Автоматизированный обход блокировок Роскомнадзора на Mikrotik

Как известно, Роскомнадзор вносит досадные коррективы в наши жизни, что в значительной степени тратит нервы и снижает удобство использования интернета. В этой статье пойдет речь об автоматизированном обходе блокировок Роскомнадзора с помощью роутера Mikrotik. Я, как человек, который любит технику индустриального масштаба (а Mikrotik именно к такой и относится!) просто не мог пройти мимо того, как можно использовать мощности такого замечательного промышленного роутера. В моем случае это мой домашний Mikrotik CRS125, но эта инструкция применима и для другого подобного железа, главное, чтобы объем памяти был достаточен для загрузки файла с выгрузкой заблокированных IP РКН, на совсем уж младших моделях система может работать медленно во время импорта. Например, на hAP, с его памятью в 32M, будет тяжеловато, и, вполне возможно, что нормально не заработает (увы, сейчас нет hAP под рукой, протестировать не могу).

Схема [1)] работы выглядит так - на сервере генерируется файл с заблокированными адресами Роскомнадзором (изначально парсится отсюда) в понятном для Mikrotik виде, затем все это импортируется в Mikrotik, и он уже сортирует трафик - IP, которые не заблокированы (следовательно, в списке не находятся) отправляются напрямую, а если IP находится в списке - он идет через VPN. В итоге имеем замечательную схему работы - список обновляется автоматически, в итоге - Роскомнадзор может чудить, что хочет - мы этого не заметим, так как умный роутер свое дело сделает, и все будет открываться и работать так, как это было задумано.

Необходимые ингредиенты

- VPS (желательно)
- VPN (обязательно, можно использовать поднятый на VPS)
- Mikrotik
- Прямые руки
- Бутылка шотландского эля (опционально)

Подготовка сервера

Для того, чтобы выгружать базу заблокированных IP нам нужно использовать свой VPS, я пользуюсь Digital Ocean, но может подойти также и Hetzner (или другие). Брать можно самый дешевый, но если на нем же будет поднят еще и VPN - стоит обратить внимание на то, чтобы трафик был не лимитирован. В качестве операционной системы я использую Debian/Ubuntu, лично мне так удобнее.

Создаем на сервере несколько файлов по пути `/usr/local/bin/update-rkn` со следующим содержимым:

update-rkn

```
#!/bin/bash -e
set -o pipefail
curl -sS
'https://raw.githubusercontent.com/zapret-info/z-i/master/dump.csv' |
iconv -f cp1251 \
| sed -re 's/^([^;]*);.*$/\1/' -e 's/ \| /\n/g' | sort -u \
| { echo '/ip firewall address-list';
    echo 'remove [/ip firewall address-list find list=rkn]';
    sed -rne 's/^([0-9]+\.){3}[0-9]+(\/[0-9]+)?$/add list=rkn
address=&/p';
    echo '/log info' $(date -d "+3 hours" +%D/%T); } \
>/tmp/rkn.rsc
mv /tmp/rkn.rsc rkn.rsc
```

И делаем его исполняемым:

```
sudo chmod +x /usr/local/bin/update-rkn
```

Этот скрипт формирует в понятном для Mikrotik виде выгрузку IP, теперь нам надо сделать так, чтобы он исполнялся через определенное время. Я для себя решил это делать на сервере раз в час, потому что у меня используется не один роутер, и загружают они данные в разное время.

Создаем файл по пути `/etc/systemd/system/update-rkn.service`.

update-rkn.service

```
[Unit]
Description=Update RKN list
Wants=network-online.target.wants

[Service]
Type=oneshot
ExecStart=/usr/local/bin/update-rkn
User=root
WorkingDirectory=/var/www/mr-allen.com/public_html
```

Создаем файл таймера (переодичность запуска скрипта и генерации выгрузки IP) `/etc/systemd/system/update-rkn.timer`.

update-rkn.timer

```
[Unit]
Description=Update RKN list hourly

[Timer]
OnCalendar=hourly
Persistent=true
```

```
[Install]
WantedBy=timers.target
```

И выдаем файлам нужные права:

```
sudo chmod 755 /etc/systemd/system/update-rkn.service
sudo chmod 755 /etc/systemd/system/update-rkn.timer
```

Задача этих файлов - выгружать список в каталог веб-сервера, откуда его уже будет забирать Mikrotik. Теперь можем их запустить, и понять, что файл успешно генерируется в каталоге веб-сервера.

```
systemctl start update-rkn.service
systemctl start update-rkn.timer
systemctl enable update-rkn.timer
```

Проверить, правильно ли генерируется выгрузка можно следующим образом:

```
tail /var/www/mr-allen.com/public_html/rkn.rsc
```

В ответ должен быть показан скрипт выгрузки и последней строчкой дата его генерации:

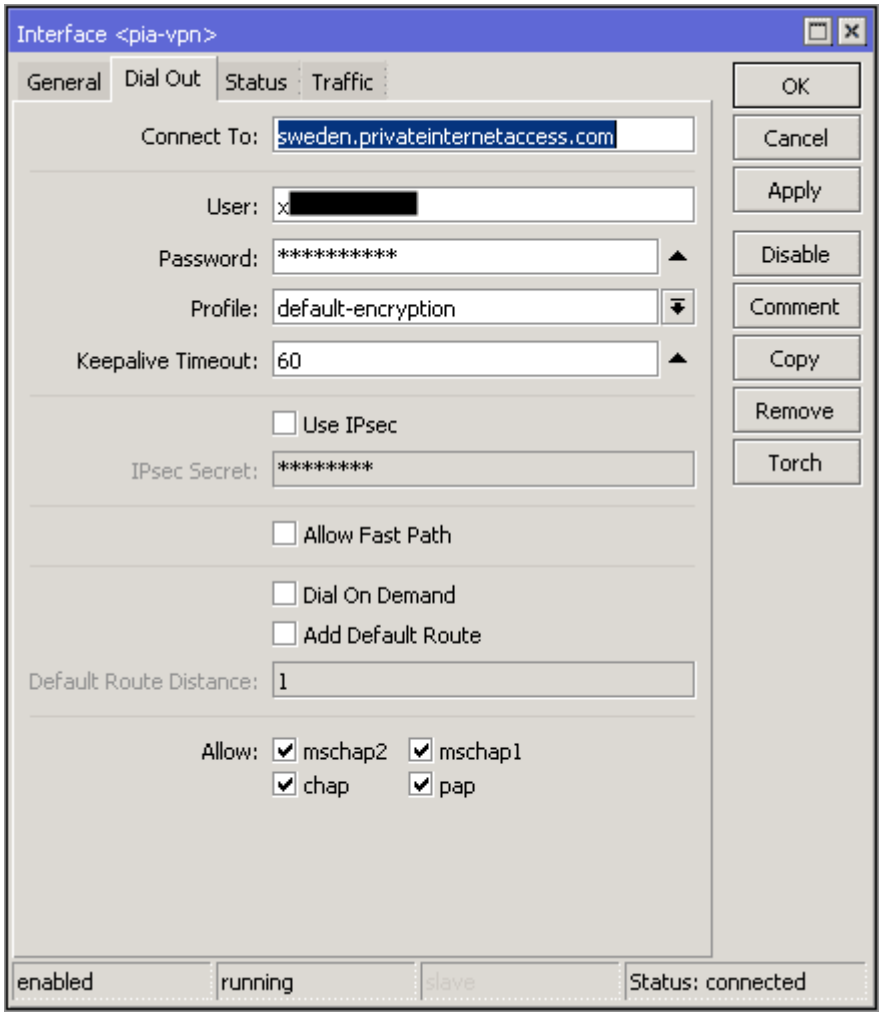
```
add list=rkn address=99.192.231.70
add list=rkn address=99.192.231.87
add list=rkn address=99.192.247.195
add list=rkn address=99.192.247.35
add list=rkn address=99.192.247.37
add list=rkn address=99.192.247.50
add list=rkn address=99.192.247.61
add list=rkn address=99.192.254.165
add list=rkn address=99.192.254.198
/log info 09/07/18/18:00:37
```

Если все получилось именно так, значит настройка сервера завершена, и можно переходить к настройке роутера.

Настройка PIA VPN на Mikrotik

Я в своем случае не стал поднимать VPN на сервере, и решил использовать VPN от Private Internet Access.

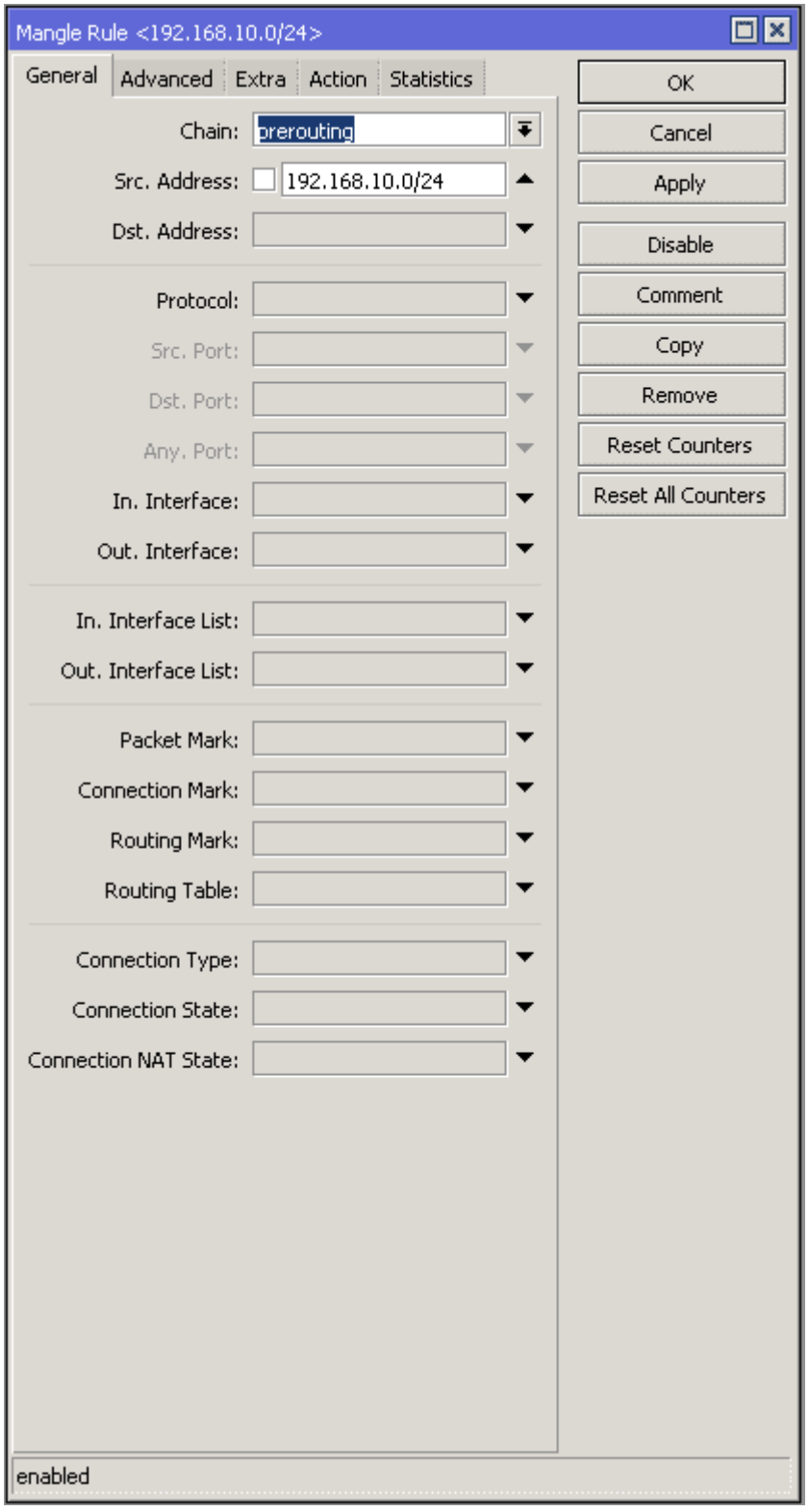
Заходим в Winbox по пути PPP ⇒ Interface ⇒ L2TP Client ⇒ Add (+) и задаем параметры, как на скриншоте:

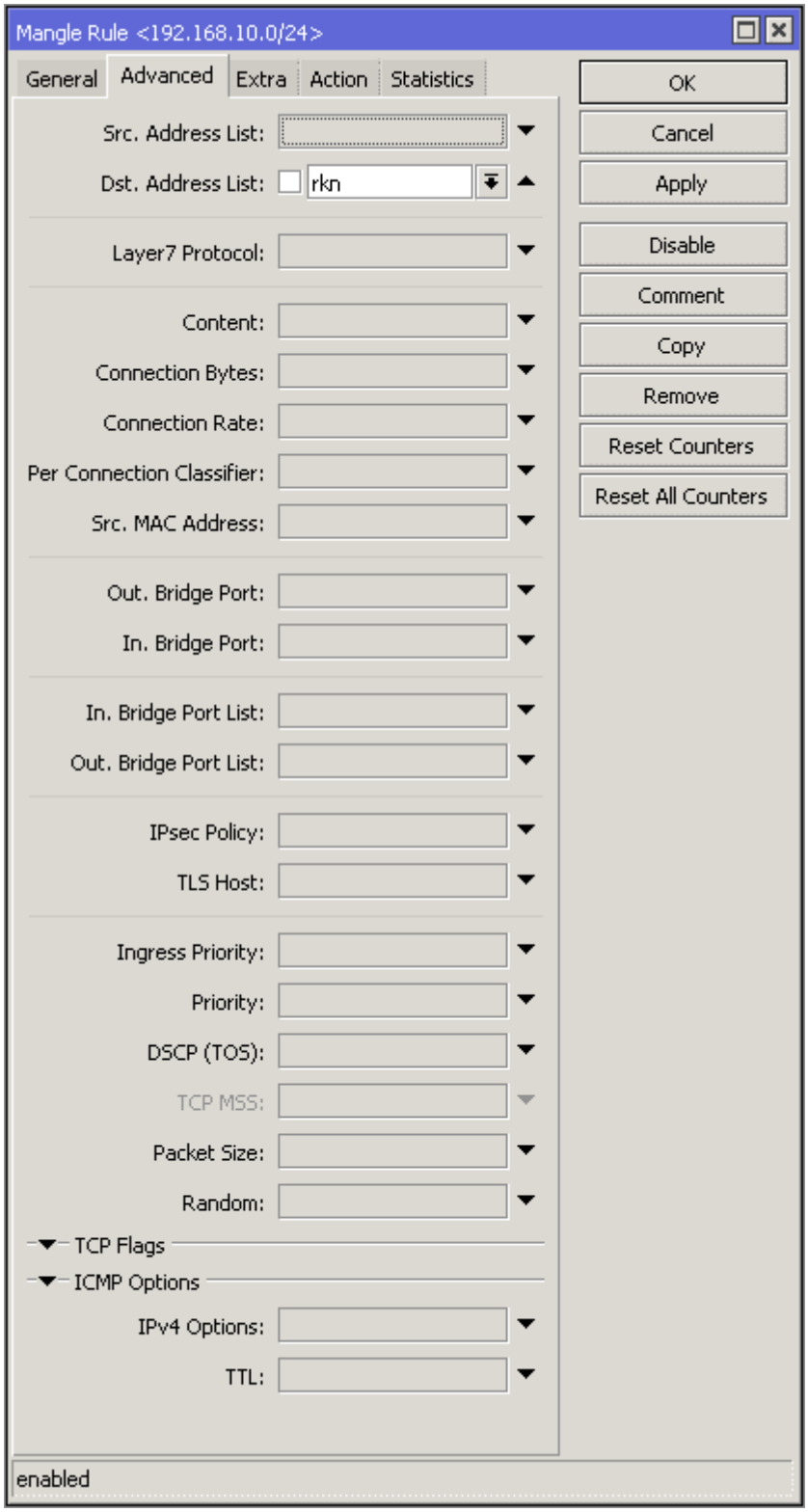


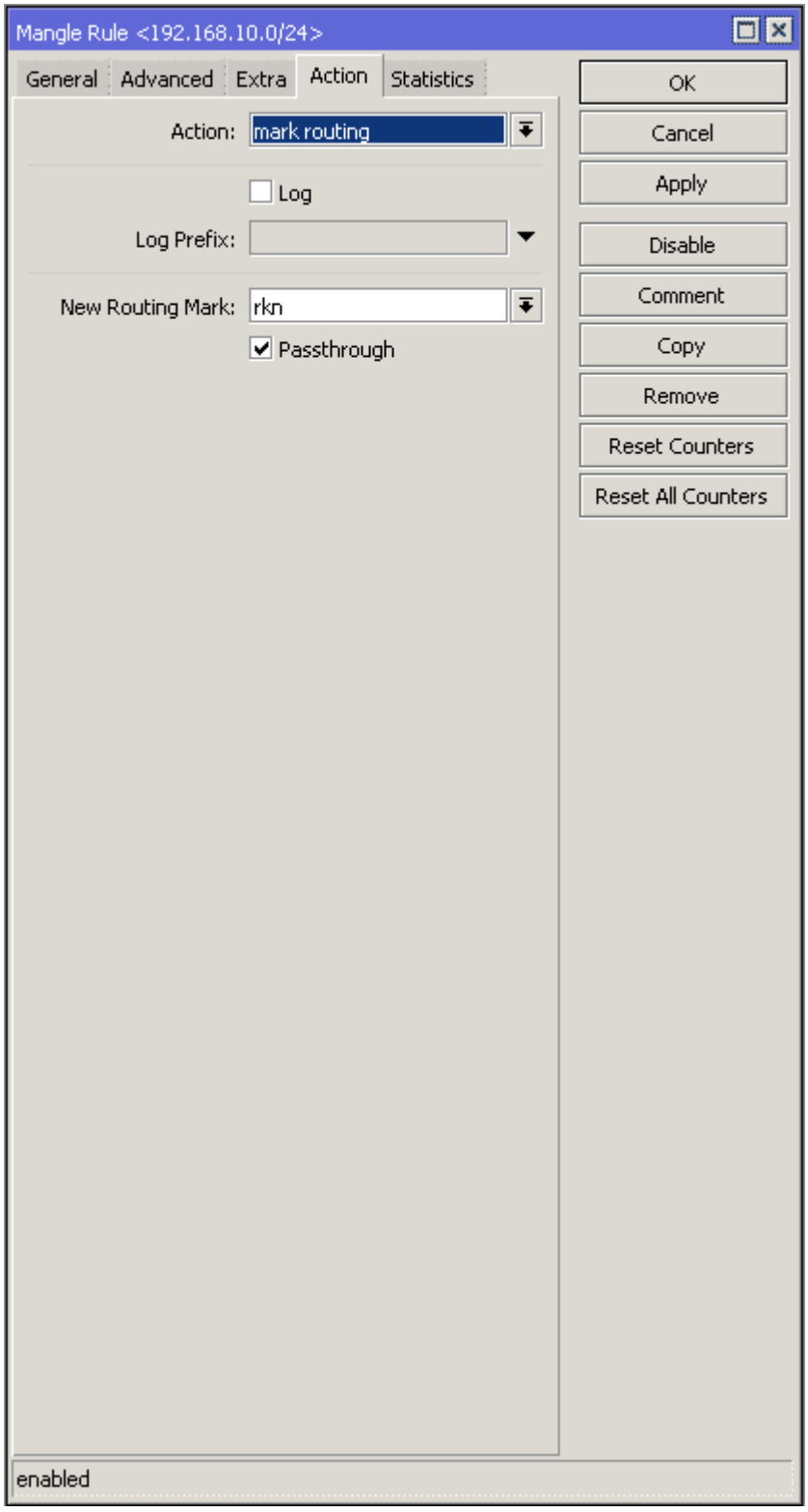
Нельзя ставить галочку `Add Default Route`, иначе весь трафик пойдет через VPN

Соединение с VPN установлено, теперь надо объяснить роутеру, что пускать через него, в этом нам поможет маркировка трафика.

Заходим по пути `IP ⇒ Firewall ⇒ Mangle ⇒ Add (+)`.

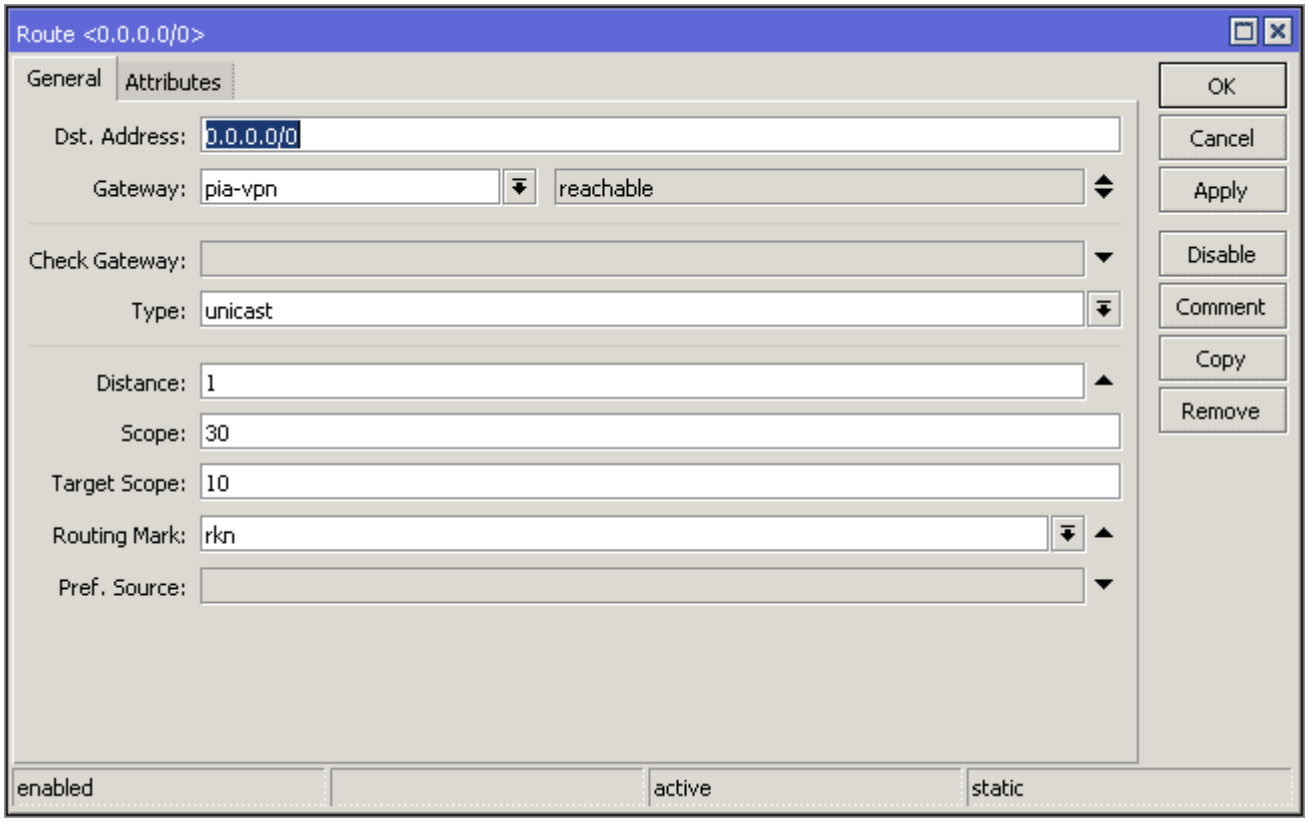

Mangle Rule <192.168.10.0/24>
General
Advanced
Extra
Action
Statistics
OK
Cancel
Apply
Disable
Comment
Copy
Remove
Reset Counters
Reset All Counters
Chain:
prerouting
Src. Address:
192.168.10.0/24
Dst. Address:
Protocol:
Src. Port:
Dst. Port:
Any. Port:
In. Interface:
Out. Interface:
In. Interface List:
Out. Interface List:
Packet Mark:
Connection Mark:
Routing Mark:
Routing Table:
Connection Type:
Connection State:
Connection NAT State:
enabled


Mangle Rule <192.168.10.0/24>
General
Advanced
Extra
Action
Statistics
OK
Cancel
Apply
Disable
Comment
Copy
Remove
Reset Counters
Reset All Counters
Src. Address List:
Dst. Address List:
rkn
Layer7 Protocol:
Content:
Connection Bytes:
Connection Rate:
Per Connection Classifier:
Src. MAC Address:
Out. Bridge Port:
In. Bridge Port:
In. Bridge Port List:
Out. Bridge Port List:
IPsec Policy:
TLS Host:
Ingress Priority:
Priority:
DSCP (TOS):
TCP MSS:
Packet Size:
Random:
TCP Flags
ICMP Options
IPv4 Options:
TTL:
enabled



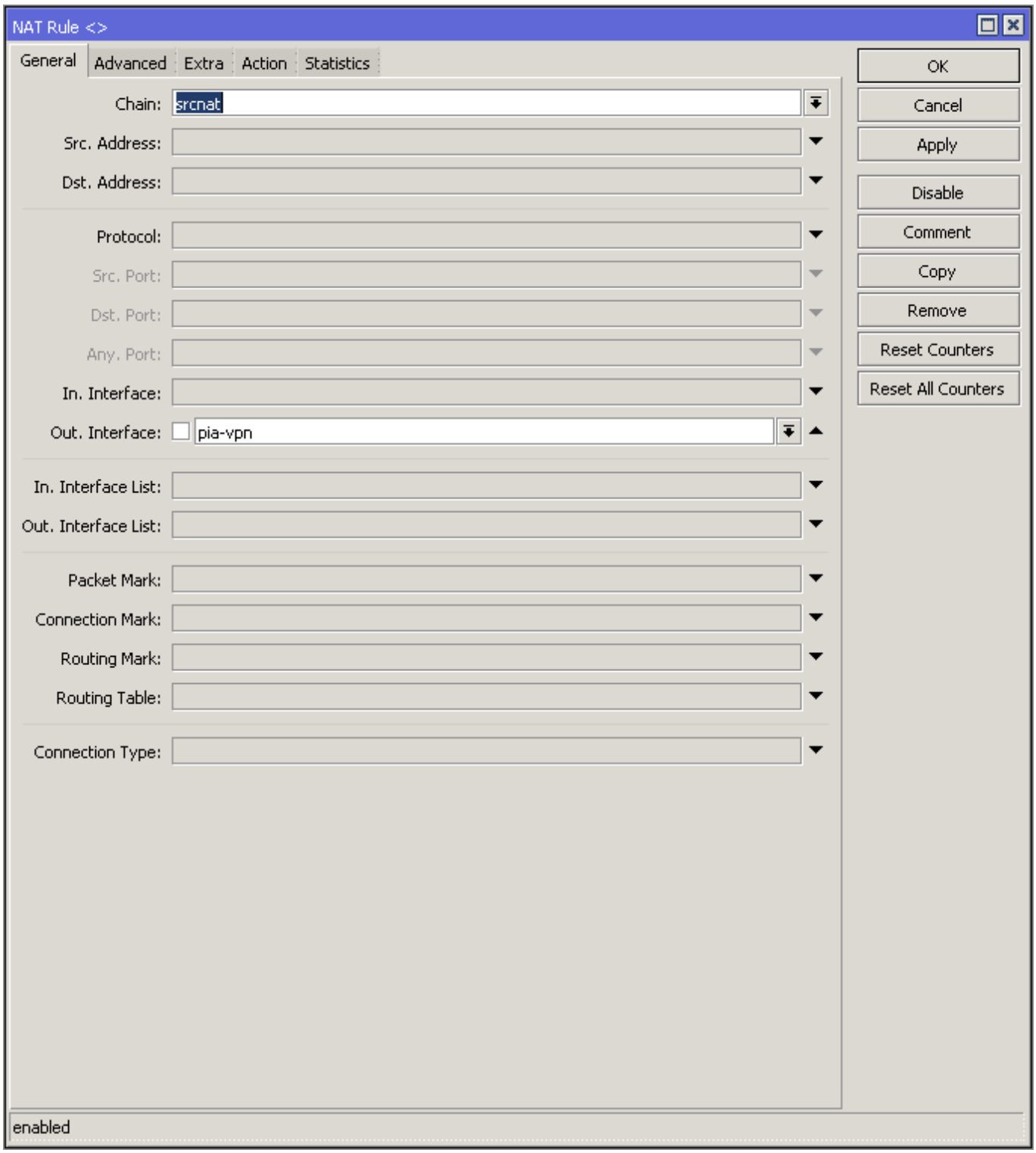
Src. Address - это наша локальная сеть, Dst. Address List - название списка адресов, куда будут импортироваться заблокированные IP, а во вкладке `Action ⇒ Mark Routing ⇒ New Routing Mark`—имя метки. Теперь все пакеты, которые ходят по адресам из списка `rkn` помечаются меткой `rkn`. Но что же делать роутеру с этой меткой? Объясним ему это.

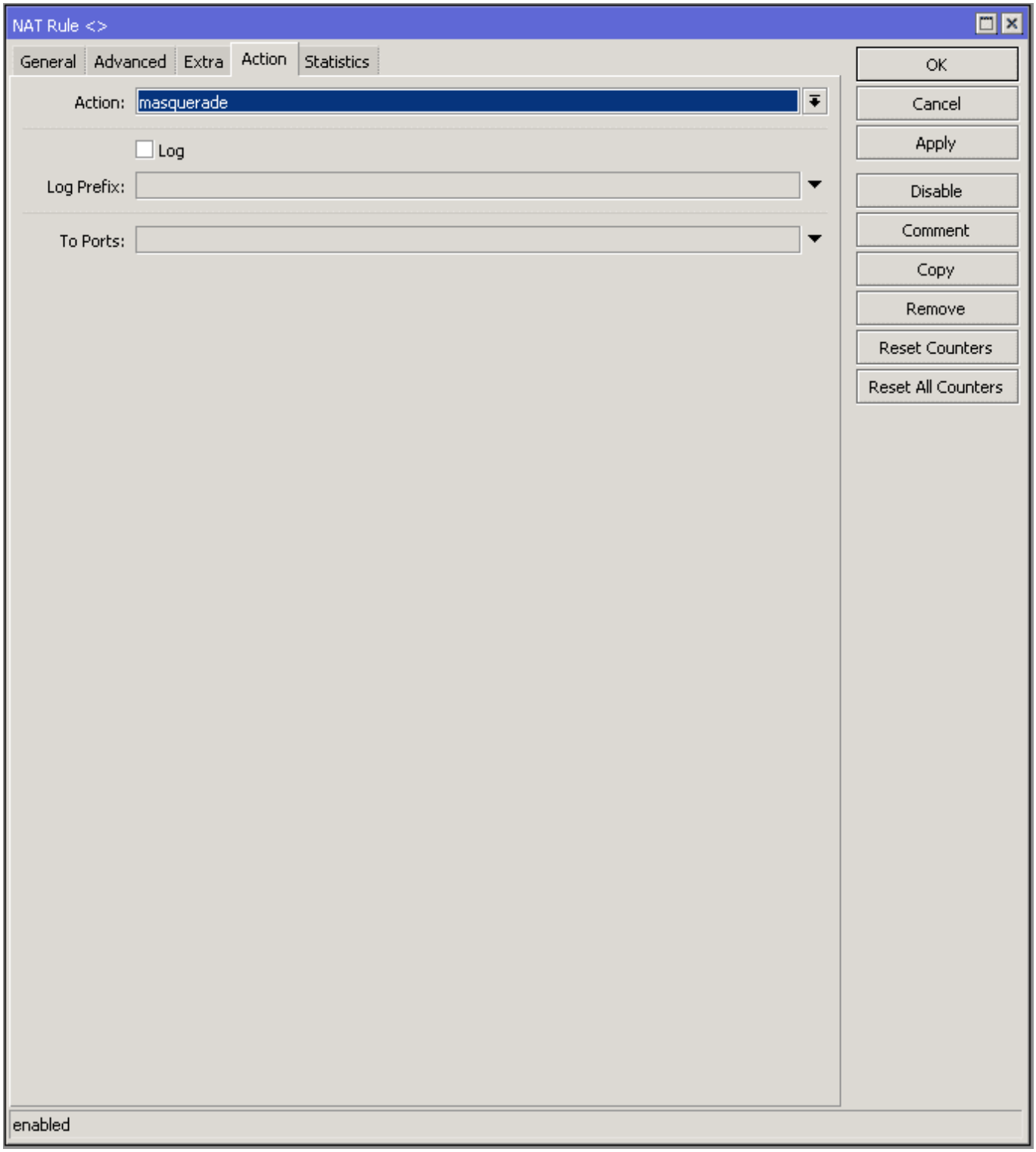
Заходим по пути `IP ⇒ Firewall ⇒ NAT ⇒ Add (+)`



Указываем все также, как и на скриншоте. Где `Gateway` - это наш VPN, а где `rkn` - наша метка.

Теперь нам нужно, чтобы роутер мог использовать VPN, для этого настроем маскарадинг - `IP ⇒ Firewall ⇒ NAT ⇒ Add (+)`.

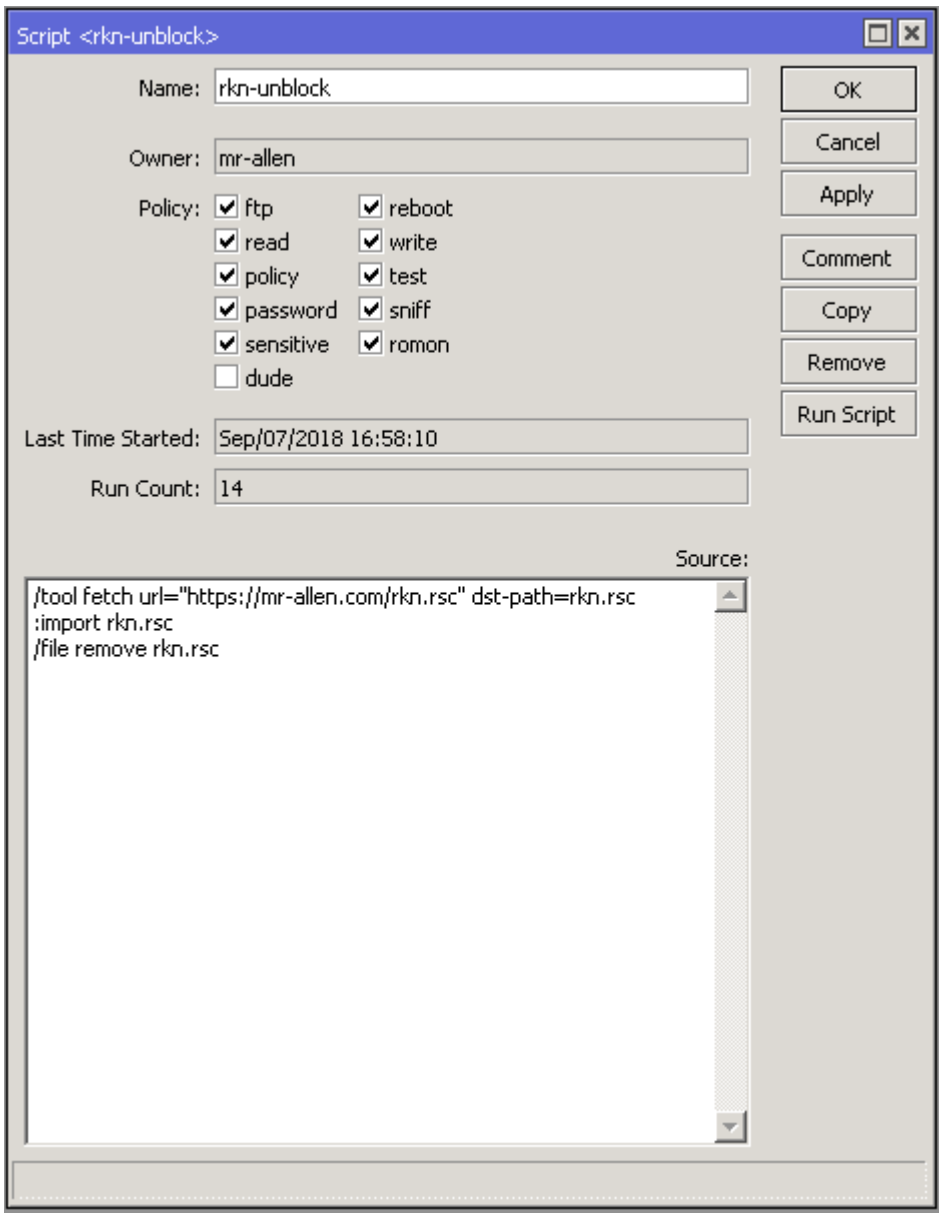

NAT Rule <>
General
Advanced
Extra
Action
Statistics
Chain:
srcnat
Src. Address:
Dst. Address:
Protocol:
Src. Port:
Dst. Port:
Any. Port:
In. Interface:
Out. Interface:
pia-vpn
In. Interface List:
Out. Interface List:
Packet Mark:
Connection Mark:
Routing Mark:
Routing Table:
Connection Type:
OK
Cancel
Apply
Disable
Comment
Copy
Remove
Reset Counters
Reset All Counters
enabled



Настройка VPN завершена, теперь нам осталось только настроить обновление списка заблокированных адресов в автоматическом режиме, для этого нам помогут `Scripts` и `Scheduler`. Сначала добавляем скрипт в `System ⇒ Scripts`.

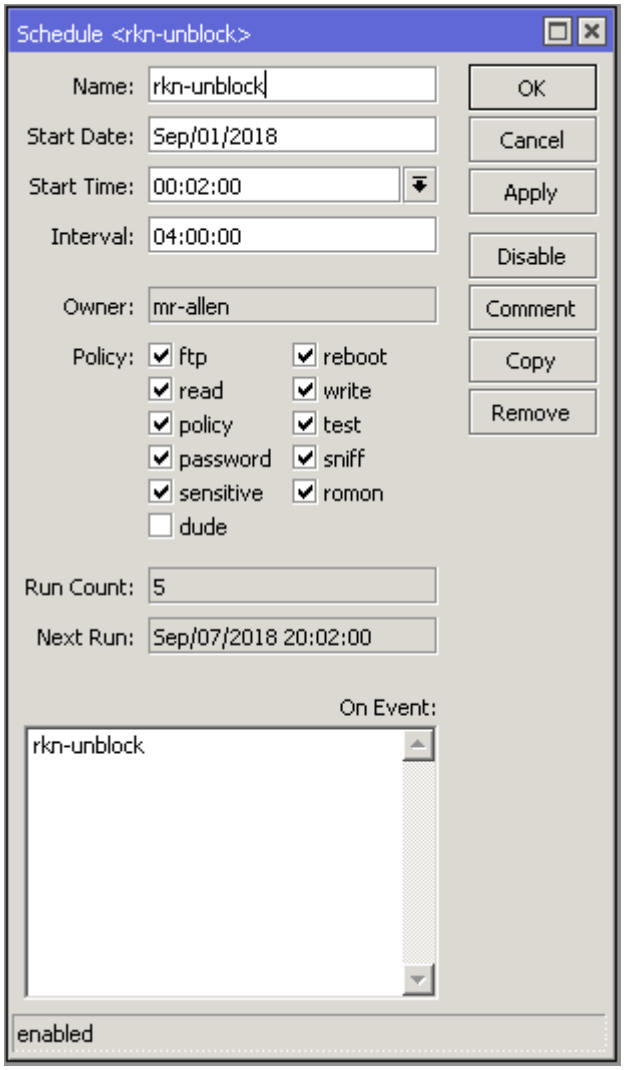
Вы можете загружать мой скрипт, который генерируется раз в час, но я крайне НЕ советую этого делать, так как загружать конфигурацию с чужого сервера - достаточно глупо.

```
/tool fetch url="https://mr-allen.com/rkn.rsc" dst-path=rkn.rsc
:import rkn.rsc
/file remove rkn.rsc
```



Ссылку необходимо заменить на вашу, которая генерируется на вашем сервере и доступна из браузера. После сохранения скрипта, его можно запустить - должны начать открываться заблокированные сайты, нам осталось только сделать так, чтобы база обновлялась сама.

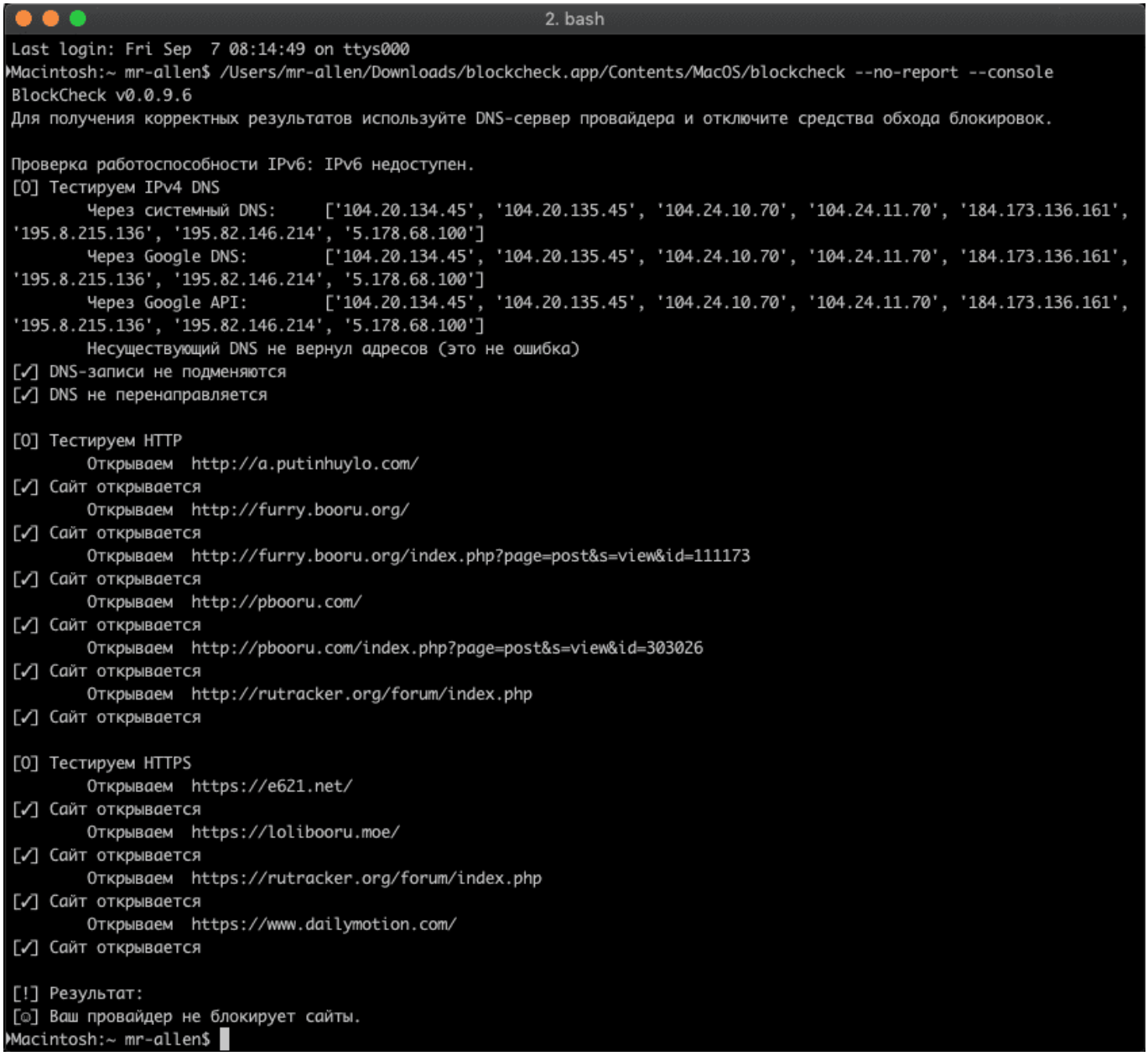
Для этого переходим в `System ⇒ Scheduler` и настраиваем автоматический запуск скрипта. У меня настроен запуск 1 раз в 4 часа.



Время старта `00:00:02` - это первый запуск в 12:02 ночи, двухминутный запас нужен для того, чтобы сервер успел сгенерировать файл (обычно на это уходит не более 40 секунд). Интервал можно выставить такой, какой вам нужен (главное, чтобы файл успевал обновиться на сервере). Не рекомендую это делать чаще, так как роутер достаточно долго прожевывает базу (около 10 минут на обновление базы заблокированных IP), и в итоге получается так, что раз в 4 часа могут быть на несколько минут недоступны заблокированные сайты. Рекомендуемый интервал - 1 день, ночью.

Проверка работы

Если вы все правильно настроили, то теперь должны открываться все ранее недоступные сайта, а также проверка с помощью BlockCheck показывать именно то, что у меня на скриншоте. Когда будете тестировать - не забудьте указать аргумент `–no-report`.



linux, vpn, vps, mikrotik, dpi, роскомнадзор

[1)]

за основу исходный код взят - Alexander Shpilkin's GIT, немного подкорректирован на мое усмотрение, огромный респект автору!