Расширенное администрирование Linux.
Блок 7.

v 1.02

Оглавление

Настраиваем серверы PPTP и NFS на базе Linux

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. Из-за опасений по поводу патентных претензий протокола MPPE до недавнего времени в дистрибутивах Linux отсутствовала полноценная поддержка PPTP. Однако, начиная с версии 2.6.13, появилась полная поддержка PPTP.

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 в Linux

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 в Linux. 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. При использовании RADIUS легко интегрируется в сети Windows.

Настройка PPTP-сервера в Ubuntu

В качестве исходных данных будем использовать: офисный интернет-шлюз под управлением Ubuntu Server 8.10. Для начала устанавливаем всё необходимое:

```
apt-get install pptpd
```

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

```
# IP-адрес PPTP-сервера
localip 10.0.0.1

# Диапазон адресов для клиентов PPTP-сервера
remoteip 10.0.0.2-254
```

Следующим шагом дописываем в файл /etc/ppp/pptpd-options следующие две строчки:

```
# требуем авторизацию у клиентов
auth

# Используем шифрование
require-mppe-128
```

Ну и наконец открываем в редакторе файл /etc/ppp/chap-secrets и заполняем строчками вида:

```
# Если пользователь должен динамически получать IP-адрес
# из диапазона remoteip в pptpd.conf:
user1   pptpd   1234       "*"

# Если мы хотим привязать определённый IP  к логину:
user2   pptpd   1234       "10.0.0.101"
```

После этого перезапускаем pptpd:

```
/etc/init.d/pptpd restart
```

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

```
# Разрешаем протокол GRE для всех;
iptables -A INPUT -p gre -j ACCEPT

# Разрешаем соединение с PPTP-сервером для всех;
iptables -A INPUT -p tcp --dport 1723 -j ACCEPT
```

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 в Ubuntu

Во-первых нужно установить PPTP-клиента:

```
apt-get install pptp-linux
```

Во-вторых создаём файл /etc/ppp/peers/vpn0 следующего содержания:

```
pty "pptp 192.168.2.1 --nolaunchpppd"
name user1
file /etc/ppp/options.pptp
remotename PPTP
ipparam vpn0
```

Следующим шагом раскомментируем в файле /etc/ppp/options.pptp следующую строчку:

```
require-mppe-128
```

Далее открываем в редакторе файл /etc/ppp/chap-secrets и добавляем строку:

```
user1   PPTP    1234 "192.168.2.1"
```

Наконец выполняем команду:

```
pon vpn0
```

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

```
ifconfig | grep ppp
```

Если же соединения не происходит, то можно попытаться выполнить команду:

```
pon vpn0 debug dump logfd 2 nodetach
```

И посмотреть какие ошибки будут выданы на экран.

Если нужно чтобы соединение выполнялось автоматически при загрузке компьютера, то нужно добавить в файл /etc/network/interfaces строки:

```
auto tunnel
iface tunnel inet ppp
provider vpn0
```

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 в Ubuntu

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 Ubuntu Linux.

Установка и настройка NFS-сервера

Устанавливаем NFS-сервер:

```
apt-get install nfs-kernel-server nfs-common portmap
```

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

```
nano /etc/exports
```

В приведенном ниже примере я выделил директорию /data (директория с данными на сервере) в совместное пользование всем компьютерам с IP - 10.10.103.1 - 10.10.103.255 с правами чтения и записи:

```
/data 10.10.103.0/24(rw,all_squash,async)
```

Или еще пример:

```
/home/sboronin/ 10.10.103.2(ro,async)
```

домашняя директория пользователя sboronin стала доступной в режиме только чтение для компьютера с IP 10.10.103.2. Все остальные компьютеры сети к этому разделу доступа не имеют.

Опции:
ro - права только на чтение. Можно и не указывать, так как она установлена по умолчанию.
rw - дает клиентам право на запись.
all_squash — заменять всех пользователей на пользователя с UID=65534.
noaccess - запрещает доступ к указанной директории. Может быть полезной, если перед этим вы задали доступ всем пользователям сети к определенной директории, и теперь хотите ограничить доступ в поддиректории лишь некоторым пользователям.
О других опция можно почитать здесь.

Теперь нужно перезапустить nfs-kernel-server:

```
/etc/init.d/nfs-kernel-server restart
```

Если после этого вы захотите поменять что-нибудь в файле /etc/exports, то для того, чтобы изменения вступили в силу, достаточно запустить следующую команду:

```
exportfs -a
```

Все. NFS-сервер установлен и настроен. Можно переходить к настройке NFS клиента.

Установка и настройка NFS-клиента

Установка:

```
apt-get install portmap nfs-common
```

Монтирование:

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

названием data:

```
cd ~
mkdir data
```

Монтировать можно двумя способами - каждый раз вручную или прописав опции монтирования в файл /etc/fstab. Мне больше нравится второй способ:

```
nano /etc/fstab
```

и в конце дописываем следующую строку:

```
10.10.103.1:/data /data nfs rw,hard,intr 0 0
```

Вместо 10.10.103.1:/data впишите IP или имя сервера и путь к директории совместного пользования.
Опции монтирования можно изменить.

Записав и сохранив изменения, можно монтировать:

```
mount /data
```

Если все прошло успешно, то набрав в терминале

```
cd /data
ls
```

вы увидете содержимое папки /data, находящейся на NFS-сервере. Можете проверить скорость работы и запустить, например, фильм с этого раздела.