Ранее я уже рассматривал настройку программных роутеров на операционных системах freebsd и centos. Сегодня я хочу настроить интернет шлюз для локальной сети на основе популярного linux дистрибутива Debian. Выполним подготовку сервера и реализуем основной функционал, необходимый для выхода в интернет из локальной сети.

Содержание:

Данная статья является частью единого цикла статьей про сервер Debian.

Введение

Я буду работать со следующим сервером:

```
# uname -a
```

```
Linux debian 3.16.0-4-amd64 #1 SMP Debian 3.16.7-ckt20-1+deb8u3 (2016-01-17) x86_64 GNU/Linux
# cat /etc/debian_version
8.3
```

Если у вас еще нет готового сервера, то рекомендую статью на тему установки debian. Там подробно описан весь процесс от и до.

На сервере имеются 2 сетевых интерфейса:

Описание сетевых интерфейсов

Интерфейс	Описание	IP
eth0	внешний интерфейс, подключен к провайдеру, настройки получает по dhcp автоматически	192.168.1.24
eth1	внутренний интерфейс, смотрит в локальную сеть, статический ip адрес	10.0.15.1

Файл конфигурации сетевых интерфейсов выглядит следующим образом:

```
# cat /etc/network/interfaces

source /etc/network/interfaces.d/*

auto lo
iface lo inet loopback

allow-hotplug eth0
iface eth0 inet dhcp

allow-hotplug eth1
iface eth1 inet static
address 10.0.15.1
netmask 255.255.255.0
```

Как настроить сеть в debian я подробно рассказал отдельно. Если вы еще не сделали это и не выполнили предварительную настройку debian, то рекомендую ознакомиться с материалами.

Если у вас недостаточно опыта и вы не чувствуете в себе сил разобраться с настройкой шлюза самому с помощью консоли сервера - попробуйте дистрибутив на основе centos для организации шлюза и прокси сервера в локальной сети - clearos. С его помощью можно через браузер настроить весь необходимый функционал. В отдельной статье я подробно рассказал об установке clearos.

Подготовка шлюза

Выше я привел ссылку на подробную статью с описанием настройки сервера общего назначения. Сейчас мы выполним некоторые подготовительные действия без подробностей и описания. Их вы можете почитать отдельно. Сейчас просто необходимые команды.

Сеть на будущем программном роутере настроили, доступ в интернет на сервере есть. Обновим его:

```
# apt-get update
# apt-get upgrade
```

Установим MC, мне в нем удобнее всего работать, в том числе в редакторе **mcedit**:

```
# apt-get -y install mc
```

Настроим часовой пояс, если раньше не сделали это:

```
# dpkg-reconfigure tzdata
```

Устанавливаем сервис **ntp** для автоматического обновления времени:

```
# apt-get -y install ntp
```

На этом основные подготовительные действия закончены. Приступаем к настройке шлюза.

Настройка маршрутизации, firewall и nat

Первым делом включим маршрутизацию пакетов между сетевыми интерфейсами. Для этого редактируем конфиг */etc/sysctl.conf*:

```
# mcedit /etc/sysctl.conf

net.ipv4.ip_forward=1
```

Либо раскомментируйте эту строку, либо добавьте, если ее нет. Но она по-умолчанию быть должна, закомментированная со значением 1. Применяем эту настройку:

```
# sysctl -p
```

На выходе работы команды в консоли будет выведен измененный параметр со значением 1.

Теперь приступаем к самому главному - настройке фаервола iptables и nat в нем для обеспечения выхода в интернет из локальной сети. Я очень подробно рассмотрел эту тему в отдельной статье. Хотя там речь идет о другом дистрибутиве, сами правила iptables абсолютно одинаковые с точностью до строчки, за исключением маленького нюанса, связанного с тем, что правила нужно сохранять в другой файл для применения их после перезагрузки.

Я приведу здесь сразу готовый вариант файла с правилами iptables, необходимых для работы интернет шлюза в debian. В файле даны подробные комментарии ко всем значениям, так что вы без проблем разберетесь и закомментируете или наоборот раскомментируете необходимые вам значения. Качаем скрипт правил iptables - iptables-debian.sh

Копируем содержимое файла и создаем скрипт с правилами на сервере:

```
# mcedit /etc/iptables.sh
```

Вставляем в редактор правила. Редактируем их под свои нужды, обязательно заменяя переменные WAN и LAN на свои. Сохраняем файл.

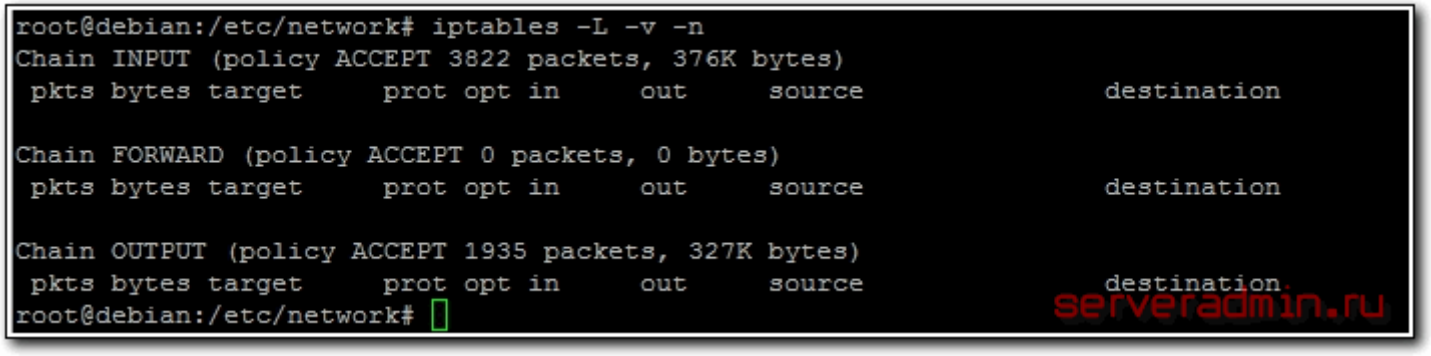
> Прежде чем двигаться дальше предупреждаю, что все работы по настройке фаервола должны производиться только если у вас есть доступ к консоли сервера, чтобы в случае ошибки и потери удаленного доступа вы смогли откатить изменения. Даже если вы абсолютно уверены в своих знаниях, вас может подвести банальная ошибка или опечатка. Я сам, к сожалению, сталкивался с такими ситуациями, поэтому считаю необходимым предупредить об этом вас.

Делаем файл с правилами исполняемым:

```
# chmod 0740 /etc/iptables.sh
```

Прежде чем применить новые правила, посмотрим на текущие:

```
# iptables -L -v -n
```



Видим, что на настраиваемом роутере firewall полностью открыт. Теперь применим новые правила и посмотрим на результат:

```
# /etc/iptables.sh
```

```
root@debian:/etc/network# /etc/iptables.sh
root@debian:/etc/network# iptables -L -v -n
Chain INPUT (policy DROP 1 packets, 78 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     all  --  eth1   *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     icmp --  *      *       0.0.0.0/0            0.0.0.0/0           icmptype 0
    0     0 ACCEPT     icmp --  *      *       0.0.0.0/0            0.0.0.0/0           icmptype 3
    0     0 ACCEPT     icmp --  *      *       0.0.0.0/0            0.0.0.0/0           icmptype 11
    0     0 ACCEPT     icmp --  *      *       0.0.0.0/0            0.0.0.0/0           icmptype 8
    5   404 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0           state INVALID
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x3F/0x00
    1    40 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:!0x17/0x02 state NEW
    0     0 ACCEPT     tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0           state INVALID
    0     0 ACCEPT     all  --  eth1   eth0    0.0.0.0/0            0.0.0.0/0
    0     0 REJECT     all  --  eth0   eth1    0.0.0.0/0            0.0.0.0/0           reject-with icmp-port-unreachable

Chain OUTPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  *      lo      0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     all  --  *      eth1    0.0.0.0/0            0.0.0.0/0
    5   600 ACCEPT     all  --  *      eth0    0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:!0x17/0x02 state NEW
root@debian:/etc/network#
```


Все в порядке, правила применились, доступ к серверу я не потерял. Теперь сделаем так, чтобы новые правила применялись после перезагрузки. В последней строчке скрипта есть команда:

```
/sbin/iptables-save  > /etc/iptables.rules
```

С ее помощью готовый набор правил iptables выгружаются в файл. Нам нужно сделать так, чтобы эти правила применялись при включении сетевого интерфейса во время загрузки сервера. Для этого открываем файл *interfaces* на редактирование и добавляем в самый конец строчку:

```
# mcedit /etc/network/interfaces

post-up iptables-restore < /etc/iptables.rules
```

Для проверки перезагружаем шлюз и проверяем, все ли в порядке. По сути основная настройка программного роутера на debian завершена. Осталось сделать небольшое дополнение и настроить **dhcp** и **dns** сервер в локальной сети. Я для этих целей использую простой и легкий в настройке dnsmasq.

Установка и настройка dnsmasq в Debian

Выполним установку dnsmasq на дебиан:

```
# apt-get install -y dnsmasq
```

Сделаем минимальную настройку программы. Нам нужно просто выдавать сетевые настройки пользователям. Для этого приводим конфигурационный файл dnsmasq к следующему виду:

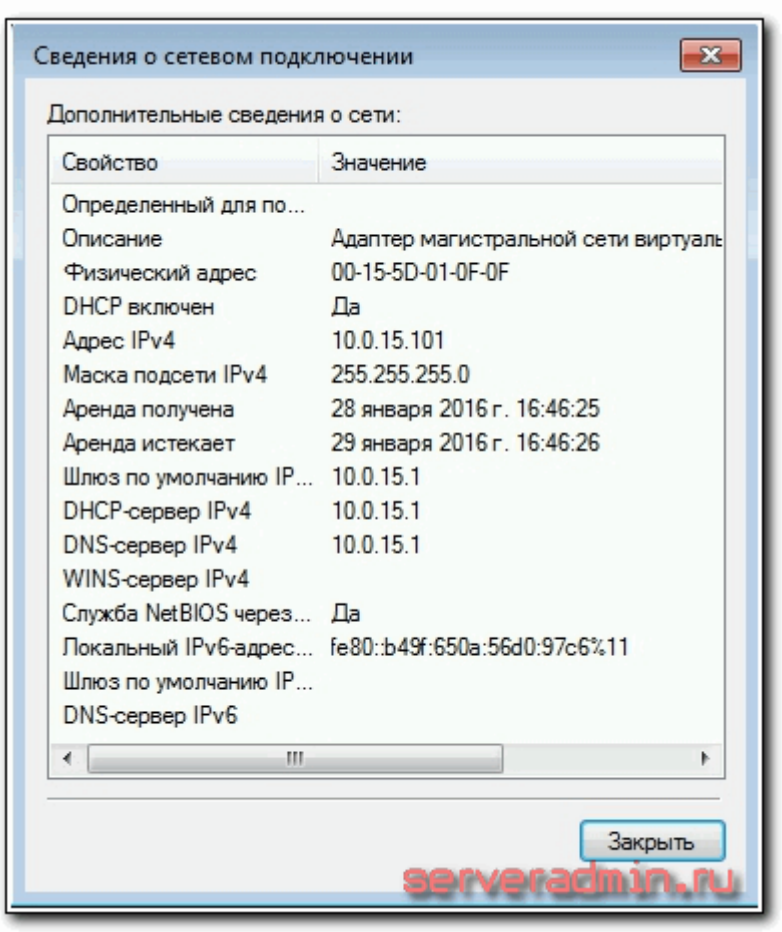
```
# mcedit /etc/dnsmasq.conf

domain-needed
bogus-priv
interface=eth1
```

```
dhcp-range=eth1,10.0.15.50,10.0.15.150,24h
```

В данном случае мы будем выдавать пользователям ip адреса в диапазоне от 10.0.15.50 до 150. Сохраняем конфиг, добавляем программу в автозагрузку и запускаем.

```
# insserv dnsmasq
# /etc/init.d/dnsmasq start
```

Теперь можно запускать компьютер пользователя локальной сети, получать сетевые настройки по dhcp и проверять работу интернет шлюза.



Посмотреть выданные leases можно в файле */var/lib/misc/dnsmasq.leases*. На этом настройка интернет шлюза на debian 8 закончена. Все что нужно для обеспечения доступа в интернет из локальной сети сделано. Получился программный роутер с широкими возможностями по наращиванию функционала.

Просмотр загрузки сети с помощью iftop

Теперь представим ситуацию, что кто-то забил весь интернет канал и вам надо быстро выяснить, кто это сделал. По-умолчанию, никаких подручных и удобных средств на шлюзе для этого нету. Установим одно из таких средств - программу **iftop**. Это простая консольная утилита, которая дает возможность оперативно посмотреть статистику загруженности сетевого интерфейса в реальном времени.

Устанавливаем iftop на debian:

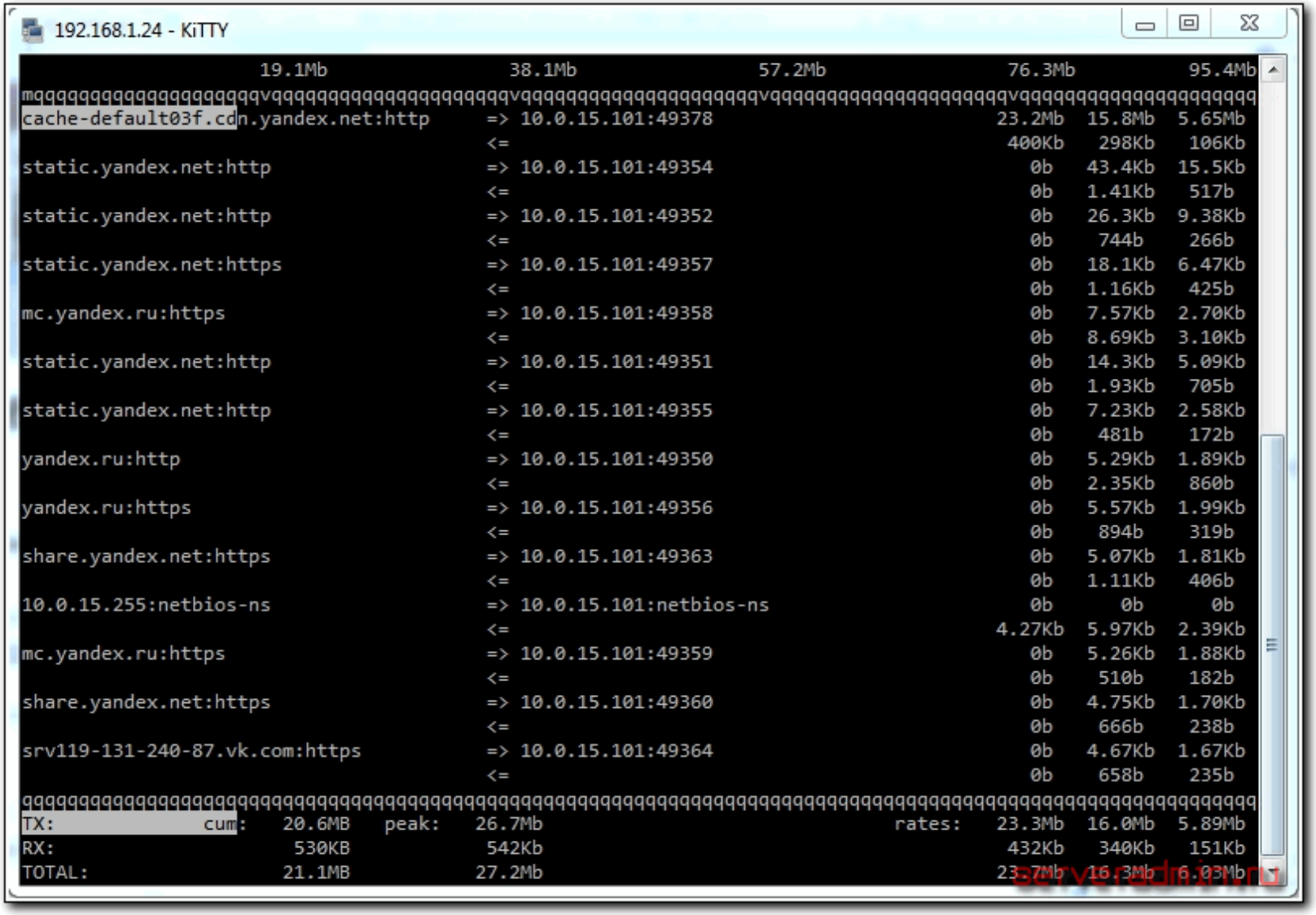
```
# apt-get install -y iftop
```

Для просмотра активности сетевого интерфейса, запускаем утилиту, указывая необходимый ключ:

```
# iftop -i eth1
```

Чтобы увидеть порты, по которым идет трафик, добавляем ключ -P:

```
# iftop -i eth1 -P
```


192.168.1.24 - KiTTY
19.1Mb 38.1Mb 57.2Mb 76.3Mb 95.4Mb
cache-default03f.cdn.yandex.net:http => 10.0.15.101:49378 23.2Mb 15.8Mb 5.65Mb
<= 400Kb 298Kb 106Kb
static.yandex.net:http => 10.0.15.101:49354 0b 43.4Kb 15.5Kb
<= 0b 1.41Kb 517b
static.yandex.net:http => 10.0.15.101:49352 0b 26.3Kb 9.38Kb
<= 0b 744b 266b
static.yandex.net:https => 10.0.15.101:49357 0b 18.1Kb 6.47Kb
<= 0b 1.16Kb 425b
mc.yandex.ru:https => 10.0.15.101:49358 0b 7.57Kb 2.70Kb
<= 0b 8.69Kb 3.10Kb
static.yandex.net:http => 10.0.15.101:49351 0b 14.3Kb 5.09Kb
<= 0b 1.93Kb 705b
static.yandex.net:http => 10.0.15.101:49355 0b 7.23Kb 2.58Kb
<= 0b 481b 172b
yandex.ru:http => 10.0.15.101:49350 0b 5.29Kb 1.89Kb
<= 0b 2.35Kb 860b
yandex.ru:https => 10.0.15.101:49356 0b 5.57Kb 1.99Kb
<= 0b 894b 319b
share.yandex.net:https => 10.0.15.101:49363 0b 5.07Kb 1.81Kb
<= 0b 1.11Kb 406b
10.0.15.255:netbios-ns => 10.0.15.101:netbios-ns 0b 0b 0b
<= 4.27Kb 5.97Kb 2.39Kb
mc.yandex.ru:https => 10.0.15.101:49359 0b 5.26Kb 1.88Kb
<= 0b 510b 182b
share.yandex.net:https => 10.0.15.101:49360 0b 4.75Kb 1.70Kb
<= 0b 666b 238b
srv119-131-240-87.vk.com:https => 10.0.15.101:49364 0b 4.67Kb 1.67Kb
<= 0b 658b 235b
TX: cum: 20.6MB peak: 26.7Mb rates: 23.3Mb 16.0Mb 5.89Mb
RX: 530KB 542Kb 432Kb 340Kb 151Kb
TOTAL: 21.1MB 27.2Mb 23.7Mb 16.3Mb 6.03Mb
serveradmin.ru

На основе этой картинки уже можно сделать определенные выводы по использованию интернет канала. Обращаю внимание, что я смотрю загрузку локального интерфейса eth1. Если смотреть на eth0, то мы увидим только исходящие соединения сервера.

Заключение

Вот так легко и быстро можно настроить роутер, маршрутизатор или шлюз в интернет. Названия разные, а суть одна. В данном случае я использовал операционную систему Debian, но схожий функционал легко организовать на Freebsd или CentOS. Для решения текущей задачи разница в работе не будет заметна. Каждый выбирает то, что больше нравится и к чему привык.

Пройдемся быстренько по этапам того, что сделали:

1. Подготовили сервер Debian к настройке шлюза.
2. Настроили маршрутизацию, iptables, нат. Проверили, что весь функционал восстанавливается после перезагрузки.
3. Установили и настроили простой dhcp сервер и кэширующий dns сервер - dnsmasq. С его помощью автоматизировали поучение сетевых настроек пользователями.
4. Установили простое средство мониторинга сетевой активности в консоли в режиме реального времени с помощью утилиты iftop.

На этом мы закончили настройку. Как продолжение развития темы интернет шлюза можно заняться настройкой прокси сервера для управления доступам к ресурсам интернета, или сервера openvpn для подключения филиалов или удаленных сотрудников. Для примера привел ссылки на другие дистрибутивы. Со временем планирую описать реализацию этого функционала на debian. Принципиальных отличий нет, только нюансы разных дистрибутивов.

Напоминаю, что данная статья является частью единого цикла статьей про сервер Debian.

Помогла статья? Подписывайся на telegram канал автора

Анонсы всех статей, плюс много другой полезной и интересной информации, которая не попадает на сайт.

Дополнительные материалы по Debian

Рекомендую полезные материалы по Debian:

Настройки системы

- Установка
- **Базовая настройка**
- Настройка сети
- Обновление 8 до 9
- Обновление 7 до 8
- Включение логов cron

Подробная установка Debian 9 Stratch с помощью графического инсталлятора со скриншотами и пояснениями к каждому пункту установщика.
Базовая настройка сервера Debian. Приведены практические советы по улучшению безопасности и удобства администрирования.
Подробное описание настройки сети в Debian - задать ip адрес, dhcp, отключить ipv6, dns, hostname, статические маршруты и др.
Обновление предыдущей версии Debian 8 Jessie до последней Debian 9 Stratch. Подробная инструкция с описанием по каждому этапу обновления.
Обновление версии Debian 7 wheezy до Debian 8 Jessie. Подробная инструкция с описанием по каждому этапу обновления.
Включение записи логов cron в Debian в отдельный файл и настройка ротации этого файла. Отключение логов в syslog.

Настройка программных комплексов

• Proxmox
• Шлюз в интернет
• Установка Asterisk
• Asterisk+Freepbx
• PostgreSQL для 1C
• Настройка pptp

Подробное описание установки гипервизора proxmox на raid1 mdadm на базе операционной системы Debian 8. Приведены практические советы по настройке.
Настройка интернет шлюза на Debian. Включает в себя настройку iptables, nat, dhcp, dns, iftop.
Чистая установка Asterisk 13 на сервер под управлением Debian 8. Никаких дополнений и GUI, только vanilla asterisk.
Установка Freepbx 12 и Asterisk 13 на сервер под управлением Debian/Ubuntu. Подробное описание и разбор ошибок установки.
Рассказ об установке и небольшой настройке сервера бд postgresql для работы с базами 1С. Задача не сложная, но есть небольшие нюансы как по настройке, так и по выбору дистрибутива.
Описание установки и настройки pptp сервера в Debian с передачей статических маршрутов клиенту для организации доступа к ресурсам сети.

Разное

• Бэкап с помощью rsync
• Тюнинг postgresl для 1C

Подробное описание настройки бэкапа с помощью rsync на примере скрипта инкрементного архива на системе Centos, Debian, Ubuntu, Windows.
Ускорение работы 1С с postgresql и диагностика проблем производительности