



Quagga - превращение сервера в полнофункциональный маршрутизатор

Опубликовано muff в Чт, 2009-08-27 01:38

Немного общей информации для начала... Quagga — пакет программного обеспечения, позволяющий превратить обычный компьютер в маршрутизатор TCP/IP с поддержкой разнообразных протоколов маршрутизации. Поддерживаются следующие протоколы маршрутизации:

- [статическая маршрутизация](#);
- Routing Information Protocol (RIP): v1, v2, v3
- [Open Shortest Path First](#) (OSPF): v2, v3
- [Border Gateway Protocol](#) [1] (BGP): v4

На данный момент возникла необходимость в статической маршрутизации. Соответственно опишем установку и начальную настройку [демона zebra](#).

Устанавливаем quagga из портов:

```
# cd /usr/ports/net/quagga/  
# make install clean
```

Как видим из предложенных опций, в последних версиях появилась поддержка протокола IS-IS (Intermediate System to Intermediate System). Но поскольку поддержка данного протокола находится в экспериментальном режиме, то пока откажемся от его установки. По ходу, все опции оставляю по умолчанию.

По завершению установки добавляем необходимость запуска quagga и демона zebra при запуске системы:

```
# cat /etc/rc.conf | grep quagga  
quagga_enable="YES"  
quagga_daemons="zebra"  
watchquagga_enable="YES"  
watchquagga_flags="-dz -R '/usr/local/sbin/zebra -d;' zebra"
```

Watchquagga - это дополнение к quagga. Мониторит доступность указанных демонов quagga, и в случае, если демон не отвечает - перезапускает его.

Создаем каталог, где будут лежать конфигурационные файлы наших демонов маршрутизации. Создаем файл конфигурации для демона статической маршрутизации zebra, выставляем необходимые права:

```
# touch /usr/local/etc/quagga/zebra.conf  
# chmod 600 /usr/local/etc/quagga/zebra.conf  
# chown -R quagga:quagga /usr/local/etc/quagga
```

Наполняем файл следующим содержимым:

```
# cat /usr/local/etc/quagga/zebra.conf  
  
password pass1  
enable password pass2
```



Дело в том, что подключаться к демону маршрутизации для последующей настройки мы будем по виртуальному терминалу, а в случае, если пароль не задан, то терминал это подключение отбросит (требование безопасности).

Кстати, не помешает создать каталог, где будет хранить свои логи quagga:

```
# mkdir /var/log/quagga
# chown quagga:quagga /var/log/quagga
```

Запускаем quagga и пробуем подключиться к запущенному демону:

```
# sh /usr/local/etc/rc.d/quagga start
Starting zebra.
# telnet localhost zebra
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is Quagga (version 0.99.14).
Copyright 1996-2005 Kunihiro Ishiguro, et al.
  User Access Verification

Password:
> enable
Password:
#
```

Консоль управления полностью идентична Cisco CLI. Поэтому многим будет знакома. Если же нет - смотрите комментарии к набираемым командам (выделено красным).

Итак, мы уже подклюены к виртуальной консоли. Начнем настройку:

Перейдем в режим глобальной конфигурации:

```
# configure terminal
```

Обратите внимание на то, что изменилось приглашение командной строки. Это как-бы подсказка, в каком режиме мы находимся...

```
(config)#
```

Первым делом дадим название данному роутеру, чтобы не путаться в последующем, в каком из демонов маршрутизации мы находимся, и на каком из роутеров. Обычно название я даю, исходя из демона маршрутизации и названия роутера.

```
(config)# hostname zebra-router
zebra-router(config)#
```

Кстати, имейте ввиду, что все изменения конфигурации вступают в силу немедленно после ввода команды!

Дальше рекомендую включить сервис шифрования паролей. Не очень хорошо то, что они хранятся в открытом виде.

```
zebra-router(config)# service password-encryption
```

Настроим логирование:



```
zebra-router(config)# log stdout
```

```
zebra-router(config)# log file /var/log/quagga/zebra.log
```

Вроде пока все гуд... Сохраним текущую конфигурацию (ведь на данный момент все изменения хранятся в оперативной памяти).

```
zebra-router(config)# exit
```

```
zebra-router# write memory
```

Configuration saved to /usr/local/etc/quagga/zebra.conf

Посмотрим что у нас получилось:

```
zebra-router# show startup-config
```

```
!  
! Zebra configuration saved from vty  
! 2009/09/22 15:24:43  
!  
hostname zebra-router  
password 8 bJdy6GGF2QMTg  
enable password 8 4D7IL52IMQH4Q  
log file /var/log/quagga/zebra.log  
log stdout  
service password-encryption  
!  
interface em0  
  ipv6 nd suppress-ra  
!  
interface lo0  
!  
interface re0  
  ipv6 nd suppress-ra  
!  
interface re1  
  ipv6 nd suppress-ra  
!  
ip forwarding  
!  
!  
line vty  
!  
zebra-router#
```

Для добавления статического маршрута необходимо в режиме глобальной конфигурации набить команду:

```
zebra-router(config)# ip route x.x.x.x/x y.y.y.y
```

Где x.x.x.x/x - адрес сети, для которой необходимо изменить маршрут, y.y.y.y - адрес следующего хопа (ip следующего маршрутизатора). Вместо ip-адреса следующего хопа можно указывать интерфейс, в который перенаправлять пакеты.

Просмотреть текущий маршрут к хосту можно командой show ip route (чтобы просмотреть все текущие маршруты, команду набиваем без ключей):

```
zebra-router# show ip route y.y.y.y
```

```
Routing entry for x.x.x.x/x  
  Known via "static", distance 1, metric 0, best
```



```
* z.z.z.z, via tun0
```

OSPF

Собственно продолжим "терзать" quagga. Есть необходимость организовать обмен маршрутами между роутерами по протоколу [ospf](#) [2]. Будем отталкиваться от того, что сам пакет уже установлен, zebra работает...

Создадим файл конфигурации демона ospfd, выставим ему необходимые права:

```
# touch /usr/local/etc/quagga/ospfd.conf
# chmod 600 /usr/local/etc/quagga/ospfd.conf
# chown -R quagga:quagga /usr/local/etc/quagga
```

Наполняем файл следующим содержимым:

```
# cat /usr/local/etc/quagga/ospfd.conf
```

```
password pass1
enable password pass2
```

Изменяем данные rc.conf (добавляем поддержку ospfd):

```
# cat /etc/rc.conf | grep quagga
quagga_enable="YES"
quagga_daemons="zebra ospfd"
watchquagga_enable="YES"
watchquagga_flags="-dz -R '/usr/local/sbin/zebra -d; /usr/local/sbin/ospfd -d;' zebra ospfd"
```

Перезапускаем quagga:

```
# sh /usr/local/etc/rc.d/quagga restart
ospfd not running? (check /var/run/quagga/ospfd.pid).
Stopping zebra.
Starting zebra.
2010/02/17 22:54:51 ZEBRA: can't get ip6forwarding value
Starting ospfd.
```

Подключаемся к терминалу и приступаем к конфигурированию ospfd (соединение между роутерами реализовано по виртуальному интерфейсу vlan17).

Конфигурация Router1

```
# telnet localhost ospfd
```

```
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^['.
```

```
Hello, this is Quagga (version 0.99.15).
Copyright 1996-2005 Kunihiro Ishiguro, et al.
User Access Verification
```

```
Password:
```

```
> enable
```

```
Password:
```

Переходим в режим глобальной конфигурации

```
# configure terminal
```

Даем название роутеру, отталкиваясь от демона маршрутизации:



```
(config)# hostname ospfd-router1
```

Включаем шифрование паролей

```
ospfd-router1(config)# service password-encryption
```

Настройка логирования

```
ospfd-router1(config)# log stdout
```

```
ospfd-router1(config)# log file /var/log/quagga/ospfd.log
```

Переход к конфигурированию демона ospf

```
ospfd-router1(config)# router ospf
```

Задаем идентификатор маршрутизатора

```
ospfd-router1(config-router)# ospf router-id 10.16.17.1
```

Какие сети будем анонсировать (непосредственно подключенные и статические маршруты)

```
ospfd-router1(config-router)# redistribute connected
```

```
ospfd-router1(config-router)# redistribute static
```

Задаем идентификатор области, в которую входит подключенная сеть

```
ospfd-router1(config-router)# network 10.16.17.0/29 area 0.0.0.0
```

```
ospfd-router1(config-router)# exit
```

Приступаем к настройке интерфейса

```
ospfd-router1(config)# interface vlan17
```

Описание. Нужная штука, особенно если OSPF поднят на нескольких интерфейсах

```
ospfd-router1(config-if)# description Internal_interface
```

Задаем "стоимость" канала. Используется при вычислении оптимального маршрута

```
ospfd-router1(config-if)# ip ospf cost 100
```

Промежуток времени в секундах, между передачей интерфейсом двух последовательных пакетов Hello

```
ospfd-router1(config-if)# ip ospf hello-interval 8
```

Время в секундах, по истечению которого соседний маршрутизатор считается неработающим. Время исчисляется от момента приема последнего пакета Hello от соседнего маршрутизатора. Значение этого параметра анонсируется в пакетах Hello.

```
ospfd-router1(config-if)# ip ospf dead-interval 24
```

Интервал времени в секундах по истечении которого маршрутизатор повторно отправит пакет на который не получил подтверждения о получении

```
ospfd-router1(config-if)# ip ospf retransmit-interval 4
```

```
ospfd-router1(config-if)# exit
```

```
ospfd-router1(config)# exit
```

```
ospfd-router1# write memory
```

Configuration saved to /usr/local/etc/quagga/ospfd.conf

По аналогии конфигурируем Router2 (подразумевается, что quagga уже установлена, подготовительные манипуляции завершены, осталось только сконфигурировать):

```
# telnet localhost ospfd
```

Trying ::1...

Trying 127.0.0.1...

Connected to localhost.

Escape character is '^['.

Hello, this is Quagga (version 0.99.15).

Copyright 1996-2005 Kunihiro Ishiguro, et al.

User Access Verification

Password:

```
> enable
```

Password:

```
# configure terminal
```

```
(config)# hostname ospfd-router2
```

```
ospfd-router2(config)# service password-encryption
```



```
ospfd-router2(config)# log stdout
ospfd-router2(config)# log file /var/log/quagga/ospfd.log
ospfd-router2(config)# router ospf
ospfd-router2(config-router)# ospf router-id 10.16.17.2
ospfd-router2(config-router)# redistribute connected
ospfd-router2(config-router)# redistribute static
ospfd-router2(config-router)# network 10.16.17.0/29 area 0.0.0.0
ospfd-router2(config-router)# exit
ospfd-router2(config)# interface vlan17
ospfd-router2(config-if)# description Internal_interface
ospfd-router2(config-if)# ip ospf cost 100
ospfd-router2(config-if)# ip ospf hello-interval 8
ospfd-router2(config-if)# ip ospf dead-interval 24
ospfd-router2(config-if)# ip ospf retransmit-interval 4
ospfd-router2(config-if)# exit
ospfd-router2(config)# exit
ospfd-router2# write memory
Configuration saved to /usr/local/etc/quagga/ospfd.conf
```

Проверим теперь состояние ospf-сессии:

```
ospfd-router2# show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface	RXmtL	RqstL	DBsmL
10.16.17.1	1	Full/Backup	18.859s	10.16.17.1	vlan17:10.16.17.2	0	0	0

Проверим, какие маршруты получены по OSPF:

```
ospfd-router2# show ip ospf database
```

OSPF Router with ID (10.16.17.2)

Router Link States (Area 0.0.0.0)

Link ID	ADV Router	Age	Seq#	CkSum	Link count
10.16.17.1	10.16.17.1	497	0x80000005	0xf641	1
10.16.17.2	10.16.17.2	501	0x80000004	0xf63f	1

Net Link States (Area 0.0.0.0)

Link ID	ADV Router	Age	Seq#	CkSum
10.16.17.2	10.16.17.2	501	0x80000001	0xa303

AS External Link States

Link ID	ADV Router	Age	Seq#	CkSum	Route
10.16.16.0	10.16.17.1	502	0x80000002	0x5d0a	E2 10.16.16.0/24 [0x0]
10.16.32.0	10.16.17.1	502	0x80000002	0xaca	E2 10.16.32.0/24 [0x0]
10.16.33.0	10.16.17.1	502	0x80000002	0xa1b4	E2 10.16.33.0/24 [0x0]
192.168.1.0	10.16.17.2	501	0x80000003	0x8c98	E2 192.168.1.0/24 [0x0]

Базовая настройка завершена. Более подробно с протоколом OSPF можете ознакомиться [здесь](#) [3].



Источник (получено 2025-05-30 08:46):

<http://muff.kiev.ua/content/quagga-prevrashchenie-servera-v-polnofunktsionalnyi-marshrutizator>

Ссылки:

[1] <http://muff.kiev.ua/content/nastraivaem-border-gateway-protocol-c-pomoshchyu-quagga>

[2] <http://ru.wikipedia.org/wiki/OSPF>

[3] <http://muff.kiev.ua/files/books/RFC-2328.pdf>