Визитка

РАШИД АЧИЛОВ, поклонник FreeBSD с многолетним опытом использования ее в совмещенных с Windows сетях и сторонник Open Source. Администратор сетей и средств защиты крупной торговой сети

Построение корпоративных VPN

Использование IPSec для связи с аппаратным роутером

В третьей части статьи рассматривается использование IPSec для подключения аппаратного роутера со статическим и динамическим адресами. Приводятся также скрипты, о которых упоминалось в предыдущей части

Маршрутизатор со статическим адресом

Противостояние аппаратного и программного маршрутизатор – тема столь же неисчерпаемая на всевозможных форумах, как и Linux vs Windows. У каждой стороны в этом бесконечном споре есть свои сторонники, есть свои противники и все аргументы, в общем-то, заслуживают внимания. Недавно мне на деле пришлось столкнуться с плюсами и минусами решения на аппаратном маршрутизаторе, и, понятное дело, невозможно построить сколько-нибудь развернутую сеть так, чтобы в ней не было ни одного такого устройства.

Для определенности будем использовать одно из самых недорогих устройств – DI-804HV. Настраивается оно достаточно просто, и нас в основном будет интересовать, не что нужно вписывать, а почему нужно вписывать именно это. Как включить устройство, как попасть в веб-интерфейс настройки, как настроить LAN- и WAN-адреса устройства, изложено в паспорте на устройство, скачать который можно с сайта [1].

Переходим в раздел VPN, разрешаем работу модуля, вводим имя туннеля (см. рис. 2).

Можно выбрать два метода настройки – IKE и Manual. В зависимости от выбранного варианта будут доступны разные установки на основном экране, появляющемся при выборе по кнопке «More». Первый метод – Manual – в точности совпадает с тем, что обсуждалось в предыдущей части статьи, – все параметры (номера SPI, виды протоколов шифрования и аутентификации, ключи шифрования) необходимо задавать вручную. Доступны только протоколы шифрования DES и 3DES. Второй – IKE – ориентирован на использование протокола IKE. Настройки, задаваемые здесь, мы рассмотрим более подробно.

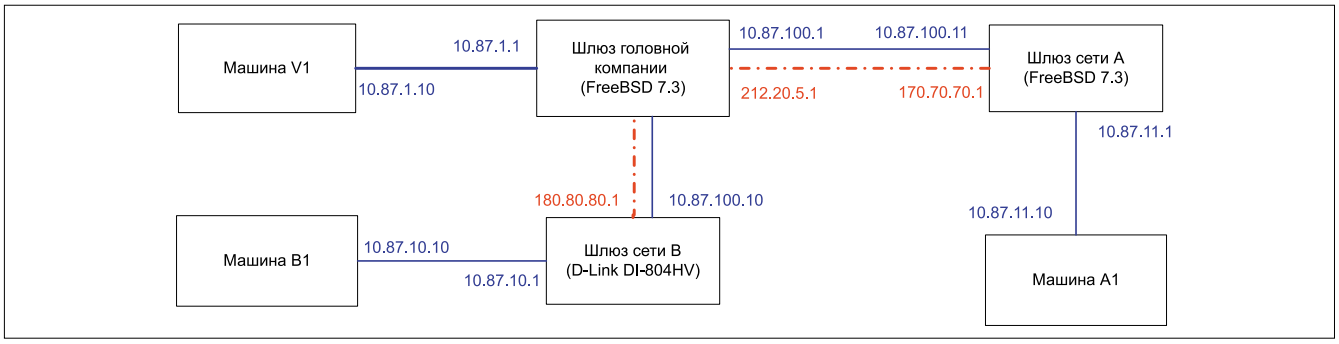
Aggressive Mode – ну, с этим вопросов быть не должно – включает использование режима aggressive (напоминаю, что режим main не работает при использовании динамических адресов).

Local Subnet/Local Netmask, а также Remote Subnet/Remote Netmask – соответственно локальная внутренняя (туннелируемая) подсеть и ее маска и удаленная внутренняя подсеть и ее маска. Эти подсети мы и связываем туннелем.

Remote Gateway – реальный IP-адрес удаленной стороны, куда будет подключаться маршрутизатор. Racoon на «той» стороне должен содержать секцию remote с соответствующим адресом или секцию anonymous (или и то, и другое).

IKE Keep Alive – адрес в удаленной внутренней подсети, на который будут отправляться пакеты, поддерживающие соединение (используется обычный ICMP Echo Request/Echo Reply). Внимательнее относитесь к выбору адреса для

Рисунок 1. Схема имитационной модели VPN



Keep Alive – если он вдруг окажется недоступен, маршрутизатор будет постоянно перезапускать VPN.

Preshare Key – ну это, конечно же, общий ключ из файла psk.txt, расположенного в каталоге /usr/local/etc/racoon на стороне сервера. Ключи должны соответствовать друг другу. Если при наборе ключа случайно ошибетесь, соединение не сможет установиться, причем в журнале racoon не будет никакой внятной диагностики по этому поводу. Так что, если все вроде правильно, а соединение не устанавливается, проверьте общий ключ!

Extended Authentication (xAUTH) – расширенные методы аутентификации. Позволяют задать имя пользователя и пароль, которые на стороне сервера будут проверены через PAM, LDAP-сервер или по локальной базе пользователей. При наличии статического IP-адреса в его применении нет необходимости, но при использовании динамического адреса из него можно извлечь пользу (пример приведу далее).

IPSec NAT Traversal – включает поддержку протокола NAT-T, который во FreeBSD 7.x добавляется отдельным патчем на ядро, а во FreeBSD 8.x уже встроен в систему.

Auto-Reconnect – если связь оборвалась, восстановить.

Remote ID, Local ID – аналоги параметров my_identifier и peers_identifier в racoon, задают идентификаторы удаленной и локальной сторон. При использовании режима main могут быть только типа address (IP Address). Для того чтобы применять идентификаторы FQDN и User_FQDN, необходимо выбрать aggressive-режим.

IKE Proposal Index – позволяет для Phase1 выбрать параметры установления соединения (на сервере обычно описываемые в подсекции proposal секции remote).

IPSec Proposal Index – точно так же позволяет определить параметры для Phase2 (на сервере обычно задаваемые в секции или секциях sainfo).

Внутри страницы IKE Proposal можно настроить те параметры, которые обычно задаются в подсекции proposal секции remote, – DH Group (dh_group), Encrypt algorithm (encryption_algorithm), Auth algorithm (hash_algorithm), Life Time (lifetime). Правда, выбор куда более скромный – Encrypt только 3DES и DES, Auth только SHA1 и MD5.

Возможности страницы IPSec Proposal позволяют настроить параметры, которые обычно задаются в секции sainfo, – DH Group (pfs_group), Encap protocol (у этого параметра нет соответствующего параметра в racoon.conf, он задается непосредственно в SP, в D-Link здесь можно задать ESP или AH), Encrypt algorithm (encryption_algorithm), Auth algorithm (hash_algorithm), Life Time (lifetime). Выбор скромный – Encrypt только 3DES и DES, Auth только SHA1, MD5 или None. Создаваемая на сервере SA прекратит свое действие именно по истечении задаваемого здесь, а не в IKE Proposal lifetime.

После ввода всех параметров настраиваете маршрутизацию по [1], указываете дополнительные значения и перегружаете устройство. Туннель должен установиться сразу, на странице статуса в разделе VPN status появится надпись IKE established. Если же там надпись IKE establishing, нужно посмотреть журнал устройства – там могут оказаться полезные сообщения об ошибках. Обычные причины того, что туннель не устанавливается, – неверные ключ и идентификаторы, различия в lifetime или DH group.

Для создания интерфейсов туннелей и настройки маршрутизации используются скрипты, запускаемые в начале и окончании Фазы 1 (Phase1 up и Phase1 down). Скрипты приведены после описания подключения с динамическим адресом.

Маршрутизатор с динамическим адресом

Усложним задачу. Предположим, необходимо подключить устройство к сети, где не выдают статических IP-адресов (например, случай временного подключения в момент аварии основного канала). И вот тут нам потребуется передать на сервер идентификацию маршрутизатора, чтобы правильно настроить создаваемый туннель и роутинг. Если в случае со статическими адресами все можно было настроить заранее, то с адресами динамическими это работать не будет. А если внешний адрес устройства (на схеме 180.80.80.1) передается при запуске скрипта самой программой racoon, то как сообщить информацию о том, какая у устройства внутренняя сеть и какой его адрес в этой сети? Вот здесь нам и поможет xAUTH.

Рисунок 2. Общие настройки модуля VPN D-Link

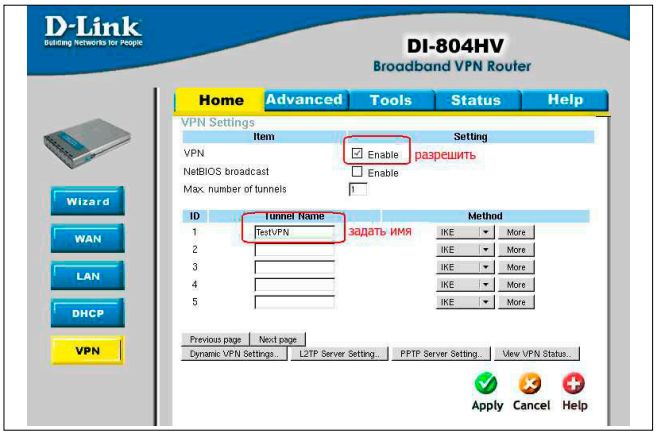
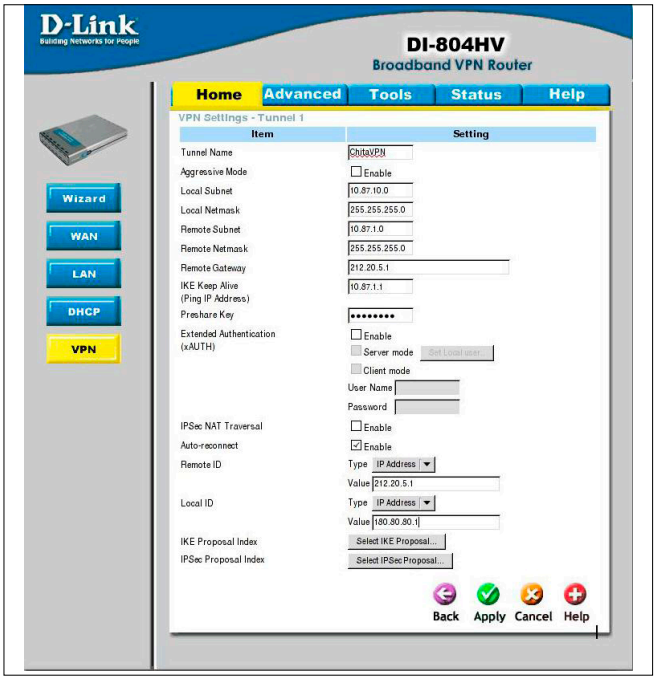


Рисунок 3. Настройки VPN при использовании IKE



На странице настроек VPN (см. рис. 3) включаем Extended Authentication, Client mode и в полях User Name и Password задаем имя и пароль пользователя. Имя должно присутствовать в файле /usr/local/etc/racoon/extnodes (что это за файл и зачем он нужен, описано ниже) на сервере. Пароль может быть настоящим, действующим, если хотите реально проверять наличие данного пользователя в какой-либо базе, или же бессмысленным текстом, если не хотите. Так как с динамическим адресом main mode не работает, на странице настроек включаем Aggressive Mode. Собственно, все. Все остальные «магические пассы» проводятся на сервере.

Серверная часть

Общие секции конфигурационного файла racoon.conf (padding{}, listen{}, timer{}, mode_cfg{}) и описание путей я приводить не буду – они рассмотрены в предыдущей статье. Приведу только секции remote anonymous{}, remote 180.80.80.1{} и sainfo anonymous{}.

remote anonymous

```
{
    exchange_mode main,aggressive;
    doi ipsec_doi;
    situation identity_only;
    nonce_size 16;
    lifetime time 24 hour;
    initial_contact on;
    proposal_check strict;
    generate_policy unique;
    ike_frag on;
    passive on;
    dpd_delay 0;
    script "pam_linkup.sh" phase1_up;
    script "linkdown.sh" phase1_down;
    proposal {
        encryption_algorithm 3des;
        hash_algorithm sha1;
        authentication_method xauth_psk_server;
        dh_group 1;
        lifetime time 28800 sec;
    }
}
```

Exchange mode задан таким, чтобы работало подключение с динамическим адресом, который заранее неизвестен, и, следовательно, подключение будет обслуживаться секцией anonymous. Generate_policy обеспечит возможность не задавать на сервере SP, которые будут сгенерированы автоматически. Script phase_1up и script phase1_down обеспечат возможность запуска некоторых скриптов в тот момент, когда установлена или завершена Фаза 1 соединения. В подсекции proposal задаются параметры, которые для D-Link устанавливаются при настройке IKE Proposal (см. рис. 4). Hash_algorithm – это Auth Algorithm. Обратите внимание на authentication_method – указан xauth_psk_server для того, чтобы работала возможность xAUTH.

remote 180.80.80.1 inherit anonymous

```
{
    exchange_mode main;
    my_identifier address 212.20.5.1;
    peers_identifier address 180.80.80.1;
    verify_identifier on;
    script "linkup.sh" phase1_up;
    proposal {
        encryption_algorithm 3des;
        hash_algorithm sha1;
        authentication_method pre_shared_key;
        dh_group 1;
        lifetime time 28800 sec; }
}
```

При статическом адресе aggressive mode можно не использовать, поэтому он отключается. Включается проверка идентификаторов, изменяются имя скрипта и метод аутентификации – xAUTH необходим только при применении динамического адреса.

sainfo anonymous

```
{
    pfs_group 1;
    lifetime time 3600 sec;
    encryption_algorithm 3des;
    authentication_algorithm hmac_md5;
    compression_algorithm deflate;
}
```

В этой секции задаются параметры, соответствующие странице IPSec Proposal в настройках D-Link, в частности,

Рисунок 4. Настройки IKE Proposal

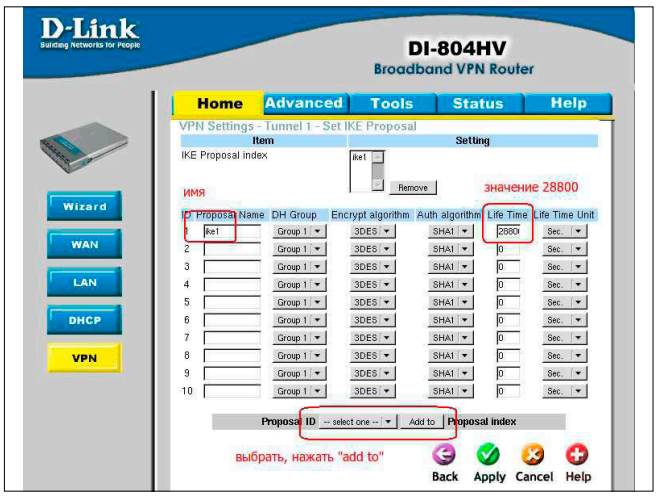
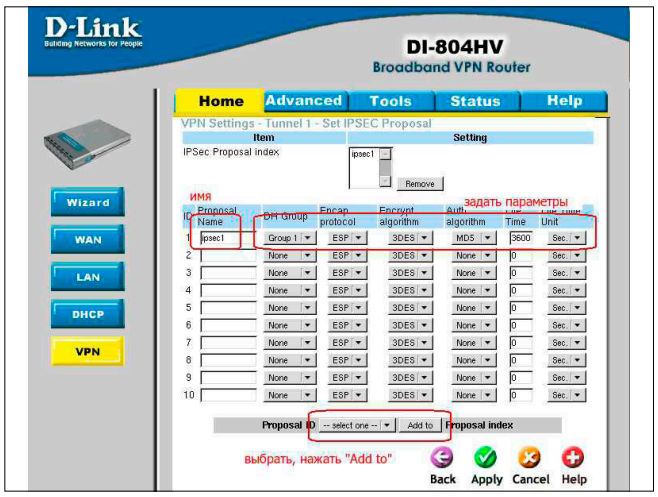


Рисунок 5. Настройки IPSec Proposal



именно здесь устанавливается таймер, по истечении которого будут удалена SA и повторно установлено соединение.

А теперь о том, зачем нам вообще нужны скрипты, запускаемые racoon. На самом деле из трех компонентов VPN – Virtual Private Network – IPSec вообще и racoon в частности обеспечивают только среднюю – Private. Создание виртуальных интерфейсов (Virtual), а также объединение их в сеть и маршрутизация в этой сети (Network) – задача администратора сети. И если для D-Link можно написать инструкцию по настройке, действуя по которой, ошибиться нельзя, то для сервера, к которому он подключается, все приходится делать вручную. Поэтому мною было разработано несколько скриптов, а именно:

linkup.sh – скрипт, выполняемый на сервере при подключении со статическим адресом;

pam_linkup.sh – скрипт, выполняемый на сервере, при подключении с динамическим адресом;

linkdown.sh – скрипт, выполняемый на сервере, при отключении оконечной точки (неважно с каким адресом).

Все скрипты имеют общую часть, которая совершенно одинакова – содержит проверку параметров, чтение конфигурационного файла и создание рабочего каталога в случае его отсутствия. Приводится упрощенный вариант скриптов, без диагностического вывода, полный вариант можно скачать с [2].

```
PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin: ↵
        /usr/local/sbin
config="/usr/local/etc/racoon/extnodes.conf"
if [ -e $config ]; then
  . $config
fi
if [ ! -e $connected_spool ]; then
  mkdir $connected_spool
  chmod 0700 $connected_spool
fi
connfile="$connected_spool/$REMOTE_ADDR"
spindownfile="$connected_spool/$REMOTE_ADDR.spindown"
spinupfile="$connected_spool/$REMOTE_ADDR.spinup"
```

Каждое устройство, будь то D-Link или шлюз на базе FreeBSD, кроме внешнего адреса, имеет еще внутренний – в той подсети, которую нужно связать с удаленной, а также собственно адрес и маску этой подсети. Эти данные каким-то образом нужно получить и привязать к конкретному подключающемуся устройству. В качестве идентификаторов, по которым выполняется поиск для устройств со статическим адресом, выступают, конечно же, IP-адреса, а с динамическим – имена пользователей, передаваемые с помощью xAUTH. Поиск выполняется в простом текстовом файле, который называется extnodes и располагается там же, где и racoon.conf. Формат файла очень простой:

```
<ID> <внешний_адрес> <внутренний_адрес> <внутренняя_подсеть>
```

Для статических адресов <ID> не используется, и обычно в него записывается «static». Для динамических адресов соответственно не задействован <внешний_адрес>, и в него записывается «dynamic». Таким образом, для обоих узлов схемы при применении статических и динамических адресов файл будет содержать четыре записи:

```
static    170.70.70.1    10.87.11.1  10.87.11.0/24
static    180.80.80.1    10.87.10.1  10.87.10.0/24
gate11    dymanic        10.87.11.1  10.87.11.0/24
gate10    dynamic        10.87.10.1  10.87.10.0/24
```

Для задания собственных параметров сетевых интерфейсов сервера, путей и других настроек существует конфигурационный файл extnodes.conf. Настроек в нем немного:

```
# Наш внешний сетевой адрес (ESG)
esg_address="87.103.172.93"
# Наш внутренний сетевой адрес (RNG)
rng_address="10.38.5.254"
# Наша подсеть и маска в формате CIDR
int_network="10.38.5.0/24"
# Файл со списком узлов
extnodes_file="/usr/local/etc/racoon/extnodes"
# Каталог для сохранения данных о подключившихся клиентах
connected_spool="/var/spool/racoon"
# Имя файла журнала
logfile="/var/log/pam_linkup"
# Параметры для вывода в журнал в виде facility.priority
#(см. man syslog.conf)
log_priority="local1.info"
```

Основная часть скрипта linkup.sh следующая:

```
# Если рабочий каталог racoon еще не существует, создаем его

if [ ! -e $connected_spool ]; then
  mkdir $connected_spool
  chmod 0700 $connected_spool
 else
   while [ -e $spinupfile ]
    do
     lockupprintdate=`ls -l -D "%T %d-%m-%Y" $spinupfile`
     lockupcheckdate=`ls -l -D "%s" $spinupfile`
     nowcheckdate=`date "+%s"`
     diffdate=$(($nowcheckdate-$lockupcheckdate))
     if [ $diffdate -le 300 ]; then
       exit
     fi
     set $lockupprintdate
     sleep 1
    done
   while [ -e $spindownfile ]
    do
     lockdownprintdate=`ls -l -D "%T %d-%m-%Y" $spinupfile`
     set $lockdownprintdate
     sleep 1
    done
fi
touch $spinupfile
devline=`cat $extnodes_file | grep $REMOTE_ADDR`
if [ ${#devline} -eq 0 ]; then
  rm -f $spinupfile
  exit 113
 else
   set $devline
   _lng=$3
   _lns=$4
fi
ifconfig gif create > $connfile
_iface=`cat $connfile`
ifconfig $_iface tunnel $esg_address $REMOTE_ADDR
ifconfig $_iface inet $rng_address ↵
        $_lng netmask 255.255.255.255 mtu 1450
route add -net $_lns $_lng
echo $_lns >> $connfile
```

Основная часть скрипта linkdown.sh следующая:

```
touch $spindownfile
if [ -e $spinupfile ]; then
  rm -f $spinupfile
fi
cmdfile=`mktemp $connected_spool/$REMOTE_ADDR.XXXXXX`
echo "deleteall $esg_address $REMOTE_ADDR esp;" >> $cmdfile
echo "deleteall $REMOTE_ADDR $esg_address esp;" >> $cmdfile
setkey -f $cmdfile
if [ -e $connfile ]; then
  _iface=`head -n 1 $connfile`
  _lns=`tail -n +2 $connfile`
```

```
  route delete $_lns
  ifconfig $_iface down
  ifconfig $_iface destroy
  rm -f $connected_spool/$REMOTE_ADDR
fi
rm -f $spindownfile
rm -f $cmdfile
```

Обратите внимание на два момента.

- Мы не создавали новых SP при обработке подключения клиента – параметр generate_policy unique обеспечивает создание SP автоматически. Поэтому на серверной стороне обычно /etc/ipsec.conf почти пустой.
- При обработке отключения клиента мы вручную удаляем все SA. Зачем? Эксперименты показали, что при переходе SA из состояния mature в состояние dying (т.е. тогда, когда SA скоро будет удалена по истечении таймера) может появиться «посторонняя» SA, которая впоследствии помешает нормальному восстановлению связи, поэтому мы подстрахуемся и удалим все SA, связанные с данными адресами.

А в чем же суть «магических пассов» при использовании динамической адресации? Здесь нас выручает такая хорошо известная вещь, как PAM. Создаем файл сервиса, называем его racoon, в секцию auth (которая вызывается два раза, видимо, по числу создаваемых политик) вписываем вызов модуля, который будет проверять пароль (pam_unix.so, pam_winbindd.so), или pam_permit.so, если проверка не нужна, а в секцию account – вызов pam_linkup.sh через pam_exec.so.

```
auth    required  pam_permit.so
account required  pam_exec.so   /usr/local/etc/racoon/ ↵
                                 pam_linkup.sh
```

Идентификация устройства производится через имя пользователя, задаваемое в xAUTH. Это имя, а также IP-адрес подключившегося узла передаются в параметрах окружения, формируемых через PAM, – PAM_USER и PAM_RHOST. PAM_USER используется для поиска дополнительных параметров устройства, а PAM_RHOST – для команд создания туннеля. По сути дела, pam_linkup.sh – это тот же linkup.sh, только вместо REMOTE_ADDR здесь используется PAM_RHOST.

```
--- linkup.sh   2011-02-15 23:33:03.000000000 +0600
+++ pam_linkup.sh       2011-02-17 23:15:30.000000000 +0600
@@ -33,13 +33,13 @@
 if [ -e $config ]; then
   . $config
 fi
-connfile="$connected_spool/$REMOTE_ADDR"
-spinupfile="$connected_spool/$REMOTE_ADDR.spinup"
-spindownfile="$connected_spool/$REMOTE_ADDR.spindown"
+connfile="$connected_spool/$PAM_RHOST"
+spinupfile="$connected_spool/$PAM_RHOST.spinup"
+spindownfile="$connected_spool/$PAM_RHOST.spindown"
 if [ ! -e $connected_spool ]; then
@@ -68,13 +68,13 @@
 touch $spinupfile
-devline=`cat $extnodes_file | grep $REMOTE_ADDR`
+devline=`cat $extnodes_file | grep $PAM_USER`
 if [ ${#devline} -eq 0 ]; then
@@ -89,7 +89,7 @@
_iface=`cat $connfile`
-ifconfig $_iface tunnel $esg_address $REMOTE_ADDR
+ifconfig $_iface tunnel $esg_address $PAM_RHOST
  status=$?
```

Зачем нам понадобились такие сложные манипуляции с файловыми блокировками и т.д.? Практика показала, что скрипты racoon запускает асинхронно, то есть запустил скрипт и тут же перешел к обработке следующего сообщения. Теперь представим, что существующая SA закончила срок действия. Racoon посылает на удаленную сторону сообщение, удаленная сторона выполняет Phase1_down, то есть запускает соответствующий скрипт... и тут же переходит к обработке следующего сообщения, которое при плотном потоке данных окажется на удаленной стороне мгновенно. Им, конечно же, станет Phase1_up, и в результате скрипт linkup, который должен бы запуститься, когда соединение отсутствует, будет запущен одновременно с linkdown, а то и раньше. Более того, если за определенный интервал времени соединение установлено не будет, то устройство пошлет еще один запрос на Phase1_up, который, конечно же, снова вызовет запуск linkup. В результате этого в памяти может находиться четыре-пять копий linkup, которые будут запускаться по мере того, как отработает более ранняя копия.

Поэтому при запуске сначала проверяется, не выполняется ли сейчас другая копия скрипта. Если файл блокировки существует, то проверяется дата его создания. Если она отличается от текущей даты менее чем на 300 секунд (5 минут), скрипт прекращает работу, считая это ошибочным повторным запуском. Если же больше, ждет, пока не будет удален файл блокировки linkup. Если существует файл блокировки linkdown, это означает, что linkup запустился раньше linkdown, но для нормальной работы должен дождаться его завершения. И только если нет никаких блокировок, linkup сам создает файл блокировки, который сохраняется все время, пока не запустится linkdown (который запустится по Phase1_down).

Ну что ж. Аппаратные маршрутизаторы мы готовы подключать в любых вариантах – и со статическими адресами, и с динамическими. Необходимо только помнить, что PAM работает только в том случае, если есть секция mode_cfg{} и в ней задан параметр auth_source pam. Блок mode_cfg{} должен присутствовать даже тогда, когда использование IPSec-листов (компьютеров или устройств под управлением Windows, Symbian и подобных ОС, не являющихся шлюзами) не планируется.

***

Охватить все или хотя бы основные возможности IPSec в рамках одной-двух-трех статей очень сложно. Мы рассмотрели только подключение аппаратных роутеров, являющихся одной из основ для построения VPN, с различными вариантами адресации и аутентификации. Это уже позволит построить основной скелет сети – объединить все точки присутствия (офисы, магазины, склады), но пока не позволит подключить программные шлюзы (где работает тот же racoon), а также произвольных клиентов с ноутбуками, нетбуками или мобильными устройствами. Развитие темы – в продолжении. EOF

1. Мануад для DI-804HV – http://ftp.dlink.ru/pub/Router/DI-804HV/Description/DI-804HV_Manual_RUS.rar.
2. Полные версии всех упоминаемых скриптов – http://openoffice.mirahost.ru/scripts/racoon.tar.bz2.