Расширенное администрирование FreeBSD.
Блок 7.
v 1.04

Оглавление

Настраиваем PPTP-сервер на базе FreeBSD

Сегодня перед системными администраторами все острее встает проблема обеспечения мобильных и удаленных пользователей полноценным и защищенным доступом к корпоративной сети. Благодаря встроенной поддержке туннельного протокола «точка-точка» в операционных системах Windows, одним из самых популярных решений является построение защищенных туннелей на основе PPTP. Настройкой такого сервера мы сегодня и займемся.

Протокол PPTP

Протокол PPTP (Point-to-Point Tunneling Protocol) позволяет создавать защищенные каналы для обмена данными по различным сетевым протоколам: IP, IPX или NetBEUI. Их данные инкапсулируются с помощью протокола PPTP в пакеты протокола IP, с помощью которого переносятся в зашифрованном виде через любую сеть TCP/IP. PPTP работает, устанавливая обычную PPP-сессию с противоположной стороной с помощью протокола туннелирования сетевых пакетов GRE (Generic Routing Encapsulation - общая инкапсуляция маршрутов). Для шифрования трафика применяется протокол MPPE (Microsoft Point-to-Point Encryption), использующий потоковый шифр RSA, RC4-ключи которого меняются в течение сессии.

Cisco первой реализовала PPTP в своих продуктах, она и лицензировала эту технологию корпорации Microsoft. Но не все так гладко. Несмотря на популярность, специалисты недолюбливают PPTP по причине слабых алгоритмов парольной аутентификации и возможности получения сессионных ключей на основе пользовательского пароля. Об этом можно почитать на сайте Брюса Шнаера (Bruce Schneier): www.schneier.com. Этот специалист занимается анализом реализации PPTP с 1998 года.

Если бы не встроенная поддержка в Windows, о PPTP, вероятно, уже давно бы все забыли. Хотя, с другой стороны, в Windows XP и более поздних версиях Windows присутствует возможность заменить пароли пользовательскими сертификатами, для этого с PPTP применяется протокол Extensible Authentication Protocol-Transport Layer Security (EAP-TLS).

Установка сервера PoPToP

Одной из популярных реализаций PPTP является сервер **PoPToP** (www.poptop.org). Изначально он написан для Linux, но без проблем работает в Solaris 2.6, OpenBSD, FreeBSD и других. Это первый проект, предоставивший возможность строить PPTP-серверы во FreeBSD. Он стартовал под руководством Matthew Ramsay и контролировался **Moreton Bay Ventures** (www.moretonbay.com). В марте 1999 года PoPToP был опубликован под лицензией GNU. Он совместим со всеми версиями Windows и никсовым PPTP-клиентом (pptpclient.sf.net). Поддерживает аутентификацию MSCHAPv2 и шифрование MPPE 40 с 128-битным RC4. Легко интегрируется в сети Windows.

Настройка PPTP-сервера

Для начала устанавливаем всё необходимое:

```
cd /usr/ports/net/poptop
make install
rehash
```

Далее приступаем к настройке. Всё достаточно просто. Первым делом открываем в редакторе файл /etc/ppp/ppp.conf и дописываем в конец следующие строки:

```
pptp:
 set ifaddr 192.168.1.X 192.168.1.100-192.168.1.200
 enable mschapv2
 set timeout 0
```

Следующим шагом записываем в файл /usr/local/etc/pptpd.conf:

```
noipparam
```

Далее открываем в редакторе файл /etc/ppp/ppp.secret и записываем в него:

```
user1   1234
```

Разрешаем запуск pptp-сервера в /etc/rc.conf:

```
pptpd_enable="YES"
```

Скорее всего на сервере стоит файрволл. Добавим в скрипт с его настройками несколько строк:

```
# Разрешаем протокол GRE для всех;
pass quick inet proto gre to any keep state
# Разрешаем соединение с PPTP-сервером для всех;
pass quick inet proto tcp to any port 1723 keep state
```

После этого перезапускаем pptpd:

```
/usr/local/etc/rc.d/pptpd start
```

На этом настройка PPTP-сервера заканчивается. Для подключения из под Windows можно воспользоваться мастером настройки сети. В качестве сервера ("шлюза") нужно указать внешний адрес нашего сервера.

Настройка PPTP-клиента

Во-первых нужно установить PPTP-клиента:

```
cd  /usr/ports/net/pptpclient
make install
rehash
```

Добавляем в файл /etc/ppp/ppp.conf следующие настройки для клиента:

```
pptpclient:
 set authname user1
```

```
set authkey 1234
set timeout 0
set ifaddr 0 0
accept mschapv2
add default HISADDR
alias enable yes
```

Теперь выполняем команду:

```
pptp 172.16.1.20 pptpclient 2> /dev/null
```

И всё работает. В системе должен появиться новый tun-интерфейс. Проверить это можно командой:

```
ifconfig | grep tun
```

Настройка клиентского соединения в Windows

Настройка PPTP-соединения практически ничем не отличается от подключения к провайдеру. Вызываем «Сетевые подключения», выбираем «Создание нового подключения» и следуем указаниям мастера. Во втором окне отмечаем пункт «Подключить к сети на рабочем месте» и в следующем – «Подключение к виртуальной частной сети», затем вводим название подключения и указываем, необходимо ли набирать номер для предварительного подключения. Если соединение осуществляется напрямую, то выбираем «Не набирать номер для предварительного подключения» и вводим IP-адрес или имя сервера, к которому необходимо подключиться. После нажатия кнопки «Готово» можно пробовать подключиться к серверу, введя логин и пароль. В зависимости от версии и настроек сервера, а также версии клиентской операционной системы, возможно, потребуется уточнить некоторые параметры подключения (протокол, обязательность шифрования, сжатие и другие), для чего необходимо выбрать «Свойства» созданного соединения.

NFS-сервер и NFS-клиент

Network File System (NFS) — это сетевая файловая система, позволяющая пользователям обращаться к файлам и каталогам, расположенным на удалённых компьютерах, как если бы эти файлы и каталоги были локальными. Главным преимуществом такой системы является то, что отдельно взятые рабочие станции могут использовать меньше собственного дискового пространства, так как совместно используемые данные хранятся на отдельной машине и доступны для других машин в сети. NFS - это клиент-серверное приложение. Т.е. в системе пользователя должен быть установлен NFS-клиент, а на компьютерах, которые предоставляют свое дисковое пространство - NFS-сервер. Здесь вы увидите, как просто установить и настроить эти программы в freebsd.

Настройка NFS-сервера

Разрешаем запуск NFS-серверav в /etc/rc.conf**:**

```
…
nfs_server_enable="YES"
nfs_reserved_port_only="YES"
rpcbind_enable="YES"
rpc_lockd_enable="YES"
rpc_lockd_flags="-p 884"
rpc_statd_enable="YES"
rpc_statd_flags="-p 885"
mountd_enable="YES"
mountd_flags="-p 883"
…
```

Добавляем разрешения в /etc/pf.conf:

```
# разрешить запросы к серверу NFS и RPCBIND
pass quick inet proto { tcp, udp } from any to port { nfsd, rpcbind } keep state
# mountd -p 883
pass quick inet proto { tcp, udp } from any to port 883 keep state
# rpc.lockd -p 884
pass quick inet proto { tcp, udp } from any to port 884 keep state
# rpc.statd -p 885
pass quick inet proto { tcp, udp } from any to port 885 keep state
```

Запускаем все необходимые службы:

```
/etc/rc.d/rpcbind start
/etc/rc.d/mountd start
/etc/rc.d/nfsd start
```

Настраиваем, какие именно директории мы хотим открыть для совместного пользования и кому. Все это делается в файле /etc/exports:

```
/usr/ports  -ro  -network 172.16.1.0 -mask 255.255.255.0
```

Теперь нужно применить изменения:

```
/etc/rc.d/mountd reload
```

Все. NFS-сервер установлен и настроен.

Монтирование NFS-ресурса

Создаем точку монтирования. Допустим, это будет папка в вашей домашней директории с названием data:

```
cd ~
mkdir data
```

Теперь можно монтировать:

```
mount -t nfs 172.16.1.20:/usr/ports  ~/data
```

Если все прошло успешно, то набрав в терминале:

```
cd ~/data
ls
```

вы увидите содержимое папки /usr/ports, находящейся на NFS-сервере.