

Простейшая установка и настройка PPPoE-сервера под Linux

Исправление ошибки LCP Config Request

Сергей Гулинов

Сегодня многие провайдеры предоставляют доступ в Интернет посредством аутентифицируемого VPN на базе протокола PPP. Однако далеко не всегда использование этого протокола является оптимальным решением.

Наверное, уже все сталкивались или по крайней мере слышали про виртуальные частные сети, VPN (Virtual Private Network). Под виртуальными частными сетями подразумевается создание логической сети поверх другой, например Интернет. При этом созданная логическая сеть использует в качестве среды передачи данных уже существующие сети.

При таком подходе протокол, используемый для создания частных сетей, инкапсулирует, т.е. вставляет свои пакеты в пакеты того протокола, который применяется в уже существующих сетях. Немного запутанно, не правда ли? Теперь более понятным языком.

В качестве протокола уже существующей сети пусть будет выступать наиболее широко используемый сегодня протокол IP (Internet Protocol – протокол сети Интернет). Тогда схема инкапсуляции будет выглядеть следующим образом (см. **рис. 1**).

Для создания виртуальной сети используется протокол PPP (Point-to-Point Protocol – протокол типа точка-точка). В качестве примера возьмём простой случай, когда данные протокола IP передаются по сети Ethernet. Тогда инкапсуляция будет выглядеть следующим образом (см. **рис. 2**).

Но и это ещё не всё. Дело в том, что пользовательские данные чаще всего передаются по виртуальным сетям, опять же по протоколу IP.

Получается этакий многослойный пирог (см. **рис. 3**).

Именно такая технология чаще всего используется у провайдеров для предоставления пользователям доступа в Интернет с использованием аутентифицируемого VPN как средства ограничения доступа к ресурсам глобальной сети.

Предположим, что нам надо создать виртуальную сеть между устройствами, находящимися в одной и той же се-

ти Ethernet, а так, кстати говоря, чаще всего и бывает. Тогда, я думаю, вы вполне со мной согласитесь, что развёртывать сеть IP, поверх которой будет работать ещё одна такая же сеть, не вполне разумно. Очевидно, что в данном случае оптимально развёртывать виртуальную сеть сразу поверх сети Ethernet, уменьшив тем самым число инкапсуляций одного протокола в другой. Вот тут нам на помощь и приходит протокол PPPoE (Point-to-point protocol over Ethernet) – сетевой протокол передачи кадров PPP через Ethernet. Этот протокол использует в качестве переносчика своей информации сеть Ethernet (см. **рис. 4**).

Таким образом, из нашего многослойного «пирога», исчезает один не нужный в данном случае уровень IP. Теперь перейдём от теории к делу и посмотрим, как настроить PPPoE-сервер под управлением Linux.

В качестве демонстрационной платформы будет выступать сервер под управлением Red Hat 9.0, однако всё написанное справедливо и для всех дистрибутивов от Red Hat, вплоть до Fedora Core 8.0 включительно.

Для настройки нам понадобятся последние версии PPP и PPPoE. На момент написания это ppp 2.4.4, скачать который можно здесь: <ftp://ftp.samba.org/pub/ppp/>, и rp-pppoe 3.8, который можно получить по адресу: <http://www.roaringpenguin.com/products/pppoe>.

Однако перед тем как описывать процесс установки, необходимо сделать небольшое отступление.

Дело в том, что на момент написания статьи на всех системах от Red Hat, PPPoE отказывался правильно функционировать в режиме демона (по поводу kernel mode ничего сказать не могу, потому как не пробовал). Ошибка кроется в демоне rpp, который при установлении соединения по протоколу PPPoE отказывается посылать ответы на клиентские запросы LCP Config Request, тем самым приводя к «зависанию» соединения на этапе проверки имени пользователя и пароля.

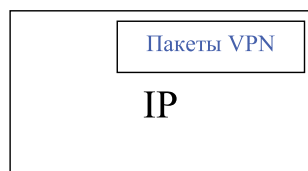


Рисунок 1. Схема инкапсуляции (IP)

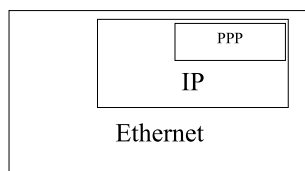


Рисунок 2. Схема инкапсуляции (PPP)

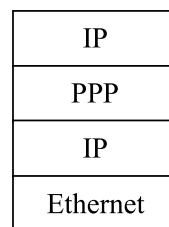


Рисунок 3. Схема передачи данных

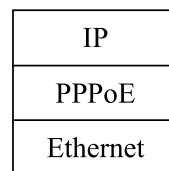


Рисунок 4. Схема передачи данных, убран протокол IP

При этом в логах, которые находятся в /var/log/messages, можно наблюдать такие ошибки:

```
Jan 30 03:24:51 Dark-server pppd[1979]: LCP: timeout sending Config-Requests
Jan 30 03:24:51 Dark-server pppd[1979]: Connection terminated.
Jan 30 03:24:51 Dark-server pppd[1979]: Modem hangup
```

Для устранения ошибки необходимо исправить файл main.c из комплекта rppd. После чего rppr надо заново откомпилировать и установить.

Итак, приступим.

Загружаем rppr 2.4.4, распаковываем, заходим в поддиректорию rppd, правим файл main.c (формат patch):

```
--- ppp-2.4.4/pppd/main.c.orig 2006-06-04 07:52:50.000000000 +0400
+++ ppp-2.4.4/pppd/main.c 2007-11-09 14:47:20.000000000 +0300
@@ -1567,6 +1567,8 @@
 if (errfd == 0 || errfd == 1)
 errfd = dup(errfd);

+ closelog();
+
/* dup the in, out, err fds to 0, 1, 2 */
if (infd != 0)
dup2(infd, 0);
@@ -1575,7 +1577,6 @@
if (errfd != 2)
dup2(errfd, 2);

- closelog();
if (log_to_fd > 2)
close(log_to_fd);
if (the_channel->close)
```

Скачать патч и более подробно о проблеме можно прочитать здесь: https://bugzilla.redhat.com/show_bug.cgi?id=222295.

Практически во всех дистрибутивах Linux демон rppr уже собран по умолчанию, и поэтому его необходимо заменить исправленной версией. Для этого выясняем, где располагается старая версия, дав команду:

```
which pppd
```

```
pppd: /usr/sbin/pppd
```

Мы видим, что старая версия находится в /usr/sbin/.

Далее, всё довольно обычно, как и при установке всех программ из исходных текстов. Запускаем скрипт конфигурации rppr, указывая параметр prefix, имея в виду, что установка должна выполняться, заменяя предыдущую версию. Компилируем, устанавливаем:

```
./configure --prefix=/usr
make
make install
```

На этом установка новой версии rppr завершена. И можно приступать к установке гр-ppрое.

Тут всё довольно тривиально и обычно. Загружаем гр-ppрое, распаковываем, запускаем скрипт установки:

```
./go
```

Если не будет никаких ошибок, то на этом установка завершена. Далее необходимо отредактировать файл опций /etc/ppp/pppoe-server-options. Это самый обычный файл конфигурации rppr. Вот пример такого файла, который работает у меня:

```
name pppoe
logfile /var/log/log.pppoe
require-pap
require-chap
require-mschap
require-mschap-v2
lcp-echo-interval 20
lcp-echo-failure 3
netmask 255.255.255.0
nobsdcomp
lock
```

Далее необходимо создать учетные записи пользователей, которым будет разрешен доступ к серверу. Имена пользователей, а также их пароли находятся в файлах: /etc/ppp/ppp-secrets и /etc/ppp/chap-secrets. Не вдаваясь в подробности, могу сказать, что эти файлы должны быть одинаковыми по содержанию. Приведу пример такого файла:

```
# Secrets for authentication using CHAP
# client      server      secret          IP addresses
"testlogin"   *          "testpass"      *
```

То есть именем пользователя будет «testlogin», а паролем «testpass». Ну вот практически и всё, теперь остаётся только запустить сервер, выполнив следующую команду:

```
pppoe-server -I eth0 -L 192.168.0.1 -R 192.168.0.2
```

где:

- eth0 – имя интерфейса, на котором запущен сервер;
- 192.168.0.1 – IP-адрес сервера;
- 192.168.0.2 – начальный IP-адрес, выдаваемый клиентам.

Всё, сервер запущен и работает.

Теперь к нему можно подсоединяться, используя в качестве имени пользователя «testlogin» и пароля «testpass».

Проконтролировать работу сервера после подключения к нему клиента можно, набрав следующую команду:

```
ifconfig
```

Ответ будет приблизительно такой:

```
eth0      Link encap:Ethernet  HWaddr 00:E0:4C:39:32:1A
          inet addr:192.168.22.200  Bcast:192.168.22.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1005 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1583 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:73417 (71.6 Kb)  TX bytes:191052 (186.5 Kb)
          Interrupt:5 Base address:0xf000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:10 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:700 (700.0 b)  TX bytes:700 (700.0 b)

ppp0      Link encap:Point-to-Point Protocol
          inet addr:192.168.0.1  P-t-P:192.168.0.2  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1480  Metric:1
          RX packets:14 errors:0 dropped:0 overruns:0 frame:0
          TX packets:9 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:1290 (1.2 Kb)  TX bytes:121 (121.0 b)
```

Среди запущенных интерфейсов можно увидеть интерфейс rpp0, который и является интерфейсом только что созданной нами сети VPN. Вот собственно и всё. Желаю успехов. 🍀