Резервирование каналов передачи и уведомление пользователей о переключении на резервного провайдера

Сычёв Андрей
Украина, Киев

www.mikrotik.net.ua
trainer@mikrotik.net.ua

Сычёв Андрей Владимирович

Сертификаты:

MTCNA, MTCRE, MTCWE, MTCTCE

Тренер

Постановка задачи:

Необходимо подключить филиалы к центральному офису и обеспечить резервирование каналов передачи.

Должна быть обеспечена бесперебойная и безопасная работа с серверными приложениями в ЦО и электронной почты.

Доступ в Интернет предоставляется только при работе на основном канале.

Решение задачи:

1. При недоступности маршрутизатора ЦО по основному каналу, переключаемся на резервный
2. Строим SSTP туннель в ЦО, настраиваем маршрутизацию

Проверка доступности маршрута

RouterOS имеет встроенную возможность проверять доступность маршрута пингуя IP адрес шлюза, но в случае когда проблема находится в сети провайдера или за ней этот метод не работает.

Скрипты и Netwach + скрипт

Для решения этой проблемы пользователи написали множество скриптов, некоторые использовали Netwatch + скрипт переключения маршрута.

Достоинством этих решенией является отсутствие т.н. “фатального недостатка”

А недостатками - сложность понимания и внесения изменений другим администратором, необходимость модификаций под различных провайдеров, большой соблазн модифицировать работающее решение по мере приобретения опыта написания скриптов почти всегда приводящее к проблемам и изредка к улучшению. Отдельной проблемой является сложность переноса скриптов между разными версиями RouterOS и разными моделями маршрутизаторов.

Recursive next-hop resolving

К счастью в RouterOS реализован функционал рекурсивного поиска маршрута.

Мы можем использовать это для того что бы указать в качестве шлюза удаленный хост (например адрес своего VPN сервера) и проверять на доступность его, а не шлюз провайдера.

В случае проблем на стороне провайдера максимум через 20 секунд маршрут станет неактивным и мы начнем работать через резервный канал.

Для работы Recursive Next-Hop Target Scope >= Scope

/ip route

add dst-address=8.8.8.8/32 gateway=**local-gateway** scope=30

add gateway=8.8.8.8 check-gateway=ping target-scope=30

Recursive next-hop resolving

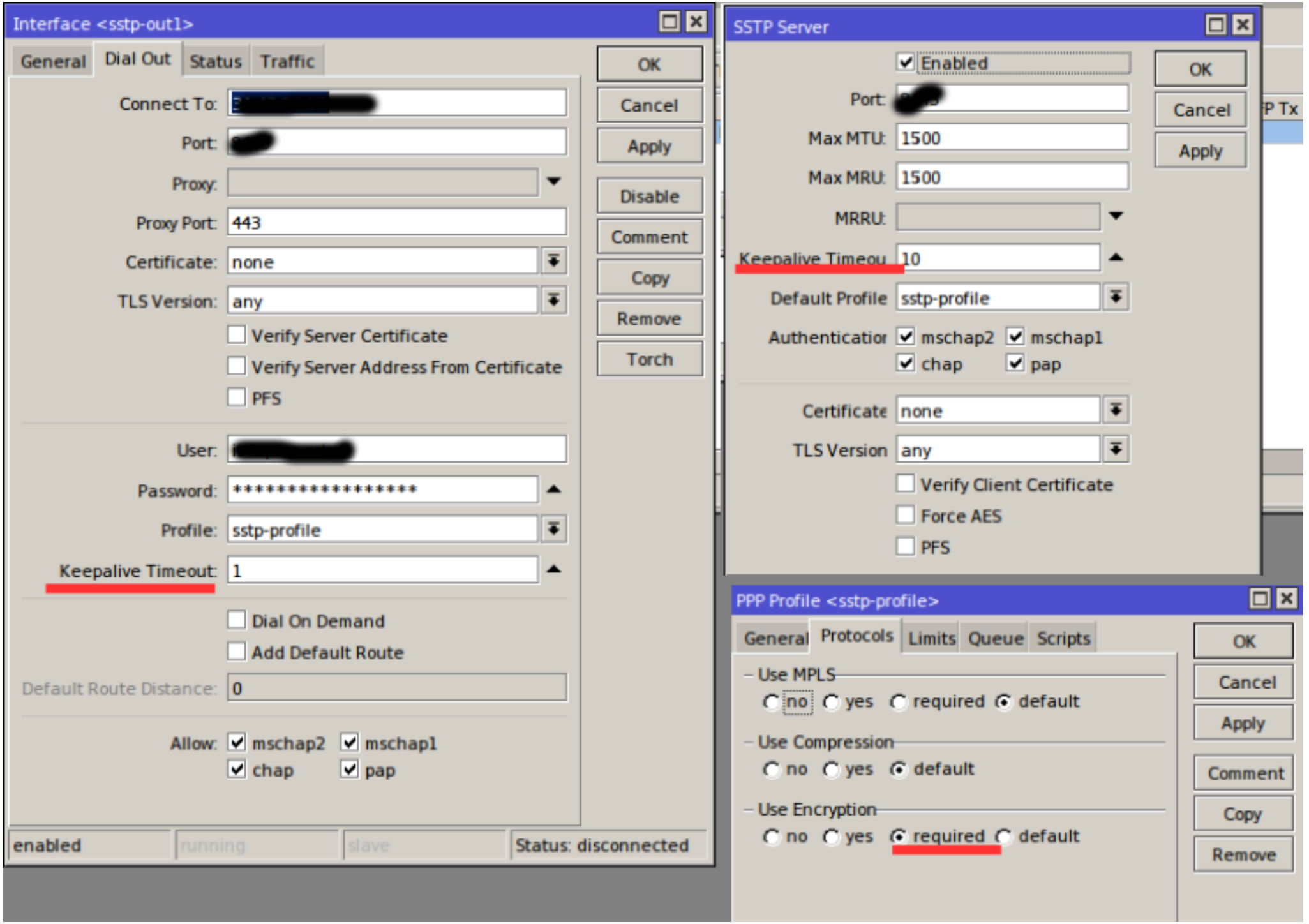
/ip route

add distance=1 dst-address=8.8.8.8/32 gateway=**IP_LocalProvider**

add check-gateway=ping comment=Main distance=1 gateway=8.8.8.8 target-scope=30

Для резервного канала установите distance=10

Настройки SSTP сервера и клиента тривиальны, главное не забыть по keepalive на обоих сторонах что бы перестраивать туннель при переключении провайдера.


Interface <sstp-out1>
General
Dial Out
Status
Traffic
OK
Cancel
Apply
Disable
Comment
Copy
Remove
Torch
Connect To:
Port:
Proxy:
Proxy Port: 443
Certificate: none
TLS Version: any
Verify Server Certificate
Verify Server Address From Certificate
PFS
User:
Password:
Profile: sstp-profile
Keepalive Timeout: 1
Dial On Demand
Add Default Route
Default Route Distance: 0
Allow: mschap2 mschap1 chap pap
enabled
running
slave
Status: disconnected
SSTP Server
Enabled
Port:
Max MTU: 1500
Max MRU: 1500
MRRU:
Keepalive Timeou 10
Default Profile sstp-profile
Authenticatior mschap2 mschap1 chap pap
Certificate none
TLS Version any
Verify Client Certificate
Force AES
PFS
PPP Profile <sstp-profile>
General
Protocols
Limits
Queue
Scripts
Use MPLS
no yes required default
Use Compression
no yes default
Use Encryption
no yes required default

Настраиваем NAT

Для построения резервного канала чаще всего использовался мобильный интернет через 3G модем. Если оставить доступ в Интернет то пользователи “забьют” канал и работать с базой данных станет неудобно. Кроме этого счета за мобильный интернет могут неприятно удивить, поэтому NAT делаем только для нужных портов.

/ip firewall nat

chain=srcnat action=masquerade out-interface=ether1-WAN-Main

chain=srcnat action=masquerade protocol=tcp out-interface=ether2-WAN-Backup dst-port=25,110,3389,33389,1494,443,5190

chain=srcnat action=masquerade protocol=udp out-interface=ether2-WAN-Backup dst-port=53,123

Человеческий фактор номер раз

При переключении на резервного провайдера приложениям в ЦО доступны а интернет нет, соответственно самые занятые работой сотрудники, а это обычно директора региональных филиалов начинают звонить с вопросами, а это отвлекает от решения проблемы.

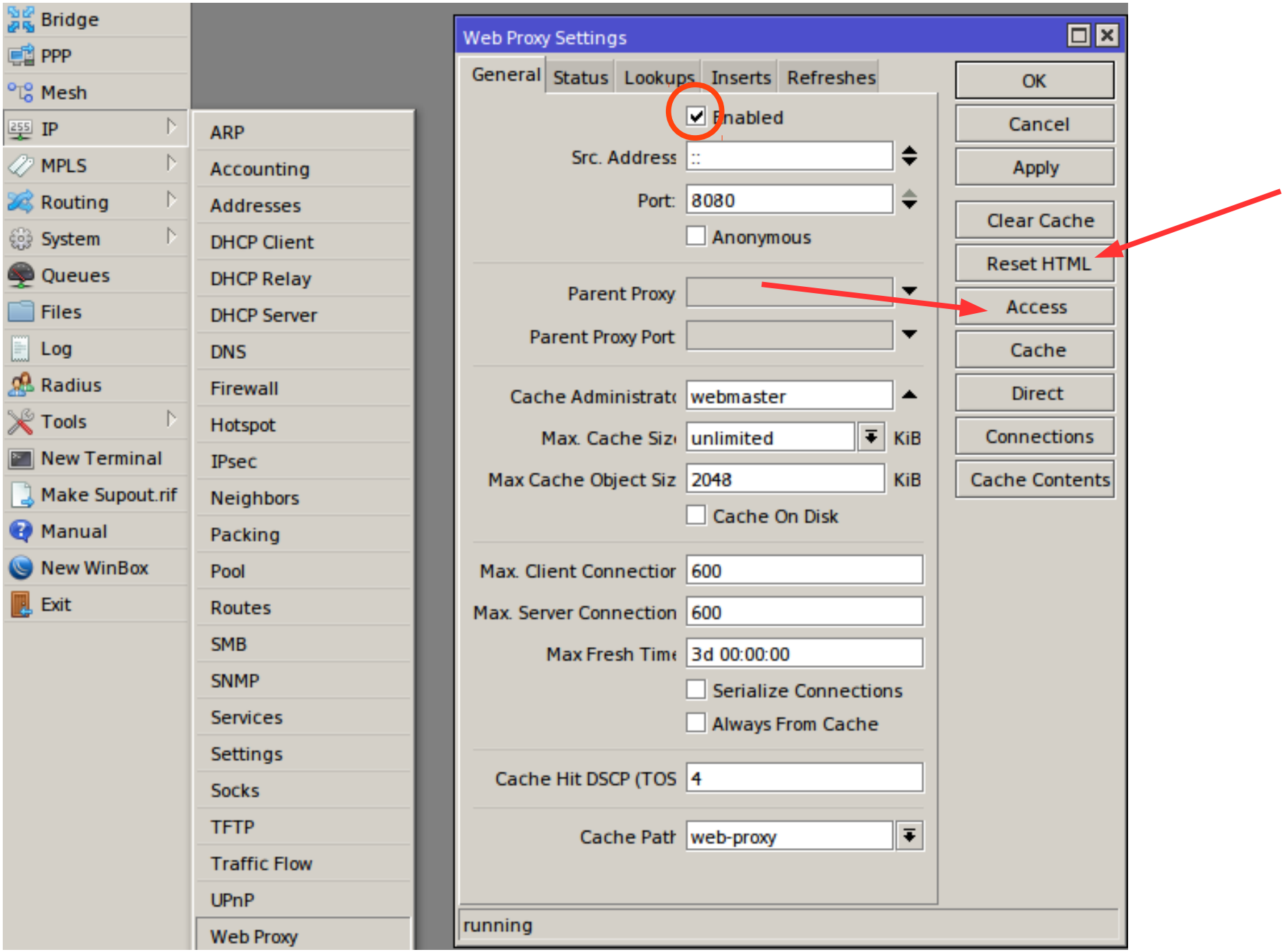
Неплохо было бы при работе на резервном канале как то сообщать пользователю об этом.

Web-proxy

В RouterOS есть реализован встроенный proxy сервер используя который можно заблокировать доступ ко всем web страницам а при попытке доступа выдавать сообщение о запрете доступа.

Нужно включить web-proxy, создать правило запрещающее доступ ко всем web страницам и кастомизировать сообщение об ошибке.

Настройка Web-proxy

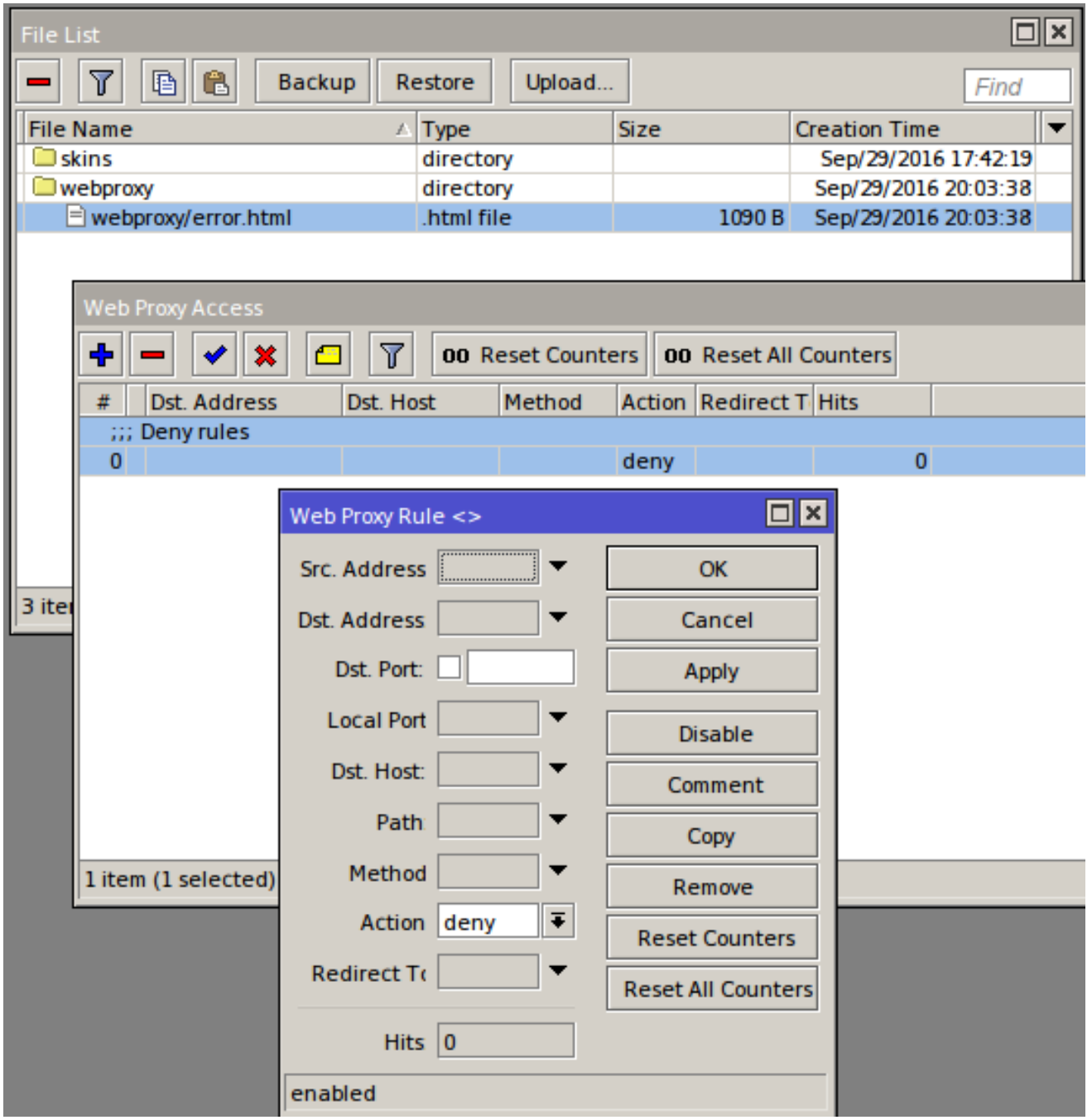


Настройка Web-proxy

Для кастомизации сообщения об ошибке нужно нажать кнопку Reset HTML и в Files появится файл webproxy/error.html – обычный html файл, его нужно подкорректировать и положить на место.

Таблица с правилами доступна по нажатию кнопки Access

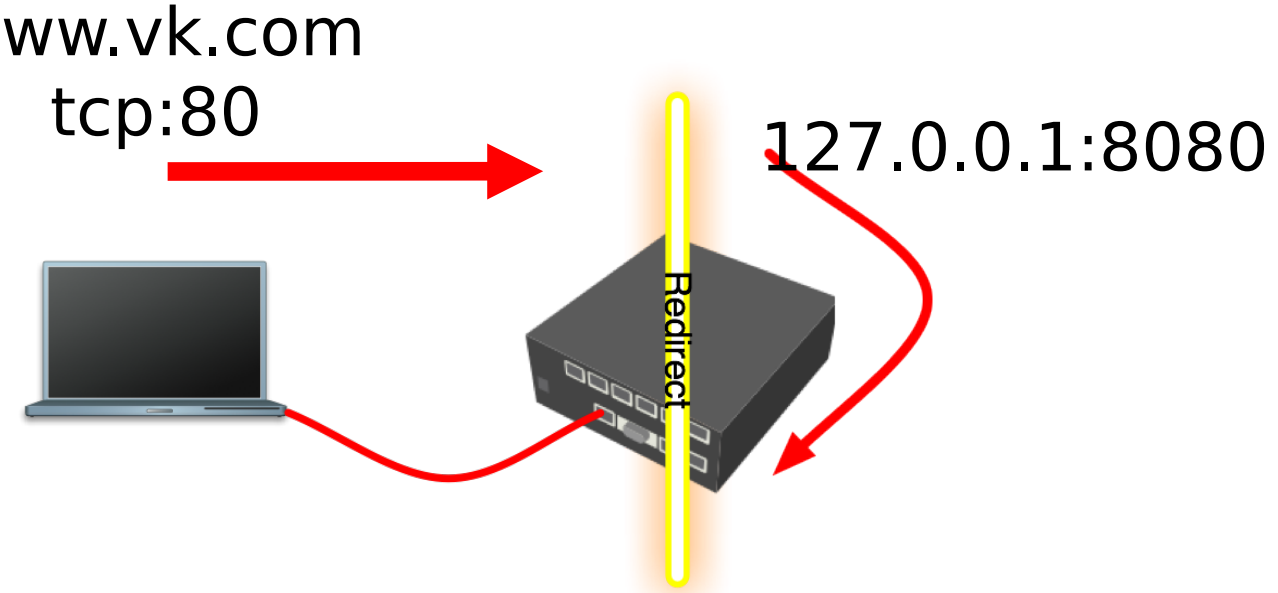
Настройка Web-proxy



Redirect

Web-proxy мы настроили но в браузере клиента нет настроенного прокси.

Нам поможет специальное правило NAT – Redirect, оно перенаправляет трафик на нужный порт нашего маршрутизатора.




```
[admin@MUM-test] > /ip firewall nat print
Flags: X - disabled, I - invalid, D - dynamic
 0 XI  ;;; Redirect
       chain=dstnat action=redirect to-ports=8080 protocol=tcp
       dst-port=80,8080,443 log=no log-prefix=""
```

Используя редирект мы превратили наш маршрутизатор в прозрачный прокси.

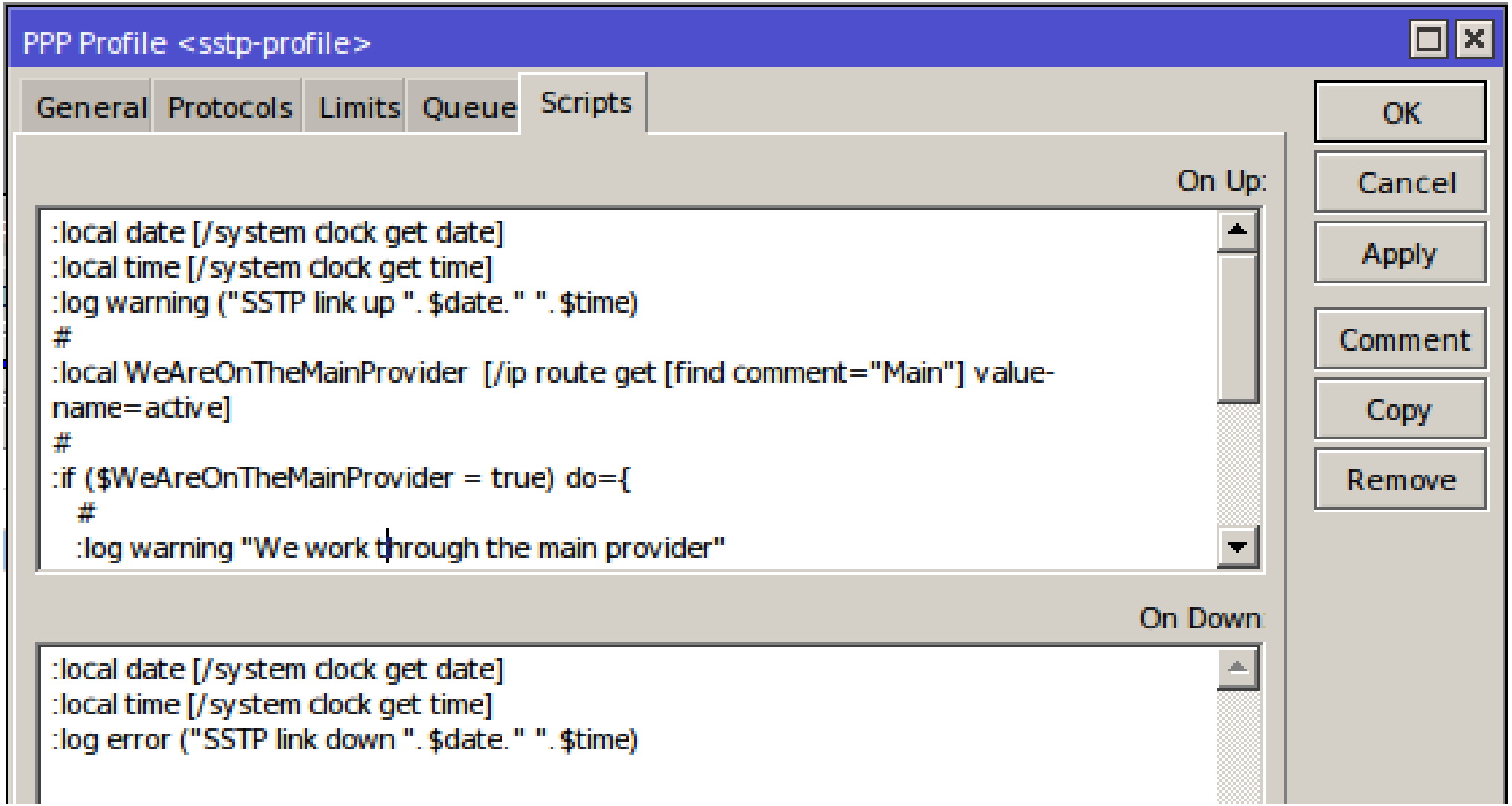
Осталось придумать как включать правило редиректа при работе на резервном канале и выключать при переходе на основной.

Можно написать скрипт определяющий переключение каналов, но это не наш метод.

Для определения факта переключения мы будем использовать то что при переходе с канала на канал sstp клиент отключается и подключается опять.

В настройках sstp клиента ничего полезного для этого нет, но давайте посмотрим в PPP Profile.

PPP Profile → Scripts



PPP Profile → Scripts

В PPP profile можно запустить скрипт при подключении и отключении клиента. Момент отключения нам особо не интересен, просто отметим его, а в момент подключения можно определить активен ли основной канал и если да то отключить редирект, а если нет то включить.

Ну и само собой все писать в логи.

Scripts on UP

```
:local date [/system clock get date]
:local time [/system clock get time]
:log warning ("SSTP link up ".$date." ".$time)


:local WeAreOnTheMainProvider  [/ip route get [find comment="Main"] value-name=active]
:if ($WeAreOnTheMainProvider = true) do={
   :log warning "We work through the main provider"
   /ip firewall nat disable [find comment ="Redirect"]
}
:if ($WeAreOnTheMainProvider = false) do={
   :log warning "We work through the backup provider"
   /ip firewall nat enable [find comment ="Redirect"]
}
```

Scripts on Down

```
:local date [/system clock get date]
:local time [/system clock get time]
:log error ("SSTP link down ".$date." ".$time)
```

Человеческий фактор номер два. Один за всех.

По мере повышения надежности основного канала переключения на резервный канал стали происходить не каждый месяц и региональные директора повадились забирать USB модемы домой справедливо рассудив что им нужнее. Соответственно при отказе канала резервный строить не получалось.

Т.к. филиал далеко то уговоры не сильно помогали.

А так как самый эффективный способ воспитания сознательности это воздействие через коллектив то будем использовать именно его.

Все за одного.

Сделаем так что бы при отсутствии USB модема в порту роутер два раза в минуту мерзко пищал.

Посмотреть что подключено по USB можно так:

```
[admin@Home] > /system resource usb print value-list
          device: 1:1                        1:9
          vendor: Linux 3.3.5 ehci_hcd       HUAWEI Technology
            name: RB400 EHCI                 HUAWEI Mobile
   serial-number: rb400_usb
       vendor-id: 0x1d6b                     0x12d1
       device-id: 0x0002                     0x1001
           speed: 480 Mbps                   480 Mbps
           ports: 1                          0
     usb-version: 2.00                       2.00
```

Скрипт проверки наличия USB модема

При отсутствии модема в порту пусть роутер пищит 10 раз меняя частоту.

```
:if ([:len [/system resource usb find vendor-id=0x12d1]] = 0) do={
:log error "No modem";
:for j from=1 to=10 step=1 do={
:beep frequency=((11-$j)*300) length=150ms;
:delay 500ms;
}
}
```

И делает это два раза в минуту:

```
/system scheduler
add disabled=no interval=30s name=schedule-check-modem on-event="/system script run 0" policy=read
```

Имитация отказа основного канала

Для тестирования полезно иметь возможность имитации отказа основного канала, при этом должен происходить переход на резервный и можно проверить правильно ли отрабатывают скрипты.

Т.к. check-gateway пингует шлюз можно просто запретить пинги к шлюзу в цепочке output и включать/выключать это правило.

```
[admin@MUM-test] > /ip firewall filter print
Flags: X - disabled, I - invalid, D - dynamic
 0    ;;; Disable main
      chain=output action=drop protocol=icmp dst-address=8.8.8.8
```

Дополнительные материалы:

http://wiki.mikrotik.com/wiki/Advanced_Routing_Failover_without_Scripting

http://wiki.mikrotik.com/wiki/Manual:Scripting

http://wiki.mikrotik.com/wiki/Manual:Scripting-examples

Спасибо за внимание.

Готов ответить на вопросы.