip route show (или ip r s)
* Добавить маршрут  
  # ip route add {network/mask} via {gateway ip}
* Удалить маршрут  
  # ip route del {network/mask} via {gateway ip}
* Добавить маршрут по умолчанию  
  # ip route add default via {gateway ip}
* Удалить маршрут по умолчанию  
  # ip route del default

Данные команды применяются на лету, но не переживают ребут системы. Чтобы изменения переживали ребут, существует несколько способов.

**Способ номер #1** - Network Scripts и файлы конфигурации.

В предыдущих релизах RedHat, основным способом конфигурации сети было использование shell скриптов. Под шел скриптами подразумевается в основном “/etc/init.d/network” и всё что он в дальнейшем вызывает. Пользователь предоставлял файлы, рассматриваемые как конфигурационные, но которые могли быть интерпретируемые как дополнения, содержащие переменные для скриптов. В настоящее время, данный способ является дополнительным, в то время как в Debian, этот способ всё ещё является способом по умолчанию.

В RedHat-подобных дистрибутивах, конфигурационные файлы интерфейсов хранятся в директории “/etc/sysconfig/network-scripts/” с именами “ifcfg-\*”, где “\*” – имя интерфейса.

Создать файл конфигурации можно обычным текстовым редактором

|  |
| --- |
| vim /etc/sysconfig/network-scripts/ifcfg-enp0s3 |

Узнать, что нужно написать в этом файле можно в файле документации “sysconfig.txt”:

|  |
| --- |
| less /usr/share/doc/initscripts-version/sysconfig.txt |

Посмотреть примеры файлов конфигурации сети можно в директории

|  |
| --- |
| ls /usr/share/doc/initscripts-version/examples/networking/ |

Не нужно запоминать эти пути! Нужно помнить, что их всегда можно найти. Логика следующая:

1. Нужно узнать какому пакету принадлежит “/etc/sysconfig/network-scripts/”  
   # rpm -qf /etc/sysconfig/network-scripts/
2. Вывести список всех файлов, установленных данным пакетом  
   # rpm -ql initscripts или rpm -ql $(rpm -qf /etc/sysconfig/network-scripts/)
3. Отфильтровать вывод можно с помощью команды grep  
   # rpm -ql initscripts | grep doc

Пример конфигурации статики:

DEVICE=enp0s3

BOOTPROTO=none

ONBOOT=yes

IPADDR=192.168.0.1

NETMASK=255.255.255.0

GATEWAY=192.168.0.254

DNS1=192.168.0.254

USERCTL=no

Чтобы задать статический маршрут/маршруты, нужно создать файл “/etc/sysconfig/network-scripts/route-\*”, где “\*” – имя исходящего интерфейса. В данном файле, нужно перечислить необходимые маршруты. Пример: “172.16.2.0/24 via 192.168.0.254”.

После выполнения вышеперечисленных действий нужно сделать: systemctl restart networking.

В Debian-подобных файл конфигурации сетевых интерфейсов находится тут “/etc/network/interfaces”. Информацию о том, как заполнять этот файл можно найти в мануле

|  |
| --- |
| man interfaces vim /etc/network/interfaces |

auto enp0s3

allow-hotplug enp0s3

iface enp0s3 inet static

address 192.168.0.6

netmask 255.255.254.0

gateway 192.168.1.237

Как добавить статические маршруты? iface поддерживает следующие опции (что конечно же описано в man interfaces): pre-up, up, post-up, pre-down, down. Данные опции позволяют выполнить любую команду, перед тем как интерфейс поднимется или выключится. Поэтому…

auto enp0s3

allow-hotplug enp0s3

iface enp0s3 inet static

address 192.168.0.6

netmask 255.255.254.0

gateway 192.168.1.237

post-up ip route add 172.16.2.0/24 via 192.168.0.254

pre-down ip route del 172.16.2.0/24 via 192.168.0.254

DNS-сервера для статически настроенных машин должны быть перечислены в файле “/etc/resolv.conf” в формате “nameserver X.X.X.X”. Данный файл может быть перезаписан, например - dhcp-client’ом, если есть интерфейсы, получающие IP-адрес динамически, поэтому как-то нужно убедиться, что данные настройки будут перманентными.

В Debian настроить через interfaces просто так не получится. Для этого нужно установить пакет “resolvconf”.

|  |
| --- |
| apt install resolvconf |

После этого в “man resolvconf” можно подсмотреть, что необходимые опции можно добавить так: “dns-nameservers 192.168.0.254 8.8.8.8”.

auto enp0s3

allow-hotplug enp0s3

iface enp0s3 inet static

address 192.168.0.6

netmask 255.255.254.0

gateway 192.168.1.237

dns-nameservers 192.168.0.254 8.8.8.8

dns-search mycorp.un

**Способ номер #2** - NetworkManager (Уневерсальный!!!)

Более современным путём настройки сети будет являться NetworkManager. NetworkManager – дефолтный (в RHEL) сетевой демон, который позволяет настраивать IP-адреса, создавать алиасы интерфейсов, добавлять статические маршруты, информацию о DNS, создавать VPN подключения. NetworkManager запускается как демон и предоставляет API через шину D-Bus, который предоставляет способ запрашивать и контролировать состояние устройств.

В стандартную поставку для NM входят следующие утилиты:

|  |  |
| --- | --- |
| nmtui | Простой псевдо-графический интерфейс. |
| nmcli | Command-line утилита, позволяющая пользователям и скриптам взаимодействовать с NetworkManager’ом. |
| nm-connection-editor | GTK+ приложение. |

Хоть в RHEL NetworkManager и является сетевым сервисом по умолчанию, разработчики сделали всё возможное чтобы shell-скрипты (network scripts) могли взаимодействовать с NM и работать параллельно. NM запускается первым, а /etc/init.d/network согласуется с NM что бы избежать вмешательства в его конфигурацию.

Что касается Debian, то там NetworkManager требует отдельной установки.

**Внимание! На машинах с GUI network-manager уже должен быть установлен. Так как апплеты конфигурации сети его и используют.**

|  |
| --- |
| apt install network-manager |

Конфигурация в “/etc/network/interfaces” не должна конфликтовать с NM, поэтому там должны остаться только настройки loopback интерфейса.

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

**# The loopback network interface**

**auto lo**

**iface lo inet loopback**

Дальше необходимо выполнить перезагрузку Debian-машин:

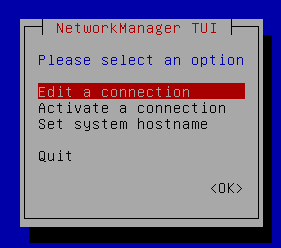
|  |
| --- |
| reboot |

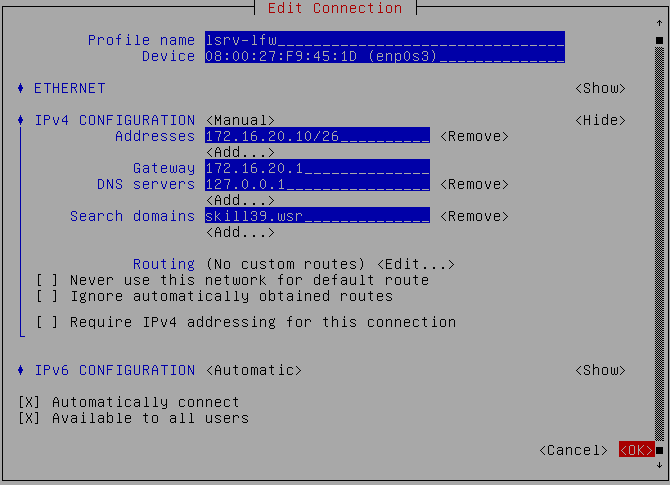
После чего можно будет воспользоваться утилитой nmtui для настройки:

* ip адресов
* DNS
* hostnames
* статических маршрутов
* gre-туннелей

|  |
| --- |
| nmtui |

Можно редактировать соединения, активировать/деактивировать и задать хостнейм.





Выполните схожие действия для всех остальных машин!

NetworkManager - имба! Используйте его!!!!

### Форвардинг пакетов

Некоторые участники сети будут выступать в качестве маршрутизаторов, поэтому нужно включить возможность форвардинга.

Просматривать и настраивать системные переменные, можно с помощью утилиты sysctl.

Найти опцию, которая отвечает за форвардинг ipv4 пакетов

|  |
| --- |
| sysctl -a | grep forward | grep ip |

Включить форвардинг на работающей системе

|  |
| --- |
| sysctl -w net.ipv4.ip\_forward=1 |

Для того чтобы изменения сохранялись при перезапуске системы, “net.ipv4.ip\_forward=1” нужно добавить в файл “/etc/sysctl.conf”.

На L-FW и R-FW

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

### NTP

Для нормальной работы многих сервисов (ОСОБЕННО с TLS), необходимо чтобы часы были синхронизированы.

Установка NTP сервера на L-SRV

|  |
| --- |
| [root@L-SRV]# apt install chrony [root@L-SRV]# echo "allow 0.0.0.0/0" >> /etc/chrony/chrony.conf  [root@L-SRV]# echo "local stratum 2" >> /etc/chrony/chrony.conf [root@L-SRV]# timedatectl set-timezone Asia/Yekateringburg [root@L-SRV]# timedatectl set-ntp 0 [root@L-SRV]# systemctl restart chrony |

Подключение клиентов.

|  |
| --- |
| [root@ANY]# timedatectl set-timezone Asia/Yekateringburg [root@ANY]# timedatectl set-ntp 1 [root@L-ANY]# apt install chrony  [root@L-ANY]# vim /etc/chrony/chrony.conf  [root@R-ANY]# vim /etc/chrony.conf  [root@ANY]# systemctl restart chronyd |

### DHCP Server

- DHCP server на Linux:

<http://www.k-max.name/linux/dhcp-server-na-linux/>

Устанавливаем DHCP сервер и делаем backup конфига.

|  |
| --- |
| [root@L-FW]# apt install isc-dhcp-server [root@L-FW]# mkdir dhcp\_bak [root@L-FW]# cp /etc/dhcp/dhcpd.conf dhcp\_bak/ |

Хорошим тоном будет убедится, что isc-dhcp-server был добавлен в автозагрузку.

|  |
| --- |
| [root@L-FW]# systemctl enable isc-dhcp-server |

Нужно указать на каких интерфейсах должен работать isc-dhcp-server.

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

INTERFACESv4="enp0s8"

После чего можно приступить к конфигурации. В dhcpd.conf уже все что необходимо есть, нужно просто отсечь всё ненужное.

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

# Суффикс доменного имени по умолчанию

option domain-name "skill39.wsr";

# Адрес/адреса DNS сервера/серверов

option domain-name-servers 172.16.20.10;

# Время аренды адреса по умолчанию

default-lease-time 600;

# Ограничение по максимальному времени аренды

max-lease-time 7200;

# Данный сервер является авторитетным для этой сети

authoritative;

# Отключить DDNS. (Опция понадобиться в будущем)

ddns-update-style none;

subnet 172.16.100.0 netmask 255.255.255.0 {

range 172.16.100.60 172.16.100.75; # Диапазон адресов

option routers 172.16.100.1; # Шлюз по умолчанию

}

Проверить конфигурацию DHCP сервера

|  |
| --- |
| [root@L-FW]# dhcpd -t |

Если с конфигурацией всё ОК, то dhcp-сервер нужно перезапустить:

|  |
| --- |
| [root@L-FW]# systemctl restart isc-dhcp-server |

### DNS Server

DNS - <https://www.youtube.com/watch?v=IKXzWNT2RTo>

DNS сервер BIND (теория/практика)

* <http://www.k-max.name/linux/dns-server-bind/>
* <http://www.k-max.name/linux/howto-dns-server-bind/>

Связка DNS + DHCP

<https://www.youtube.com/watch?v=OzD70xRsltc>

<https://www.youtube.com/watch?v=Asc47EUrCgg>

<https://www.youtube.com/watch?v=YdmaZc2L8vE>

<https://www.youtube.com/watch?v=3iSb-GR9Ylg>

<https://www.youtube.com/watch?v=9iz_jPeO9rs>

Установить пакет bind.

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

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

// Запросы, которые выходят за рамки зоны skill39.wsr

// должны пересылаться DNS-серверу ISP.

forwarders { 10.10.10.10; };

// Слушать на всех интерфейсах

listen-on { any; };

// Разрешаем все виды запросов

allow-query { any; };

// Отключить DNSSEC

dnssec-validation no;

Теперь всё что нужно сделать - это скопировать стандартные зоны, которые идут вместе с пакетом bind9. Эти файлы мы впоследствии отредактируем.

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

Расскажем bind о новых зонах. В файле “named.conf.default-zones” есть примеры описания стандартных зон.

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

Наши зоны будут выглядеть следующим образом.

zone "skill39.wsr"{

type master;

allow-transfer { 10.10.10.10; };

file "/opt/dns/db.wsr";

};

zone "16.172.in-addr.arpa"{

type master;

allow-transfer { 10.10.10.10; };

file "/opt/dns/db.172";

};

Далее, нам необходимо описать наши зоны и добавить записи в соответствии с таблицей в задании.

|  |
| --- |
| [root@L-SRV]# vim /opt/dns/db.wsr |

; Определяет «время жизни» (Time-To-Live) для конкретной записи

$TTL 86400

; @ - Имя зоны, подставляется из zone "skill39.wsr" {};

; В данном случае @ будет "skill39.wsr"

; IN - Класс. IN - интернет.

; skill39.wsr. - Первичный (Primary) DNS-сервер для этой зоны. DNS-сервер, ; на котором хранится полная исходная информация

; об этой зоне.

;root.skill39.wsr. - Адрес электронной почты админа. Поскольку символ "@" уже

; используется, email записывается в таком формате.

@ IN SOA skill39.wsr. root.skill39.wsr. (

2018060801 ; <- Серийный номер - версия файла зоны

; Как правило - год + месяц + день + версия за этот день

1h ; <- Как часто вторичные серверы должны запрашивать обновления

20m ; <- Если первичный сервер недоступен, то как долго ждать

; прежде чем повторить попытку.

1w ; <- Как долго вторичный сервер может использовать ранее

; полученные данные, если не удается обновить зону.

86400 ) ; <- Как долго хранить негативные ответы

NS skill39.wsr. ; DNS-сервер для домена @

A 172.16.20.10 ; IPv4 адрес для @

l-srv IN A 172.16.20.10

dns IN CNAME l-srv.skill39.wsr.

l-fw IN A 10.10.10.1

vpn IN CNAME l-fw.skill39.wsr.

ftp IN CNAME l-fw.skill39.wsr.

r-fw IN A 20.20.20.100

www IN CNAME r-fw.skill39.wsr.

out-cli IN A 20.20.20.5

|  |
| --- |
| [root@L-SRV]# vim /opt/dns/db.172 |

$TTL 86400

@ IN SOA skill39.wsr. root.skill39.wsr. (

2018060801

5m

5m

1w

86400 )

NS skill39.wsr.

10.20 IN PTR skill39.wsr.

10.20 IN PTR l-srv.skill39.wsr.

Проверяем конфигурацию, запускаем bind и добавляем его в автозагрузку.

|  |
| --- |
| [root@L-SRV]# named-checkconf -z [root@L-SRV]# systemctl start bind9 [root@L-SRV]# systemctl enable bind9 |

### 

### DHCP + DNS

**Warning!** Убедитесь,что время на L-SRV и L-FW совпадает!

Как настроить связку dhcp + dns ? Это в примерах описано в мануале к файлу “dhcpd.conf”. Всё что нужно сделать - переписать и подставить свои значения!



Ключ rndc генерируется при установке пакета и хранится либо в “/etc”, либо в “/etc/bind” (В зависимости от дистрибутива).

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

**include "/etc/bind/rndc.key";**

// Подключим содержимое файла "rndc-key" к конфигурации

// К нашим зонам добавляем allow-update { key rndc-key; };

zone "skill39.wsr"{

type master;

allow-transfer { 10.10.10.10; };

**allow-update { key rndc-key; };**

file "/opt/dns/db.wsr";

};

zone "16.172.in-addr.arpa"{

type master;

allow-transfer { 10.10.10.10; };

**allow-update { key rndc-key; };**

file "/opt/dns/db.172";

};

Копируем ключ на L-FW

|  |
| --- |
| [root@L-SRV]# scp /etc/bind/rndc-key root@172.16.20.1:/etc/dhcp |

На L-FW добавляем этот ключ в конфигурационный файл isc-dhcp-server

|  |
| --- |
| [root@L-FW]# cat /etc/rndc.key >> /etc/dhcp/dhcpd.conf |

После чего редактируем dhcpd.conf.

...

**ddns-update-style interim;**

...

zone skill39.wsr. {

primary 172.16.20.10;

key rndc-key;

}

zone 16.172.in-addr.arpa. {

primary 172.16.20.10;

key rndc-key;

}

### Трансляция сетевых адресов

**Netfilter, iptables and Firewalld**

Чтобы понять, как ядро обрабатывает пакеты и как на это можно влиять, нужно прочитать данные статьи:

* <http://www.k-max.name/linux/netfilter-iptables-v-linux/>
* <http://www.k-max.name/linux/iptables-v-primerax/>
* <https://www.digitalocean.com/community/tutorials/a-deep-dive-into-iptables-and-netfilter-architecture>

Эти статьи нужно прочитать столько раз, сколько потребуется для того чтобы осознать, понять, принять и простить packet-flow ядра linux.

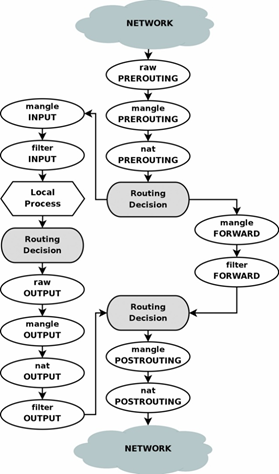
Кратенько:

**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 обеспечивает механизм определения состояния соединений.
* 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.

Действия бывают терминирующими и не терминирующими

.

Подробней о синтаксисе можно узнать в “**man iptables**”.

iptables [-t table] -A chain [matches...] [target]

iptables [-t table] {-I|-R} chain {rulenum} [matches...] [target]

iptables [-t table] -D rulenum

iptables [-t table] -P chain target

* 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

Таблица RAW 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.

Да, да, да… С точки зрения системы CONNTRACK, UDP и ICMP тоже устанавливают соединения

.

Подробней об этом и о других возможных состояниях соединений можно узнать в “**man iptables-extentions**”. Как туда попасть, если забыл? Можно подсмотреть в “**man iptables**” в разделе “**SEE ALSO**”.

Пример использования:

|  |
| --- |
| iptables -A FORWARD -s 172.16.0.0/16 -j ACCEPT iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT iptables -P DROP FORWARD |

Все эти правила будут жить до первой перезагрузки, поэтому нужно как-то позаботиться об их сохранении и загрузке.

**Сохранение правил - Вариант #1**

Выгрузить в текстовый файл и загрузить состояние правил можно с помощью следующих команд.

|  |
| --- |
| iptables-save > /root/rules.v4 iptables-restore < /root/rules.v4 |

Как можно догадаться, эти строки можно поместить в скрипт “/etc/rc.local”.

Следствие. Редактировать правила и их очередность, можно редактируя файл “rules.v4”, и загружая изменения через “iptables-restore”.

**Сохранение правил - Вариант #2**

Для Debian можно установить службу “iptables-persistent”.

|  |
| --- |
| apt-get install iptables-persistent |

Добавляем в автозагрузку

|  |
| --- |
| systemctl enable iptables-persistent |

Сохранение состояния фаервола и перезапуск:

|  |
| --- |
| netfilter-persistent save netfilter-persistent reload |

Как дела обстоят в CentOS?

В предыдущих версиях существовал демон iptables, который мог загружать правила из файла “/etc/sysconfig/iptables”.

Соответственно, сохранение правил происходило следующим образом.

|  |
| --- |
| iptables-save | tee /etc/sysconfig/iptables systemctl restart iptables |

Как дела обстоят сейчас? На место демона iptables пришел firewalld.

**Firewalld**

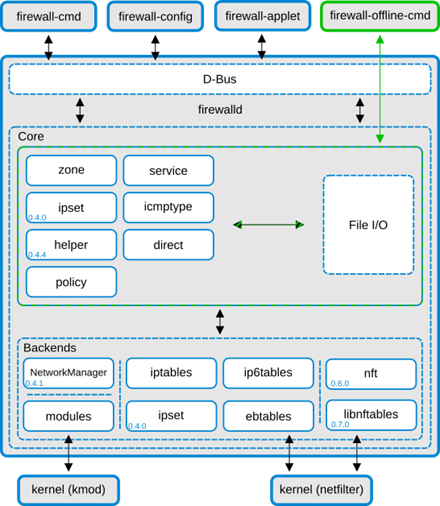
Разбираться как работает Netfilter – сложно. Разбираться как писать правила iptables – сложно. Читать чужие правила, разбираться в них и обновлять – ещё сложнее. Неудивительно, что человеки начали биться над решением проблемы - как всё упростить, особенно для тех кому – “мне бы только 22 порт заблокировать и всё”. Были придуманы ufw (Uncomplicated Firewall), FERM (for Easy Rule Making), но всё не то. И вот, Fedora 18 явила миру firewalld, который стал официальным приложением для манипуляции настройками netfilter в RedHat-подобных.

Firewalld, как и его товарищи, является надстройкой (фронтендом) над всеми известными утилитами.

Firewalld обладает рядом преимуществ.

* Работа с D-BUS API
* Умеет в бриджи и ipset
* Поддерживает работу с NAT
* Является zone-based фаерволом
* Простые описания сервисов, портов, протоколов и прочего
* Обладает богатым языком для гибких и сложных правил (В реальности сложнее и страшнее iptables)
* Простое журналирование отброшенных пакетов
* Интеграция с Puppets

Архитектура:



Команды, которые нужно запомнить:

Получить список зон:

|  |
| --- |
| firewall-cmd --get-zones |

Узнать зону по умолчанию:

|  |
| --- |
| firewall-cmd --get-default-zone |

Сделать public зоной по умолчанию:

|  |
| --- |
| firewall-cmd --set-default-zone=public |

Вывести настройки зоны public:

|  |
| --- |
| firewall-cmd --zone=public --list-all |

Получить список известных сервисов:

|  |
| --- |
| firewall-cmd --get-services |

Добавить разрешение для сервиса http в зоне public:

|  |
| --- |
| firewall-cmd --zone=public --add-service=http --permanent |

Если не указывать зону, то будет использована зона по умолчанию:

|  |
| --- |
| firewall-cmd --permanent --add-port=666/tcp |

Добавить сетевой интерфейс в зону:

|  |
| --- |
| firewall-cmd --permanent --zone=trusted --add-interface=enp0s3 |

Создать описание нового сервиса:

|  |
| --- |
| firewall-cmd --permanent --new-service=NameOfTheService firewall-cmd --permanent --service=name-service --add-port=1234/tcp |

Задать политику по умолчанию для зоны:

|  |
| --- |
| firewall-cmd --permanent --zone=public --set-target=DROP |

Перечитать правила:

|  |
| --- |
| firewall-cmd --reload |

\* --permanent — добавить постоянное правило (будет действовать после перезагрузки)

**Настраиваем NAT**

На L-FW выполняем команды.

|  |
| --- |
| [root@L-FW]# iptables -A INPUT -s 172.16.0.0/16 -j ACCEPT [root@L-FW]# iptables -A INPUT -s 172.16.0.0/16 -j FORWARD [root@L-FW]# iptables -t nat -A POSTROUTING -o enp0s9 -j MASQUERADE |

Зачем так много? Ведь для нашей цели хватило бы и последней. Нам они понадобятся в дальнейшем.

Теперь, поскольку ISP является slave-сервером для L-SRV, служба DNS должна быть доступна через L-FW, для того чтобы можно было совершать трансфер зон, поэтому нужно выполнить следующие команды. Обратите внимание, что должны быть проброшены 53 udp и tcp порты.

|  |
| --- |
| [root@L-FW]# iptables -t nat -A PREROUTING -i enp0s9 -p udp --dport 53 -j DNAT --to 172.16.20.10 [root@L-FW]# iptables -t nat -A PREROUTING -i enp0s9 -p tcp --dport 53 -j DNAT --to 172.16.20.10 |

Устанавливаем iptables-persistent. Во время установки он предложит сохранить действующие правила.

|  |
| --- |
| [root@L-FW]# apt search ^iptables  [root@L-FW]# apt install iptables-persistent [root@L-FW]# systemctl enable iptables-persistent  [root@L-FW]# systemctl status iptables-persistent |

Теперь правила можно вносить через “/etc/iptables/rules.v4”.

На R-FW:

Отключим firewalld и установим iptables-services

|  |
| --- |
| [root@R-FW]# systemctl stop firewalld [root@R-FW]# systemctl disable firewalld  [root@R-FW]# yum search iptables  [root@R-FW]# yum install iptables-services  [root@R-FW]# systemctl start iptables  [root@R-FW]# systemctl enable iptables |

Добавляем и сохраняем правила для трансляции адресов. А так же комментируем всё ненужное на данном этапе в “/etc/sysconfig/iptables”.

|  |
| --- |
| [root@R-FW]# iptables -t nat -A POSTROUTING -o ens160 -j MASQUERADE [root@R-FW]# iptables-save > /etc/sysconfig/iptables  [root@R-FW]# vim /etc/sysconfig/iptables  [root@R-FW]# systemctl restart iptables |

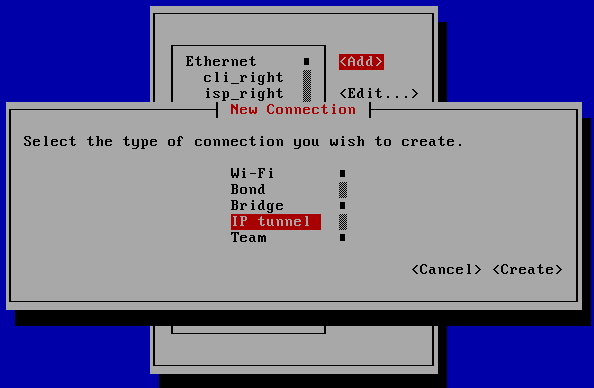
**Не удаляйте дефолтные правила из “/etc/syconfig/iptables”! Комментируйте! Они вам пригодятся!**

## Конфигурация служб удаленного доступа

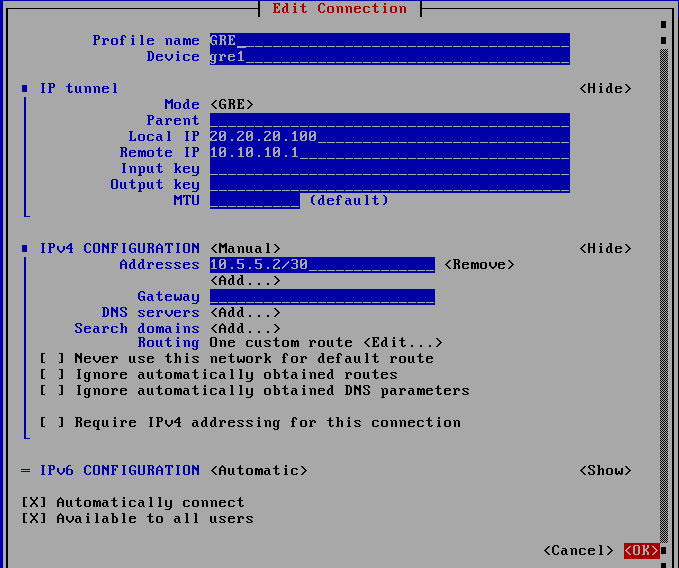
### GRE

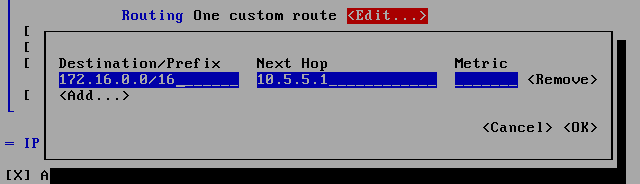
На R-FW:

Добавим сетевой интерфейс через nmtui.

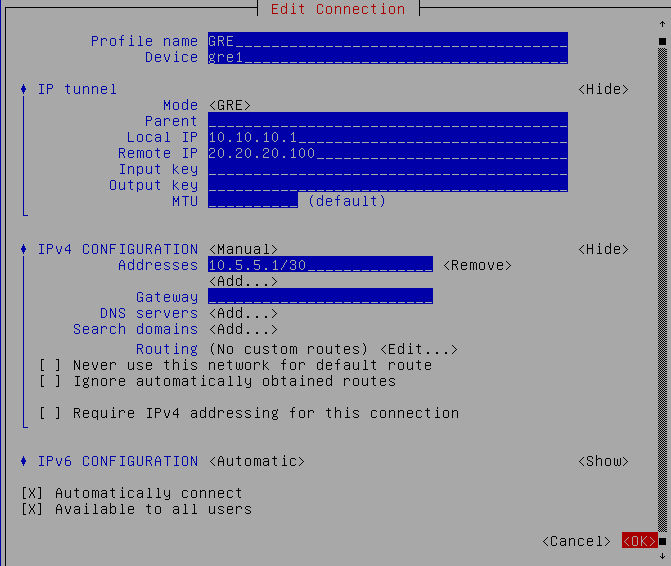


Обратите внимание, что интерфейс называется “**gre1**”, а не “gre0”.





Аналогичные действия проделываем на L-FW:



**Не забывайте добавлять маршруты до внутренних сеток!**

### OpenVPN

OpenVPN может работать в нескольких режимах::

* Без аутентификации.
* Static Key - Аутентификация с использованием pre-shared ключей
* TLS - С использованием SSL/TLS сертификатов.

#### Plaintext tunnel

|  |
| --- |
| server# openvpn --ifconfig 10.200.0.1 10.200.0.2 --dev tun --auth none client# openvpn --ifconfig 10.200.0.2 10.200.0.1 --dev tun --auth none --remote example.com |

Absolutely no encryption is performed. All of the traffic that is sent over the tunnel is encapsulated in an OpenVPN packet and then sent as is.

#### Static Key/Keys

Аутентификация с pre-shared ключом.

|  |
| --- |
| server# openvpn --ifconfig 10.200.0.1 10.200.0.2 --dev tun --secret secret.key client# openvpn --ifconfig 10.200.0.2 10.200.0.1 --dev tun --secret secret.key --remote example.com |

Ключ может быть сгенерирован командой

|  |
| --- |
| openvpn --genkey --secret /path/to/secret.key |

OpenVPN uses two symmetric keys when setting up a point-to-point connection. However, it is also possible to use shared yet asymmetric keys in point-to-point mode. OpenVPN will use four keys in this case:

* A cipher key on the client side
* An HMAC key on the client side
* A cipher key on the server side
* An HMAC key on the server side

The same keying material is shared by both sides of the point-to-point connection, but the keys that are derived for encrypting and signing the data are different for each side.

|  |
| --- |
| server# openvpn --ifconfig 10.200.0.1 10.200.0.2 --dev tun --secret secret.key 0 --verb 7 client# openvpn --ifconfig 10.200.0.2 10.200.0.1 --dev tun --secret secret.key 1 --remote example.com --verb 7 |

OpenVPN получает все четыре ключа из файла static.key file.

Настройка через конфигурационные файлы.

server:

dev tun

proto udp

port 1194

secret secret.key 0

ifconfig 10.200.0.1 10.200.0.2

client:

dev tun

proto udp

remote example.com

port 1194

secret secret.key 1

ifconfig 10.200.0.2 10.200.0.1

route 192.168.4.0 255.255.255.0

#### С использованием OpenSSL сертификатов

В принципе, первые два способа нас не особо интересуют, но ими можно воспользоваться, если отшибло память на тему как сгенерировать сертификаты.

|  |
| --- |
| [root@R-FW]# yum install /media/cdrom/easy-rsa-2.\*.rpm |

Узнаём, что скрипты были установлены в “/usr/share/easy-rsa/2.0”.

|  |
| --- |
| [root@R-FW]# rpm -ql easy-rsa |

Скопируйте содержимое в /opt/ca.

|  |
| --- |
| [root@R-FW]# cp -R /usr/share/easy-rsa/2.0/\* /opt/ca |

После чего нужно редактировать файл с переменными “vars”. Там нельзя задать пустые переменные.

Меняем

export KEY\_COUNTRY=”RU”

export KEY\_ORG=”WorldSkills Russia”

остальное оставляем как есть. Во второй версии в vars нельзя ставить точки чтобы оставить поле пустым, это нужно будет делать во время создания ключей.

После чего можно импортировать vars, после чего вам будет предложено инициализировать директорию для ключей.

|  |
| --- |
| [root@R-FW]# cd /opt/ca [root@R-FW]# source ./vars [root@R-FW]# ./clean-all |

После чего нужно сгенерировать Диффи Хеллмана и ключ CA. Диффи Хеллман на виртуальной машине будет генерироваться довольно медленно.

|  |
| --- |
| [root@R-FW]# ./build-dh [root@R-FW]# ./build-ca |

Отвечаем на вопросы build-ca, жмём либо Enter, либо ставим точку, если поле должно быть пустым.

Country Name: RU

State or Province Name: **.**

Locality Name: **.**

Organization Name: WorldSkills Russia

Organizational Unit Name: **.**

Common Name: WSR CA

Name: **.**

Email Address: **.**

Ставим точки там, где поле должно быть пустым.

Затем нужно создать ключи для сервера и клиента. Заметьте, что во второй версии easy-rsa не нужно делать запросы и подписывать их. Он всё сделает сам.

|  |
| --- |
| [root@R-FW]# ./build-key-server L-FW [root@R-FW]# ./build-key OUT-CLI |

Проверить созданные ключи можно командой:

|  |
| --- |
| [root@R-FW]# openssl x509 -in keys/ca.crt -text |

Issuer - кто выпустил сертификат

Subject - сам сертификат

У CA они будут совпадать. На остальных Issuer - это CA.

dh.pem, ca.crt, l-fw.crt и l-fw.key нужно скопировать на L-FW

ca.crt, out-cli.crt и out-cli.key нужно скопировать на OUT-CLI.

Убедитесь, что все поля были заполнены согласно задания!

Устанавливаем OpenVPN, если этого еще не сделали.

|  |
| --- |
| [root@L-FW]# apt install openvpn  [root@OUT-CLI]# mount /dev/cdrom /media [root@OUT-CLI]# yum install /media/openvpn-\* /media/pkcs\* |

Смотрим что было установлено и какие файлы у нас есть

|  |
| --- |
| [root@L-FW]# dpkg -L openvpn  [root@OUT-CLI]# rpm -ql openvpn |

Примеры конфигурационных файлов можно найти в

/usr/share/doc/openvpn/

Копируем необходимое в /etc/openvpn

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

Открываем файлы конфигурации и отсекаем всё ненужное

|  |
| --- |
| [root@L-FW]# vim /etc/openvpn/server.conf  [root@OUT-CLI]# vim /etc/openvpn/server.conf |

server:

# **Не используйте local ip-address**, иначе демон не будет

# запускаться при старте системы!

port 1122

proto udp

dev tun

dh /opt/vpn/dh.pem # Use the Diffie-Hellman key

ca /opt/vpn/ca.crt # Use the CA certificate

cert /opt/vpn/L-FW.crt # Signed certificate

key /opt/vpn/L-FW.key # Private RSA key

server 5.5.5.0 255.255.255.224

ifconfig-pool-persist ipp.txt

push “route 172.16.0.0 255.255.0.0”

push “route 192.168.100.0 255.255.255.0”

keepalive 10 120

cipher AES-256-CBC

user nobody

group nogroup

persist-key

persist-tun

status /var/log/openvpn-status.log

log /var/log/openvpn.log

log-append /var/log/openvpn.log

verb 3

explicit-exit-notify 1

client:

client

dev tun

proto udp

remote 10.10.10.1 1122

resolv-retry infinite

nobind

user nobody

group nogroup

persist-key

persist-tun

ca /opt/vpn/ca.crt # Use the CA certificate

cert /opt/vpn/OUT-CLI.crt # Signed certificate

key /opt/vpn/OUT-CLI.key # Private RSA key

remote-cert-tls server

Запускаем, добавляем в автозагрузку

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

Создаем скрипты для запуска на OUT-CLI

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

#!/bin/sh

systemctl start openvpn@client

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

#!/bin/sh

systemctl stop openvpn@client

Разрешаем выполнение скриптов.

|  |
| --- |
| [root@OUT-CLI]# chmod +x /opt/vpn/\*.sh |

Добавляем /opt/vpn в переменную PATH.

|  |
| --- |
| [root@OUT-CLI]# vim /etc/profile.d/vpn.sh |

PATH=$PATH:/opt/vpn

**Внимание! Синхронизируйте часы на всех виртуалках.** Хотя бы командой date.

### 

### Настройка SSH

Создаем пользователей и задаем пароли

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

Меняем порт и разрешаем только определенных пользователей

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

...

port 1022

AllowUsers ssh\_p ssh\_c

...

На OUT-CLI доступ к серверу L-FW должен происходить автоматически по правильному порту, без его явного указания номера порта в команде подключения. Для других серверов по умолчанию должен использоваться порт 22.

|  |
| --- |
| [root@OUT-CLI]# vim /etc/ssh/ssh\_config |

...

Host 10.10.10.1

Port 1022

Host l-fw

Port 1022

Host l-fw.skill39.wsr

Port 1022

Host vpn.skill39.wsr

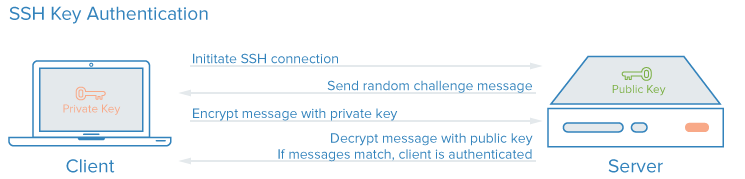
Port 1022

Host ftp.skill39.wsr

Port 1022

...

Доступ по ключу без ввода пароля



Доступ к L-FW под учетной записью ssh\_p должен происходить с помощью аутентификации на основе открытых ключей.

Генерируем пару public/private ключей. Копируем публичный ключ на L-FW.

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

## Конфигурация веб служб

|  |
| --- |
| [root@R-FW]# yum install httpd mod\_ssl |

|  |
| --- |
| [root@R-FW]# mkdir /var/www/html/out [root@R-FW]# echo skill39.wsr > /var/www/html/out/index.html |

Меняем "/var/www/html" на "/var/www/html/out"

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

Генерируем ключи для SSL

|  |
| --- |
| [root@R-FW]# ./easyrsa gen-req www.skill39.wsr nopass [root@R-FW]# ./easyrsa sign-req server www.skill39.wsr |

Меняем пути к ключам

|  |
| --- |
| [root@R-FW]# vim /etc/httpd/conf.d/ssl.conf |

Добавляем правило редиректа с http на https

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

RewriteEngine On

RewriteRule (.\*) https://%{HTTP\_HOST}%{REQUEST\_URI}

Добавляем ключ нашего CA в доверенные на всех клиентских машинах

Debian

|  |
| --- |
| dpkg -l | grep cert  dpkg -L ca-certificates  man update-ca-certificates  cp ca.crt /usr/local/share/ca-certificates/ update-ca-certificates |

CentOS

|  |
| --- |
| rpm -qa | grep cert  rpm -ql ca-certificates  man update-ca-trust  cp ca.crt /etc/pki/ca-trust/source/anchors update-ca-trust |

## Конфигурация служб хранения данных

### NFS

Устанавливаем nfs-server.

|  |
| --- |
| [root@L-SRV]# apt install nfs-kernel-server |

Убеждаемся, что он попал в автозапуск.

|  |
| --- |
| [root@L-SRV]# systemctl enable nfs-server |

Редактируем файл exports, и добавляем разрешения для клиентов. Этот файл имеет следующий формат.

**адрес\_директории** **клиент1**(опции) ... **клиентN**(опции)

Обратите внимание, что между клиентом и опциями нет пробела.

|  |
| --- |
| [root@L-SRV]# mkdir /opt/nfs  [root@L-SRV]# chmod 777 /opt/nfs  [root@L-SRV]# vim /etc/exports |

/opt/nfs L-CLI(rw,sync,no\_subtree\_check) R-CLI(rw,sync,no\_subtree\_check)

|  |
| --- |
| [root@L-SRV]# systemctl restart nfs-server |

В задании сказано, что клиенты L-CLI и R-CLI должны автоматически монтировать шару при загрузке системы. Поэтому нужно подправить файл fstab.

Для начала создадим необходимую директорию, затем подправим fstab.

|  |
| --- |
| [root@L-CLI]# apt install nfs-common  [root@L-CLI]# mkdir /opt/nfs [root@L-CLI]# vim /etc/fstab |

fstab имеет следующий формат:

Что монтировать? Куда монтировать? Какая файловая система? С какими опциями? Остальное неважно (просто два нуля).

l-srv.skill39.wsr:/opt/nfs /opt/nfs nfs defaults 0 0

### FTP

<https://interface31.ru/tech_it/2012/02/prostoy-ftp-server-na-baze-ubuntu.html>

Если при конфигурировании возникают проблемы, то сервер можно запускать командой “vsftpd”, он будет говорить, что ему не нравится. Также доступен мануал “man vsftpd.conf”

Устанавливаем vftpd

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

Редактируем конфиг.

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

Тут всё что было изменено и добавлено. Остальное оставляем как есть.

…

# Меняем NO на YES

listen=YES

# Отключаем ipv6. Оно по каким-то причинам конфликтует с listen

listen\_ipv6=NO

# Использовать локальных юзверей

local\_enable=YES

# Разрешить им заливать файлы

write\_enable=YES

# Права на созданные файлы по умолчанию

local\_umask=022

# Корень ftp сервера

local\_root=/opt/nfs

# Использовать файл userlist, но не блокировать юзеров из него, а всех остальных

userlist\_enable = YES

userlist\_deny = NO

...

Создаем пользователя

|  |
| --- |
| [root@L-SRV]# adduser ftpuser |

Добавляем ftpuser в список тех, кому можно

|  |
| --- |
| [root@L-SRV]# /etc/vsftpd.user\_list |

Права для всех и на всё

|  |
| --- |
| [root@L-SRV]# chmod 777 /opt/nfs |

Перезапускаем vsftpd

|  |
| --- |
| [root@L-SRV]# systemctl enable vsftpd [root@L-SRV]# systemctl restart vsftpd |

## Службы централизованного управления и журналирования

### Logging

* <http://www.k-max.name/linux/syslogd-and-logrotate/>
* <http://www.k-max.name/linux/rsyslog-na-debian-nastrojka-servera/>

Конфигурационный файл syslog состоит из правил. Каждое правило состоит из селектора и действия. Каждый селектор состоит из двух частей, разделенных точкой:   
**источник.важность действие**.

Syslog известны следующие источники (категории): kern, user, mail, daemon, auth, syslog, lpr, news, uucp, cron, authpriv, ftp, ntp, local0 - local7.

С local0 по local7 - зарезервированные категории для использования администратором системы.

Syslog известны следующие (степени важности) приоритеты (по увеличению важности):

- 7 - debug,

- 6 - info,

- 5 - notice,

- 4 - warning = warn,

- 3 - err = error,

- 2 - crit,

- 1 - alert,

- 0 - emerg = panic.

Ключевые слова warn, error и panic считаются устаревшими - используйте вместо них warning, err и emerg.

При указании степени приоритета учитываются также все степени, находящиеся в иерархии выше. Иными словами, селектор mail.err охватывает и все сообщения почтовой системы уровней crit, alert и emerg. Если вам требуются уведомления только с определенной степенью приоритета, поставьте перед этой степенью символ = (то есть, например, mail.=err).

В качестве действия чаще всего указывается имя одного из файлов регистрации.

Обычно файлы регистрации синхронизируются после каждого вывода. Если перед именем файла стоит символ -, Syslog не производит синхронизацию файла после каждой записи. Syslog также может переадресовывать уведомления к файлам FIFO или к именованным каналам (named pipe). В таком случае перед именем файла ставится символ |.

Вспомнить какие бывают источники и приоритеты можно в мануале к команде logger в разделе “FACILITIES AND LEVELS”.

|  |
| --- |
| [root@host]# man logger |

Logger умеет отправлять сообщения в syslog и journald и будет использоваться для отладки и проверки задания.

|  |
| --- |
| [root@host]# logger -p local7.crit Test Message Crit [root@host]# logger -p local7.err Test Message Err [root@host]# logger -p auth.info Test Message Auth |

Для того чтобы rsyslog на L-SRV мог принимать логи с других серверов, нужно в его конфиге раскомментировать следующие строки:

...

# provides UDP syslog reception

module(load=”imudp”)

input(type=”imudp” port=”514”)

...

После чего нужно перезапустить rsyslog и убедиться, что он начал слушать нужный порт.

|  |
| --- |
| [root@L-SRV]# systemctl restart rsyslog [root@L-SRV]# ss -lu |

На клиенте необходимо настройку логов по сети. На примере L-FW.

...

# отправлять от любого источника (\*)

# с приоритетами (crit и err)

# по UDP на 172.16.20.10

\*.=crit @172.16.20.10

\*.=err @172.16.20.10

...

Обратите внимание, что я написал “\*.=crit”, это значит, что будут отсылаться только критические сообщения, сообщения с большим приоритетом отсылаться не будут.

Дело в том, что я не могу понять что от нас хотят, так как таблица и задание противоречат друг другу. Не понятно что написано в “Уровень журналирования”. Это язык syslog или нет? Если да то сообщения будут дублироваться в разных файлах, а по заданию не должны. Например уровень “\*.err” означает - err, crit, alert, emerg. Это значит, что всё что равно и выше приоритетом должно писаться в error.log. А “\*.crit” значит - crit, alert, emerg и он должен писаться в crit.log. Так что согласно таблице они должны пересекаться, а согласно заданию нет. Таже ситуация обстоит с “auth”. Все auth должны писаться в auth.log, согласно таблице. Но согласно заданию, нет, err там быть не должно.

Внятных объяснений никто не даёт. Так что хз. Нужно будет разбираться с тем чего они хотят на согласовании задания перед чемпионатом, поэтому я привожу общее решение, которое можно будет легко подкорректировать.

После чего, перезапускаем службу и убеждаемся что логи появляются на L-SRV, для этого на L-SRV запускаем tail.

|  |
| --- |
| [root@L-SRV]# tail -f /var/log/syslog |

на L-FW

|  |
| --- |
| [root@L-FW]# systemctl restart rsyslog [root@L-FW]# logger -p local7.crit Test L-FW Crit Messages  [root@L-FW]# logger -p local7.err Test L-FW Err Messages  [root@L-FW]# logger -p local7.info Test L-FW Info Messages |

На L-SRV вы должны увидеть сообщения, которые вы написали. Сообщения с приоритетом info пройти не должно.

Для того чтобы распихать логи по директориям необходимо использовать сислоговские переменные и RainerScript-based filters.

**if** **условие** **then** **блок\_действий** **else** **блок\_действий**

Список переменных, как всегда, можно найти в мануале к конфигурационному файлу.

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

В разделе “PROPERTY REPLACER” есть подраздел “Available Properties”.

Переменные, которые нас интересуют:

* $hostname - имя хоста
* $syslogfacility - источник в числовом формате
* $syslogfacility-text - источник в текстовом формате
* $syslogseverity - важность (приоритет) в числовом формате
* $syslogseverity-text - важность в текстовом формате

Пример того, как могут быть оформлены правила в конфиге rsyslog на L-SRV.

if ( $hostname == "L-SRV" and $syslogseverity == 2 ) then /opt/logs/L-SRV/crit.log

if ( $hostname == "L-FW" and $syslogseverity == 2 ) then /opt/logs/L-FW/crit.log

if ( $hostname == "L-SRV" and $syslogfacility-text == “auth” ) then /opt/logs/L-SRV/auth.log

if ( $hostname == "L-FW" and $syslogseverity-text == “err” ) then /opt/logs/L-FW/error.log

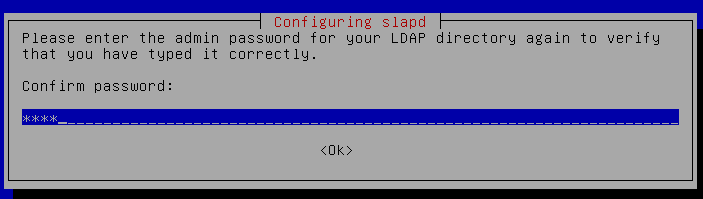
### OpenLDAP

#### Установка OpenLDAP

Устанавливаем openldap, задаём пароль и base DN.

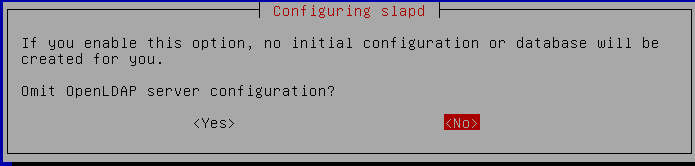
|  |
| --- |
| [root@L-SRV]# apt get install slapd ldap-utils |

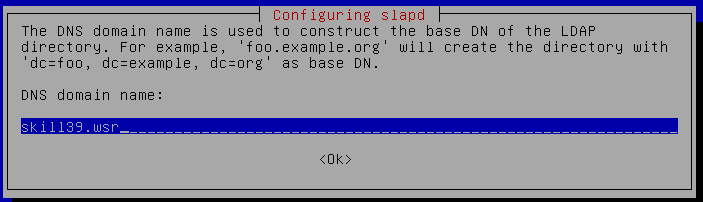
Во время установки будет предложено ввести пароль для администратора ldap.



После чего реконфигурируем пакет.

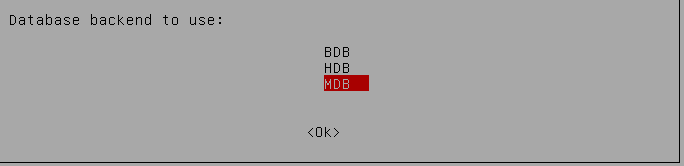
|  |
| --- |
| [root@L-SRV]# dpkg-reconfigure slapd |



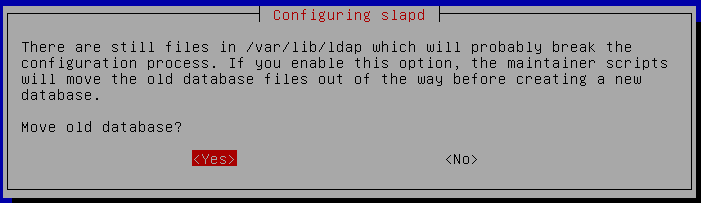












Проверяем конфигурацию! Найдем все записи в skill39.wsr от пользователя admin.

|  |
| --- |
| [root@L-SRV]# ldapsearch -D "cn=admin,dc=skill39,dc=wsr" -x -W -b "dc=skill39,dc=wsr" |

Сейчас у нас хранится только сам пользователь админ и описание нашей организации.

#### Создаем пользователей и группы

Для того чтобы руками не создавать пользователей и писать ldif файлы автоматизируем это действие на R-FW.

Нам понадобится пакет migrationtools, в котором находятся perl-скрипты, которые помогают перенести реальных пользователей из системы в ldap. Вывод данных скриптов производится в формате ldif файлов.

План такой:

1. Создаём нужные группы и нужных пользователей
2. Импортируем в ldif файлы
3. Удаляем группы и пользователей
4. Переносим ldif на L-SRV и импортируем в OpenLDAP

|  |
| --- |
| [root@R-FW]# yum install migrationtools [root@R-FW]# rpm -ql migrationtools |

Добавим группы Users и Administrators на R-FW

|  |
| --- |
| [root@R-FW]# groupadd Users [root@R-FW]# groupadd Administrators |

Создадим пользователя admin с паролем toor в группе Administrators.

|  |
| --- |
| [root@R-FW]# useradd -mg Administrators admin [root@R-FW]# passwd admin |

Создадим пользователей с user1 по user99 с паролем P@ssw0rd в группе Users.

|  |
| --- |
| [root@R-FW]# for i in {1..99}; do useradd -g Users user$i; done [root@R-FW]# for i in {1..99}; do echo "P@ssw0rd" | passwd --stdin user$i; done |

После чего перейдем в рабочую директорию migrationtools и зададим значения переменных.

|  |
| --- |
| [root@R-FW]# cd /usr/share/migrationtools [root@R-FW]# vim ./migrate\_common.ph |

Пользователи должны быть в **ou** “Users”, поэтому в секции else меняем “People” на “Users”, а группы в **ou** “Group” на “Groups”. Так же меняем base DN.

…

} else {

…

$NAMINGGCONTEXT{‘passwd’} = “ou=Users”;

…

$NAMINGGCONTEXT{‘group’} = “ou=Groups”;

…

}

…

$DEFAULT\_MAIL\_DOMAIN = “skill39.wsr”;

$DEFAULT\_BASE = “dc=skill39,dc=wsr”;

…

|  |
| --- |
| [root@R-FW migrationtools]# ./migrate\_base.pl > /root/base.ldif [root@R-FW migrationtools]# vim /root/base.ldif |

Оставляем записи про OU Users и Groups, остальное удаляем.

dn: ou=Users,dc=skill39,dc=wsr

ou: Users

objectClass: top

objectClass: organizationalUnit

dn: ou=Groups,dc=skill39,dc=wsr

ou: Groups

objectClass: top

objectClass: organizationalUnit

|  |
| --- |
| [root@R-FW migrationtools]# ./migrate\_group.pl /etc/group > /root/groups.ldif [root@R-FW migrationtools]# vim /root/groups.ldif |

Оставляем наши группы “Users” и “Administrators”, остальное удаляем.

dn: cn=Users,ou=Groups,dc=skill39,dc=wsr

objectClass: posixGroup

objectClass: top

ou: Users

userPassword: {crypt}x

gidNumber: 10000

dn: cn=Administrators,ou=Groups,dc=skill39,dc=wsr

objectClass: posixGroup

objectClass: top

ou: Administrators

userPassword: {crypt}x

gidNumber: 10001

|  |
| --- |
| [root@R-FW migrationtools]# grep user[1-99] /etc/passwd > /root/users\_source [root@R-FW migrationtools]# grep admin /etc/passwd >> /root/users\_source [root@R-FW migrationtools]# ./migrate\_passwd.pl /etc/users\_source > /root/users.ldif [root@R-FW migrationtools]# vim /root/users.ldif |

Убеждаемся, что с пользователями всё Ok.

После чего можно удалить ненужные группы и пользователей из системы.

|  |
| --- |
| [root@R-FW]# userdel -rf admin [root@R-FW]# for i in {1..99}; do userdel -rf user$i; done [root@R-FW]# groupdel Administrators [root@R-FW]# groupdel Users  [root@R-FW]# yum erase migrationtools |

Все ldif файлы нужно переместить на L-SRV и импортировать их в OpenLDAP.

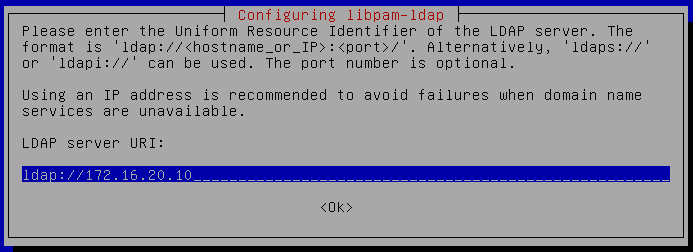
|  |
| --- |
| [root@L-SRV]# scp root@r-fw:/root/\*.ldif /root/ [root@L-SRV]# ldapadd -x -W -D 'cn=admin,dc=skill39,dc=wsr' -f ./base.ldif [root@L-SRV]# ldapadd -x -W -D 'cn=admin,dc=skill39,dc=wsr' -f ./groups.ldif [root@L-SRV]# ldapadd -x -W -D 'cn=admin,dc=skill39,dc=wsr' -f ./users.ldif |

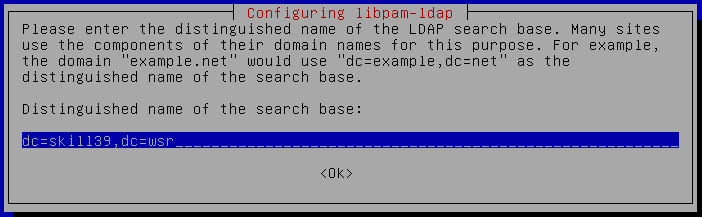
#### Подключаем клиентские машины

##### Debian (на примере L-FW)

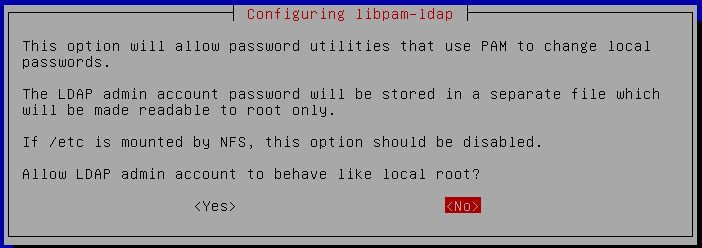
Для начала нужно установить PAM модуль аутентификации для LDAP. Этот модуль предоставляет интерфейс между LDAP сервером и системой аутентификации PAM и используется в связке с libnss-ldap.

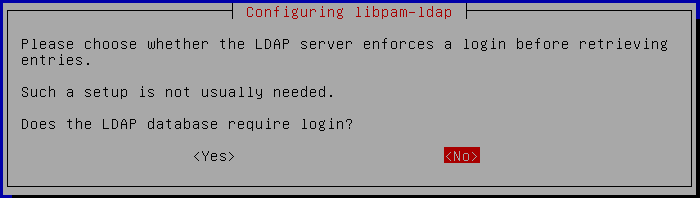
|  |
| --- |
| [root@L-FW]# apt-get install libpam-ldap |





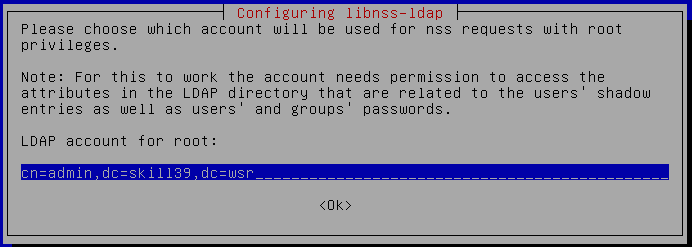


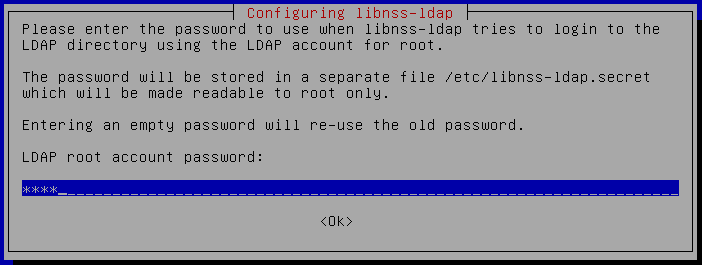


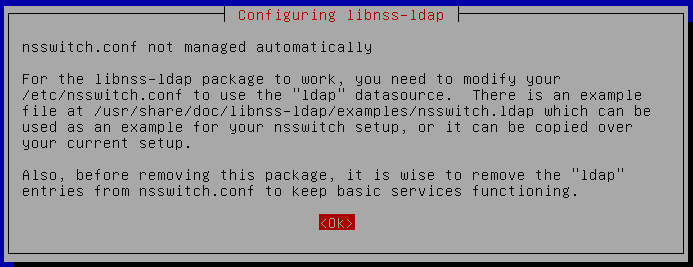


Сервис nscd (Name Service Caching Daemon) призван снизить нагрузку с того же LDAP на продакшен серверах, кэшируя запросы на резолвинг имен. NSS-LDAP (Name Services Switch) необходим для подтягивания из LDAP в систему таких данных как имена пользователей, группы и другой информации, которая обычно хранится в файлах каталога /etc.

|  |
| --- |
| [root@L-FW]# apt-get install libnss-ldap |







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

…

passwd: compat ldap

group: compat ldap

shadow: compat ldap

gshadow: files ldap

…

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

На L-FW может заходить только админ.

…

nss\_base\_passwd cn=admin,ou=Users,dc=skill39,dc=wsr

nss\_base\_shadow ou=Users,dc=skill39,dc=wsr

nss\_base\_group ou=Groups,dc=skill39,dc=wsr

…

На L-CLI

|  |
| --- |
| [root@L-CLI]# vim /etc/libnss-ldap.conf |

…

nss\_base\_passwd ou=Users,dc=skill39,dc=wsr

nss\_base\_shadow ou=Users,dc=skill39,dc=wsr

nss\_base\_group ou=Groups,dc=skill39,dc=wsr

…

|  |
| --- |
| [root@L-FW]# systemctl restart nscd.service |

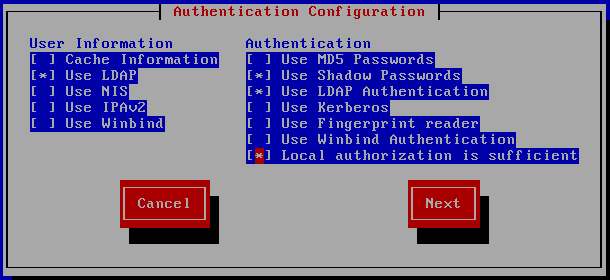
После чего можно проверить Alt-F2, логинимся как user22 (любой). Создаём файл в директории tmp и убеждаемся, что наша система знает, что файл принадлежит user22 и группе Users. Аналогично проверяем пользователя admin.

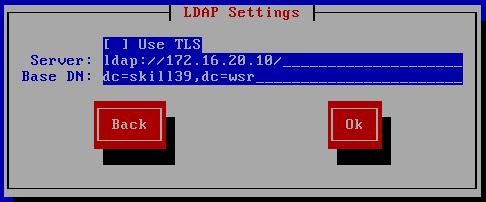
**Внимание!** На L-CLI недостаточно свободного места! Чтобы пользователи могли логинится через GUI нужно удалить что-нибудь ненужное. Например, libreoffice.

|  |
| --- |
| [root@L-CLI]# apt purge libreoffice [root@L-CLI]# apt autoremove [root@L-CLI]# apt autoclean [root@L-CLI]# apt clean |

##### CentOS (на примере R-FW) **(Если вдруг понадобится, мало ли)**

|  |
| --- |
| [root@R-FW]# yum install nss-pam-ldapd [root@R-FW]# authconfig-tui |





|  |
| --- |
| [root@R-FW]# vim /etc/nslcd.conf |

…

base group ou=Groups,dc=skill39,dc=wsr

base passwd ou=Users,dc=skill39,dc=wsr

base shadow ou=Users,dc=skill39,dc=wsr

…

## Конфигурация параметров безопасности и служб аутентификации

### Настройка CA

Сделали, когда настраивали OpenVPN и HTTPS.

### Настройка межсетевых экранов на L-FW и R-FW

Для начала нужно загрузить модуль хелпера ftp и добавить его в автозагрузку. Как известно, FTP писали на вудстоке наркоманы из 60-ых, поэтому там есть управляющее соединение и соединение для передачи файлов, которое инициируется со случайного порта и порт передается по управляющему. Что бы это дело натить нужно пройти 7 кругов ада или просто подключить хелпер для conntrack.

|  |
| --- |
| [root@L-FW]# modprobe nf\_nat\_ftp [root@L-FW]# echo nf\_nat\_ftp >> /etc/modules |

На L-FW /etc/iptables/rules.v4:

\*raw

:PREROUTING ACCEPT [0:0]

:OUTPUT ACCEPT [0:0]

**-A PREROUTING -p tcp --dport 21 -j CT --helper ftp**

COMMIT

\*nat

:PREROUTING ACCEPT [0:0]

:INPUT ACCEPT [0:0]

:OUTPUT ACCEPT [0:0]

:POSTROUTING ACCEPT [0:0]

# Пробрасываем порты для DNS запросов и для трансфера зоны

**-A PREROUTING -i ens32 -p udp --dport 53 -j DNAT --to 172.16.20.10**

**-A PREROUTING -i ens32 -p tcp --dport 53 -j DNAT --to 172.16.20.10**

# Пробрасываем FTP

**-A PREROUTING -i ens32 -p tcp --dport 21 -j DNAT --to 172.16.20.10**

# Правило для трансляции сетевых адресов

**-A POSTROUTING -o ens32 -j MASQUERADE**

COMMIT

\*filter

:INPUT **DROP** [0:0]

:FORWARD **DROP** [0:0]

:OUTPUT ACCEPT [0:0]

# Разрешаем loopback

**-A INPUT -i lo**

# Разрешаем подключения из приватных сеток

**-A INPUT -s 172.16.0.0/16 -j ACCEPT**

**-A INPUT -s 5.5.5.0/24 -j ACCEPT**

**-A INPUT -s 10.5.5.0/30 -j ACCEPT**

**-A INPUT -s 192.168.100.0/24 -j ACCEPT**

# Разрешаем GRE туннель

**-A INPUT -i ens32 -p gre -j ACCEPT**

# Разрешаем OpenVPN

**-A INPUT -i ens32 -p udp --dport 1122 -j ACCEPT**

# Разрешаем SSH

**-A INPUT -i ens32 -p tcp --dport 1022 -j ACCEPT**

# Всё что мы до этого наразрешали может инициировать соединения

# Теперь нам нужно разрешить всё что приходит в рамках установленных

# соединений

**-A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT**

# Все остальные входящие пакеты будут дропнуты политикой по умолчанию.

# ------------------------------------------------------

# Разрешаем трафик из и между приватными сетками

**-A FORWARD -s 172.16.0.0/16 -j ACCEPT**

**-A FORWARD -s 5.5.5.0/24 -j ACCEPT**

**-A FORWARD -s 10.5.5.0/30 -j ACCEPT**

**-A FORWARD -s 192.168.100.0/24 -j ACCEPT**

# Разрешаем проходить через нас DNS и FTP

**-A FORWARD -p udp --dport 53 -j ACCEPT**

**-A FORWARD -p tcp --dport 53 -j ACCEPT**

**-A FORWARD -p tcp --dport 21 -j ACCEPT**

# Разрешаем трафик для уже установленных соединений

**-A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT**

# Всё остальное будет в дропе по политике по умолчанию

COMMIT

|  |
| --- |
| [root@L-FW]# systemctl restart netfilter-persistent |

НА R-FW /etc/sysconfig/iptables

\*nat

:PREROUTING ACCEPT [0:0]

:INPUT ACCEPT [0:0]

:OUTPUT ACCEPT [0:0]

:POSTROUTING ACCEPT [0:0]

# Правило для трансляции сетевых адресов

**-A POSTROUTING -o ens160 -j MASQUERADE**

COMMIT

\*filter

:INPUT **DROP** [0:0]

:FORWARD **DROP** [0:0]

:OUTPUT ACCEPT [0:0]

# Разрешаем loopback

**-A INPUT -i lo**

# Разрешаем подключения из приватных сеток

**-A INPUT -s 172.16.0.0/16 -j ACCEPT**

**-A INPUT -s 5.5.5.0/24 -j ACCEPT**

**-A INPUT -s 10.5.5.0/30 -j ACCEPT**

**-A INPUT -s 192.168.100.0/24 -j ACCEPT**

# Разрешаем GRE туннель

**-A INPUT -i ens160 -p gre -j ACCEPT**

# Разрешаем SSH

**-A INPUT -i ens160 -p tcp --dport 22 -j ACCEPT**

# Всё что мы до этого наразрешали может инициировать соединения

# Теперь нам нужно разрешить всё что приходит в рамках установленных

# соединений

**-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT**

# Все остальные входящие пакеты будут дропнуты политикой по умолчанию.

# ------------------------------------------------------

# Разрешаем трафик из и между приватными сетками

**-A FORWARD -s 172.16.0.0/16 -j ACCEPT**

**-A FORWARD -s 5.5.5.0/24 -j ACCEPT**

**-A FORWARD -s 10.5.5.0/30 -j ACCEPT**

**-A FORWARD -s 192.168.100.0/24 -j ACCEPT**

# Разрешаем трафик для уже установленных соединений

**-A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT**

# Всё остальное будет в дропе по политике по умолчанию

COMMIT

|  |
| --- |
| [root@R-FW]# systemctl restart iptables |

## Критерии и Оценка задания

|  |  |
| --- | --- |
| **Basic configuration and Network Services** | |
| Hostnames | Random three hosts: hostname |
| IPv4 connectivity | L-SRV: ping 20.20.20.100  R-CLI: ping 10.10.10.1 |
| Software installation | Debian: dpkg -l ftp lftp vim lynx net-tools  CentOS: rpm -qa ftp lftp vim\* lynx net-tools |
| Local hostname table | Random two hosts:cat /etc/hosts |
| Name lookup order | Random two hosts:cat /etc/nsswitch.conf |
| DHCP: Basic Operation | L-CLI: ip a s && ip r s  L-FW: dhcp-lease-list |
| DHCP: Additional Parameters | L-CLI: cat /etc/resolv.conf |
| DNS: Forward zone | L-FW: host vpn.skill39.wsr 172.16.20.10 |
| DNS: Reverse zone | L-FW: host 172.16.20.10 172.16.20.10 |
| DNS: ISP Forwarders | L-FW: host worldskills.ru 172.16.20.10 |
| DNS: Secondary DNS | L-FW: host vpn.skill39.wsr 10.10.10.10 |
| DNS: Zone transfer restriction | L-FW: host -t AXFR skill39.wsr 172.16.20.10 |
| DNS: Dynamic DNS | L-FW: host l-cli 172.16.20.10 |
| Internet Gateway (Dynamic NAT) | L(R)-FW: iptables -t nat -L |
| **Centralized Management** | |
| LDAP: Users, Groups and OU | L-SRV: ldapsearch -Y EXTERNAL -H "ldapi:///" -b "ou=Users,dc=skill39,dc=wsr"  L-SRV: ldapsearch -Y EXTERNAL -H "ldapi:///" -b "ou=Groups,dc=skill39,dc=wsr" |
| LDAP: Clients authentication | L-CLI: Вход через user\* и admin. |
| LDAP: Logon restriction | L-FW: Вход только через admin. |
| Syslog: L-SRV | logger -p auth.info Message Auth |
| Syslog: L-FW | logger -p local7.crit Message Crit |
| Syslog: L-CLI | logger -p local7.err Message Err |
| Syslog: R-CLI | logger -p local7.err Message Err |
| **Routing & Remote Access** | |
| RA: OpenVPN basic | L-FW: less /etc/openvpn/server.conf  L-FW: ls -la /opt/vpn/  L-FW: systemctl status openvpn@server |
| RA: VPN Connect Script | OUT-CLI: start\_vpn.sh  OUT-CLI: ip a s  OUT-CLI: ping 5.5.5.1 |
| RA: VPN Disconnect Script | OUT-CLI: stop\_vpn.sh  OUT-CLI: ip a s |
| RA: VPN Clients have full access to LEFT and RIGHT LANs | OUT-CLI: ip route get 172.16.20.10  OUT-CLI: ip route get 192.168.100.0/24  OUT-CLI: ping 172.16.20.10  OUT-CLI: ping 192.168.100.1 |
| GRE Tunnel | L(R)-FW: ip a s  L-FW: ping 10.5.5.2  R-FW: ping 10.5.5.1 |
| SSH: Port and Users | L-FW: less /etc/ssh/sshd\_config  L-FW: ss -l4tp  L-SRV: ssh root@172.16.20.1 -p 1022 # Must fail  L-SRV: ssh ssh\_c@172.16.20.1 -p 1022 # Must be OK |
| SSH: Client’s Default Port | OUT-CLI: ssh ssh\_c@10.10.10.1  OUT-CLI: ssh root@10.10.10.1 |
| SSH: Key authentication | OUT-CLI: ssh ssh\_p@10.10.10.1 |
| **Web services, Storage and File sharing** | |
| NFS: Read and Write Access | L(R)-CLI: mount | grep nfs  L(R)-CLI: touch /opt/nfs/file |
| NFS: Automount | L(R)-CLI: cat /etc/fstab |
| FTP: Basic operation and User access | L-CLI: ftp l-srv # fail with root  L-CLI: ftp l-srv # good with ftpuser |
| FTP: Read and Write Access | L-CLI: dir  L-CLI: mkdir test |
| Web: http://www.skill39.wsr | L-FW: lynx http://www.skill39.wsr |
| Web: https://www.skill39.wsr | L-FW: lynx https://www.skill39.wsr |
| Web: Trusted SSL | При подключении по https не должно быть ошибок |
| Web: http –> https | При подключении должен быть редирект на https |
| **Security** | |
| OpenSSL: CA | R-FW: ls /opt/ca |
| Certificate Attributes | R-FW: openssl x509 -in /opt/ca/keys/ca.crt -text |
| IPTables: Block input traffic | iptables -L |
| IPTables: DNS port forwarding | R-FW: host vpn.skill39.wsr 10.10.10.1 |

## На всякий случай

### Если вам предоставлен только Debian DVD-1

а все необходимые пакеты лежат в AdditionalPackages.iso.

Дальше на примере bind9 из AdditionalPackages.v3.0.iso. Дальше предлагаю попробовать все варианты, выполняя после каждого удаление bind9 через apt.

|  |
| --- |
| [root@host]# apt remove bind9 [root@host]# apt autoremove |

Иатк. Монтируем iso и копируем содержимое на виртуальную машину.

|  |
| --- |
| [root@host]# mount /dev/cdrom /media/cdrom [root@host]# mkdir /srv/repo [root@host]# cp -R /media/cdrom/\* /srv/repo/ |

**Вариант #1 - dpkg**

Если все пакеты отсортированы по директориям и вы знаете что к чему.

Dpkg устанавливает пакеты, но не разрешает зависимости. “\*.deb” - glob-подстановка, которая означает - все файлы, которые заканчиваются на “.deb”.

|  |
| --- |
| [root@host]# dpkg -i /srv/repo/bind9/\*.deb [root@host]# apt -f install |

“apt -f install” просканирует систему и попытается установить недостающие пакеты из доступных источников. ( В вашем случае DVD-1 )

**Вариант #2 - apt**

Если все пакеты отсортированы по директориям и вы знаете что к чему. Установка пакетов происходит через пользователя \_apt, поэтому лучше сменить владельца директории и дать ему полный доступ. В любом, случае, если что будет не так, apt будет ругаться и скажет, что ему нужно.

|  |
| --- |
| [root@host]# chown -R \_apt /srv/repo [root@host]# chmod -R u+rwx /srv/repo [root@host]# apt-get install /srv/repo/bind9/\*.deb |

**Вариант #3 - делаем локальный репозиторий**

Если все пакеты свалены в кучу и вы не знаете что к чему. Чтобы создать репозиторий мало иметь директорию с пакетами. Нужно также иметь индекс и Release файл. Индекс можно сгенерировать с помощью утилиты dpkg-scanpackages из пакета “dpkg-dev”, который доступен на DVD-1. dpkg-scanpackages просканирует директорию и выведет всю информацию о найденных пакетах на stdout. В репозитории индекс должен храниться в виде архива Packages.gz.

|  |
| --- |
| [root@host]# apt install dpkg-dev [root@host]# dpkg-scanpackages /srv/repo | gzip > /srv/repo/Packages.gz [root@host]# vim /etc/apt/sources.list |

После этого необходимо добавить наш локальный репозиторий в sources.list.

Опция [trusted=yes] позволяет сказать apt, что данный репозиторий доверенный и из него можно устанавливать пакеты без проверки подписей пакетов и без наличия Release файла, который мы не хотим генерить.

...

deb [trusted=yes] file:/srv/ repo/

* file:/srv/ - URI на репозиторий
* repo/ - сам репозиторий

После этого необходимо обновить локальную базу пакетов, и можно устанавливать пакеты привычным образом через apt.

|  |
| --- |
| [root@host]# apt update [root@host]# apt install bind9 |

Apt может запрашивать DVD-1, где могут находится недостающие зависимости.

### Если на диске вам предоставили RPM пакет easy-rsa версии 3

Для создания сертификатов нам понадобится пакет EASY-RSA, если он будет доступен в виде rpm на диске. Если он доступен, устанавливаем пакет и смотрим что и куда он копирует.

|  |
| --- |
| [root@R-FW]# mount /dev/cdrom /media/cdrom [root@R-FW]# yum install /media/cdrom/easy-rsa.rpm [root@R-FW]# rpm -ql easy-rsa |

В данном случае, на CentOS доступен пакет третьей версии. Чтобы воспользоваться скриптами нужно скопировать их в рабочую директорию. Также необходимо скопировать файл с переменными “vars”.

|  |
| --- |
| [root@R-FW]# cp -Rp /usr/share/easy-rsa /opt/ca [root@R-FW]# cd /opt/ca/easy-rsa/3/ [root@R-FW ca]# cp /usr/share/doc/easy-rsa-3.0.3/vars.example ./vars  [root@R-FW ca]# vim ./vars |

В файле vars будут располагаться переменные, которые будут использоваться OpenSSL для генерации сертификатов. По заданию, переменные должны выглядеть следующим образом.

# Измените значение на "org", иначе поля ниже не будут применяться

set\_var EASYRSA\_DN "org"

# Установите переменные согласно задания

set\_var EASYRSA\_REQ\_COUNTRY "RU"

set\_var EASYRSA\_REQ\_PROVINCE "."

set\_var EASYRSA\_REQ\_CITY "."

set\_var EASYRSA\_REQ\_ORG "WorldSkills Russia"

set\_var EASYRSA\_REQ\_EMAIL ""

set\_var EASYRSA\_REQ\_OU "."

set\_var EASYRSA\_REQ\_CN "WSR CA"

О том какие команды и как применять можно посмотреть в README

|  |
| --- |
| [root@R-FW]# less /usr/share/doc/easy-rsa-3.0.3/README.quickstart.md |

Генерируем сертификат CA.

|  |
| --- |
| [root@R-FW]# ./easyrsa init-pki [root@R-FW]# ./easyrsa build-ca nopass |

Генерируем ключ для L-FW

|  |
| --- |
| [root@R-FW]# ./easyrsa gen-req l-fw nopass [root@R-FW]# ./easyrsa sign-req server l-fw |

Генерируем ключ для OUT-CLI

|  |
| --- |
| [root@R-FW]# ./easyrsa gen-req out-cli nopass [root@R-FW]# ./easyrsa sign-req client out-cli |

Генерируем DH (Диффи Хеллман)

|  |
| --- |
| [root@R-FW]# ./easyrsa gen-dh |

Копируем ключи на L-FW и OUT-CLI:

|  |
| --- |
| [root@R-FW]# scp pki/ca.crt root@10.10.10.1:/opt/vpn [root@R-FW]# scp pki/issued/L-FW.crt root@10.10.10.1:/opt/vpn [root@R-FW]# scp pki/private/L-FW.crt root@10.10.10.1:/opt/vpn [root@R-FW]# scp pki/dh.pem root@10.10.10.1:/opt/vpn [root@R-FW]# scp pki/ca.crt root@20.20.20.5:/opt/vpn [root@R-FW]# scp pki/issued/L-FW.crt root@20.20.20.5:/opt/vpn [root@R-FW]# scp pki/private/L-FW.crt root@20.20.20.5:/opt/vpn [root@R-FW]# scp pki/dh.pem root@20.20.20.5:/opt/vpn |

### Если использовать firewalld на R-FW

Разрешаем протокол GRE

|  |
| --- |
| [root@R-FW]# firewall-cmd --permanent --add-protocol=gre --zone=external |

Добавляем gre1 в доверенные

|  |
| --- |
| [root@R-FW]# firewall-cmd --permanent --add-interface=gre1 --zone=internal |

Где-то под капотом, firewalld блокирует по состоянию соединения, поэтому чтобы conntrack смог отслеживать gre соединения нужно подключить соответствующий модуль.

|  |
| --- |
| [root@R-FW]# find /lib -name "\*conntrack\*" [root@R-FW]# modprobe nf\_conntrack\_proto\_gre |

Добавляем в автозагрузку:

|  |
| --- |
| [root@R-FW]# vim /etc/modprobe.d/nf\_conn.conf |

nf\_conntrack\_proto\_gre