

Настройка черного и белого списков в роутерах Mikrotik

 interface31.ru/tech_it/2019/10/nastroyka-chernogo-i-belogo-spiskov-v-routerah-mikrotik.html

Записки IT специалиста

Технический блог специалистов ООО "Интерфейс"

- [Главная](#)
- Настройка черного и белого списков в роутерах Mikrotik

Ограничение доступа к тем или иным ресурсам сети интернет на основе списков достаточно популярный способ фильтрации, несмотря на его недостатки. Действительно, в большинстве случаев требуется заблокировать достаточно небольшой список ресурсов, скажем, соцсети, с чем данный метод вполне успешно справляется. В данной статье мы поговорим о том, как настроить фильтрацию по спискам на роутерах Mikrotik, тем более что RouterOS предоставляет нам для этого достаточно широкие возможности.



Онлайн-курс по MikroTik

Научиться настраивать MikroTik с нуля или систематизировать уже имеющиеся знания можно на [углубленном курсе по администрированию MikroTik](#). Автор курса, сертифицированный тренер MikroTik Дмитрий Скоромнов, лично проверяет лабораторные работы и контролирует прогресс каждого своего студента. В три раза больше информации, чем в вендорской программе MTCNA, более 20 часов практики и доступ навсегда.

Прежде всего поговорим об особенностях фильтрации в условиях современного интернета. Основной тенденцией последних лет является массовый переход на протокол HTTPS. Что это означает? В отличие от HTTP, который передает данные открытым текстом, HTTPS использует SSL-шифрование и все, что мы можем увидеть для такого соединения - это домен назначения. Какие именно страницы посещает пользователь на указанном домене и какие данные оттуда передаются мы видеть не можем.

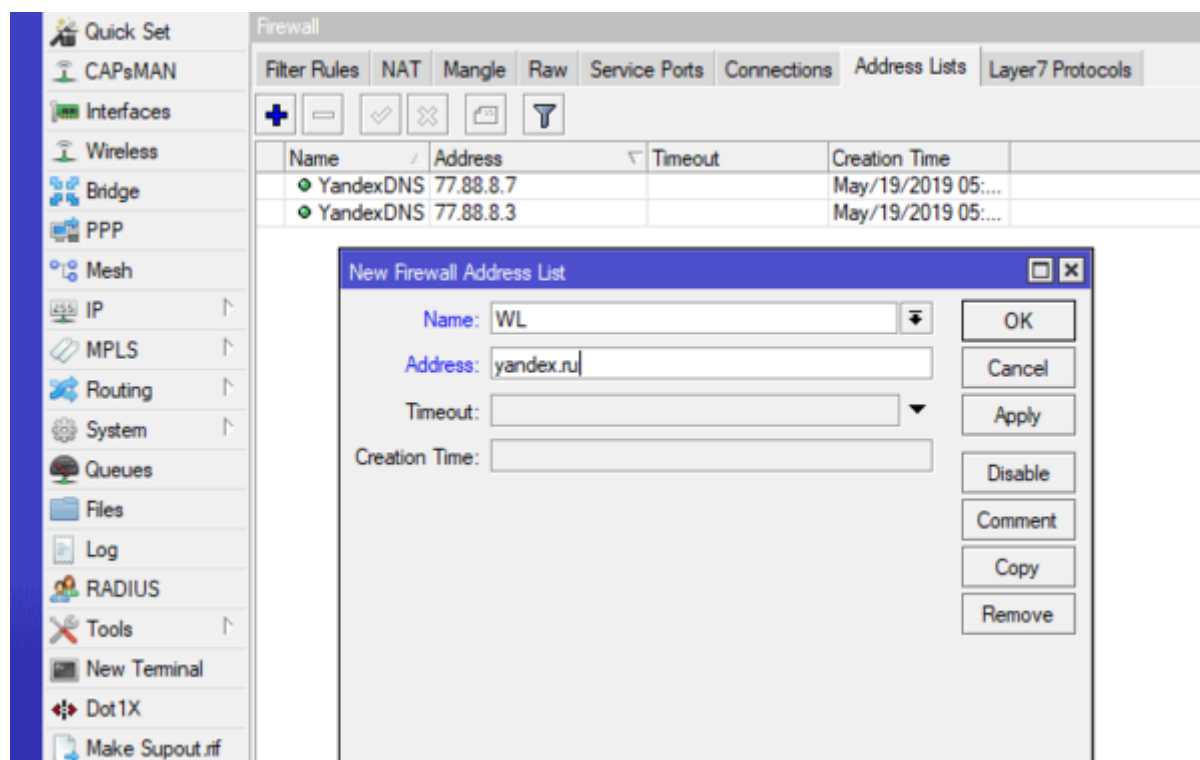
Из этого следует, что мы не можем блокировать отдельные страницы, но можем заблокировать домен целиком. Для большинства сценариев этого вполне достаточно. Но здесь нас подстерегает другая неприятность, многие сайты используют CDN (*Content Delivery Network, сеть доставки контента*), такие как CloudFlare и заблокировав нужный вам сайт вы можете также ограничить доступ к большому количеству сторонних ресурсов. Что из этого может выйти все мы видели во время ковровых блокировок РКН против Телеграм.

Также следует понимать, что блокировка посредством черных списков применима лишь к небольшому числу ресурсов, например, популярные соцсети, попытка таким образом фильтровать весь нежелательный трафик выливается в необходимость постоянной актуализации списков и может привести к высокой нагрузке на оборудование. Для таких целей лучше использовать специализированные сервисы.

Другой вариант - белые списки, при всей видимой простоте и надежности, сталкиваются с другой проблемой. Многие сайты активно используют внешние ресурсы, с которых подгружают скрипты, шрифты, стили и т.д. и т.п. и некоторые из них могут являться критичными для обеспечения полноценной работы. Также могут возникнуть проблемы с HTTPS, если браузер не сможет проверить статус SSL-сертификата. Все это потребует грамотного анализа и добавления в белый список всех тех узлов, которые необходимы для нормальной работы каждого из разрешенных сайтов.

Создаем списки

Для настройки фильтрации нам понадобятся минимум два списка: список доменов и список пользователей. С доменами понятно, это те сайты, к которым мы хотим запретить доступ или, наоборот, разрешить. Создаются такие списки просто: **IP - Firewall - Address Lists** где добавляем новый адрес, в поле **Name** вписываем имя листа, если это первая запись, либо выбираем его из выпадающего списка. В поле **Address** указываем IP-адрес или доменное имя ресурса, при указании доменного имени в список будут внесены все IP-адреса сайта, и они будут обновляться с периодичностью указанной в TTL домена.

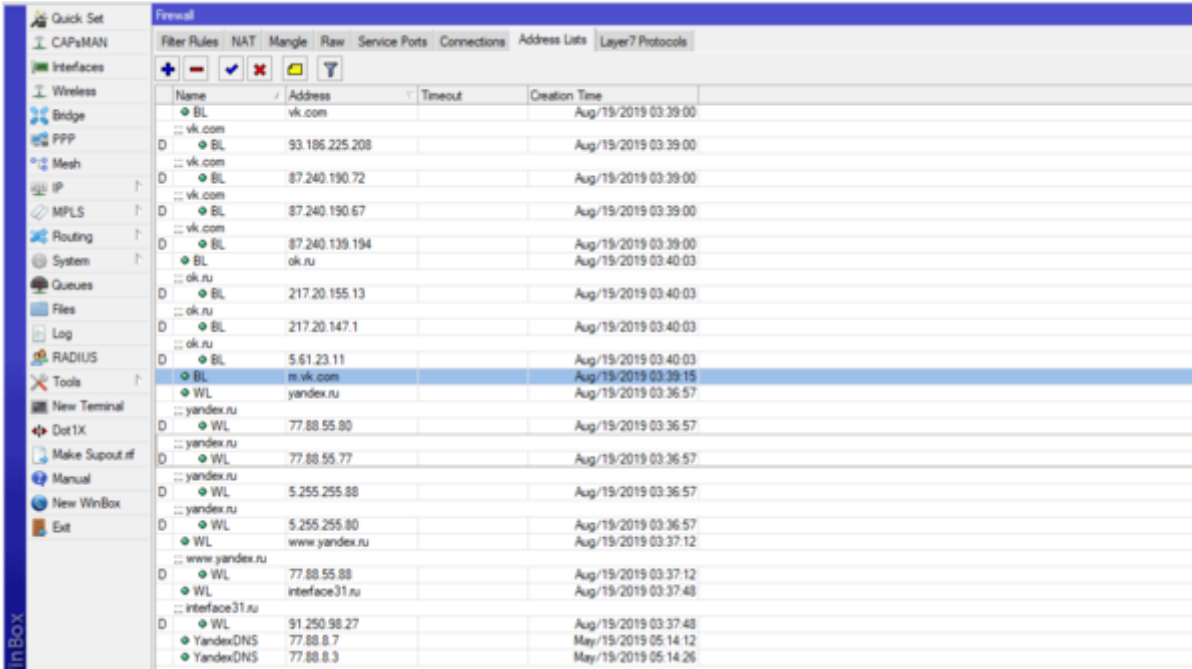


В нашем случае мы добавили домен **yandex.ru** в список **WL** (whitelist, белый список). Обратите внимание, что адреса **www.example.com** и **example.com** - это разные доменные имена, которые могут иметь разные IP-адреса (в целях балансировки нагрузки) и поэтому следует добавлять оба варианта (или проверять что между ними нет расхождений).

В командной строке это же действие можно выполнить так:

```
/ip firewall address-list  
add address=yandex.ru list=WL
```

Таких списков мы можем создать сколько нам нужно, причем один и тот же адрес может входить сразу в несколько списков. Это удобно, если нужно обеспечить для разных групп пользователей доступ к разному набору сайтов. В итоге у вас должно получиться примерно следующее:



Name	Address	Timeout	Creation Time
BL	vk.com		Aug/19/2019 03:39:00
vk.com	93.186.225.208		Aug/19/2019 03:39:00
vk.com	87.240.190.72		Aug/19/2019 03:39:00
vk.com	87.240.190.67		Aug/19/2019 03:39:00
vk.com	87.240.139.194		Aug/19/2019 03:39:00
ok.ru	217.20.155.13		Aug/19/2019 03:40:03
ok.ru	217.20.147.1		Aug/19/2019 03:40:03
ok.ru	5.61.23.11		Aug/19/2019 03:40:03
BL	m.vk.com		Aug/19/2019 03:39:15
WL	yandex.ru		Aug/19/2019 03:36:57
yandex.ru	77.88.55.80		Aug/19/2019 03:36:57
WL	77.88.55.77		Aug/19/2019 03:36:57
yandex.ru	5.255.255.88		Aug/19/2019 03:36:57
yandex.ru	5.255.255.80		Aug/19/2019 03:36:57
WL	www.yandex.ru		Aug/19/2019 03:37:12
www.yandex.ru	77.88.55.88		Aug/19/2019 03:37:12
WL	interface31.ru		Aug/19/2019 03:37:48
interface31.ru	91.250.98.27		Aug/19/2019 03:37:48
YandexDNS	77.88.8.7		May/19/2019 05:14:12
YandexDNS	77.88.8.3		May/19/2019 05:14:26

В данном примере реализовано два списка: WL - белый список и BL - черный список. Обычно в реальной жизни используется что-то одно, в нашем случае создание данных списков обусловлено сугубо учебными целями.

С доменами разобрались, остались пользователи. Существуют две политики применения правил: разрешено всем, кроме группы пользователей и запрещено всем, кроме группы пользователей. В любом случае у нас имеется группа пользователей, которая либо подвергается ограничениям, либо выводится из-под их действия. В грамотно спроектированной системе такая группа должна являться **меньшинством**, что обеспечит минимальную нагрузку на сетевое оборудование.

Также вспомним, что в руках у нас роутер, т.е. устройство, работающее на сетевом уровне (L3), а значит основные параметры, с которыми он может работать - это адрес источника и адрес назначения. Адрес назначения - это домен, выше мы его

уже разобрали. Адрес источника - это как раз пользователь, точнее - сетевое устройство пользователя. В самом простом случае мы можем создать еще один список и добавить туда IP-адреса нужных устройств.

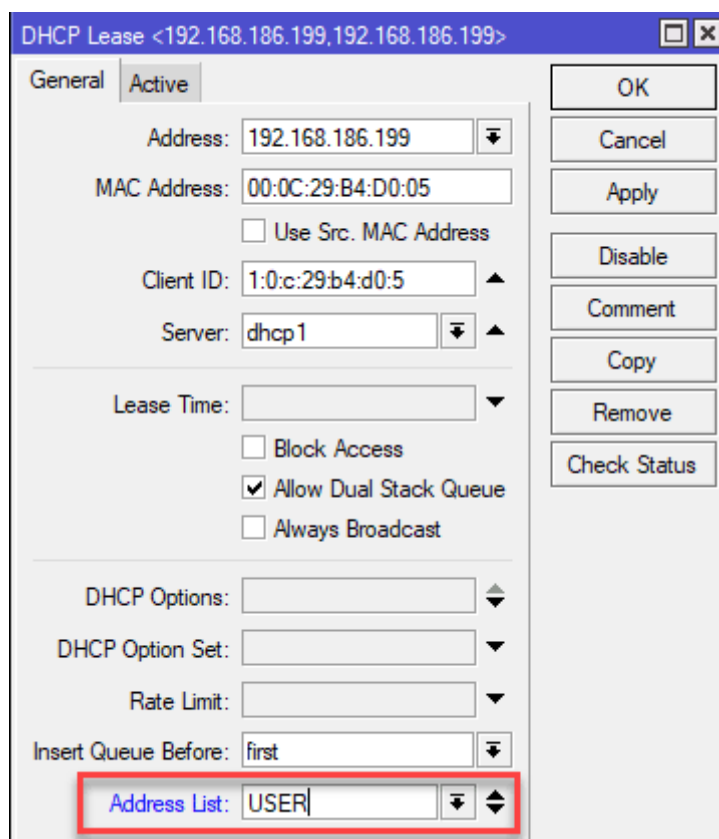
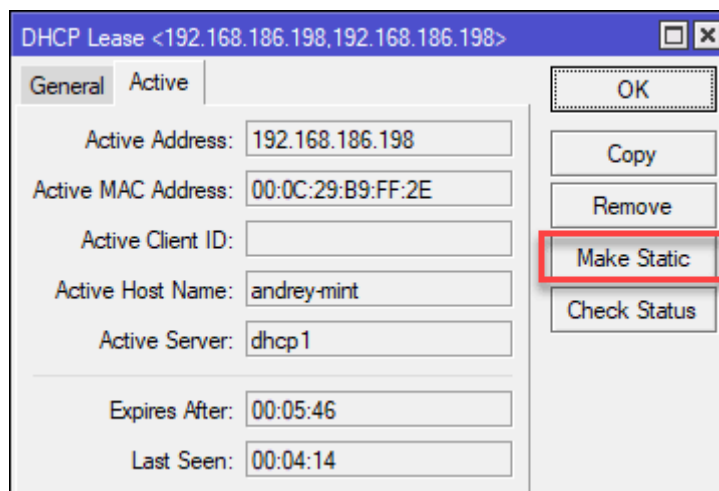
Но на практике адреса раздаются сервером DHCP, это не проблема, создаем резервирование IP-адреса, для чего следует перейти в **IP - DHCP-Server - Leases** и открыв запись нужного адреса нажать **Make Static**.

После чего закрываем и снова открываем запись и в поле **Address List** вводим, если это первая запись, или выбираем имя списка, куда будет добавлен IP-адрес данного компьютера, в нашем случае это список **USER**.

Либо через командную строку:

```
/ip dhcp-server lease  
add address=192.168.186.199  
address-lists=USER mac-  
address=00:0C:29:B4:D0:05  
server=dhcp1
```

Таким образом мы получаем список пользователей, либо несколько списков, в которых указанные адреса будут находиться до тех пор, пока на сервере активно резервирование.



Черный список

Начнем с самого простого сценария - черного списка. Сначала настроим вариант, когда такой список применяется ко всем пользователям, кроме членов списка **USER**. Для этого перейдем в **IP - Firewall - Filter Rules** и создадим новое правило.

На закладке **General** укажем **Chain - forward** и **In. Interface - bridge1**:

The screenshot shows the 'New Firewall Rule' dialog box with the 'General' tab selected. The 'Chain' dropdown is set to 'forward'. The 'In. Interface' dropdown is set to 'bridge1'. The 'Out. Interface' is empty. The 'Src. Address', 'Dst. Address', 'Protocol', 'Src. Port', 'Dst. Port', and 'Any. Port' fields are empty. On the right side, there are buttons for 'OK', 'Cancel', 'Apply', 'Disable', 'Comment', 'Copy', 'Remove', 'Reset Counters', and 'Reset All Counters'.

На закладке **Advanced** указываем **Src. Address List - USER** и ставим перед ним **восклицательный знак** (символ инверсии правила), что будет означать кроме входящих в группу. В поле **Dst. Address List** указываем **BL** - т.е. наш черный список доменов.

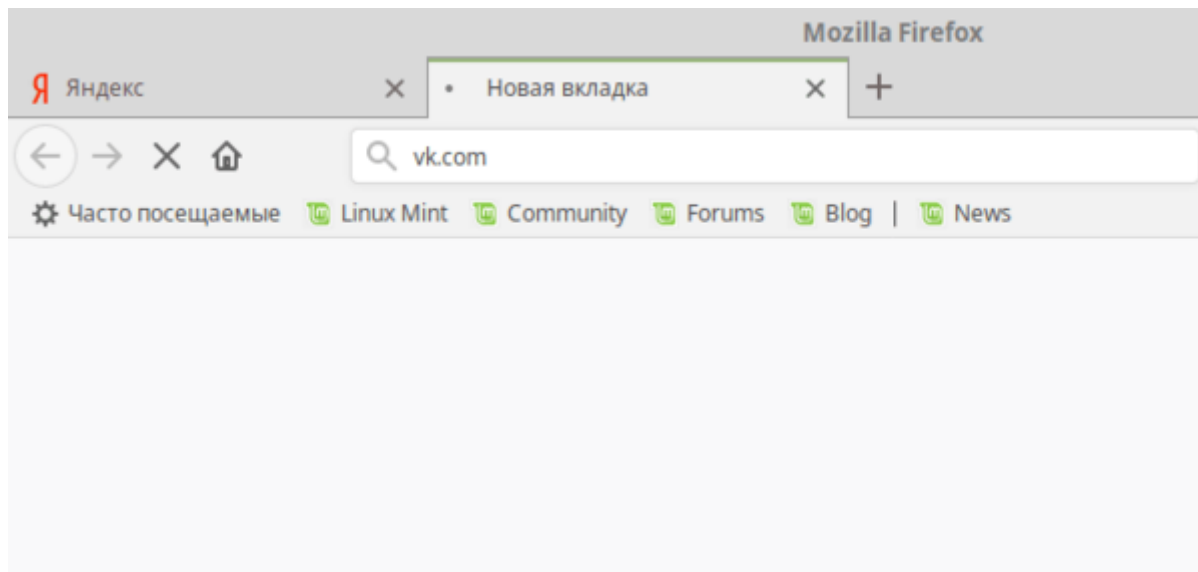
The screenshot shows the 'New Firewall Rule' dialog box with the 'Advanced' tab selected. The 'Src. Address List' dropdown is set to 'USER' with an exclamation mark icon to its left. The 'Dst. Address List' dropdown is set to 'BL'. The 'Layer7 Protocol', 'Content', 'Connection Bytes', 'Connection Rate', 'Per Connection Classifier', and 'Src. MAC Address' fields are empty. On the right side, there are buttons for 'OK', 'Cancel', 'Apply', 'Disable', 'Comment', 'Copy', 'Remove', 'Reset Counters', and 'Reset All Counters'.

И наконец на закладке **Action** указываем действие, обычно везде в интернете указывают **drop**, хорошо, укажем и мы.

The screenshot shows the 'New Firewall Rule' dialog box with the 'Action' tab selected. The 'Action' dropdown is set to 'drop'. The 'Log' checkbox is unchecked. The 'Log Prefix' field is empty. On the right side, there are buttons for 'OK', 'Cancel', 'Apply', 'Disable', 'Comment', 'Copy', 'Remove', 'Reset Counters', and 'Reset All Counters'.

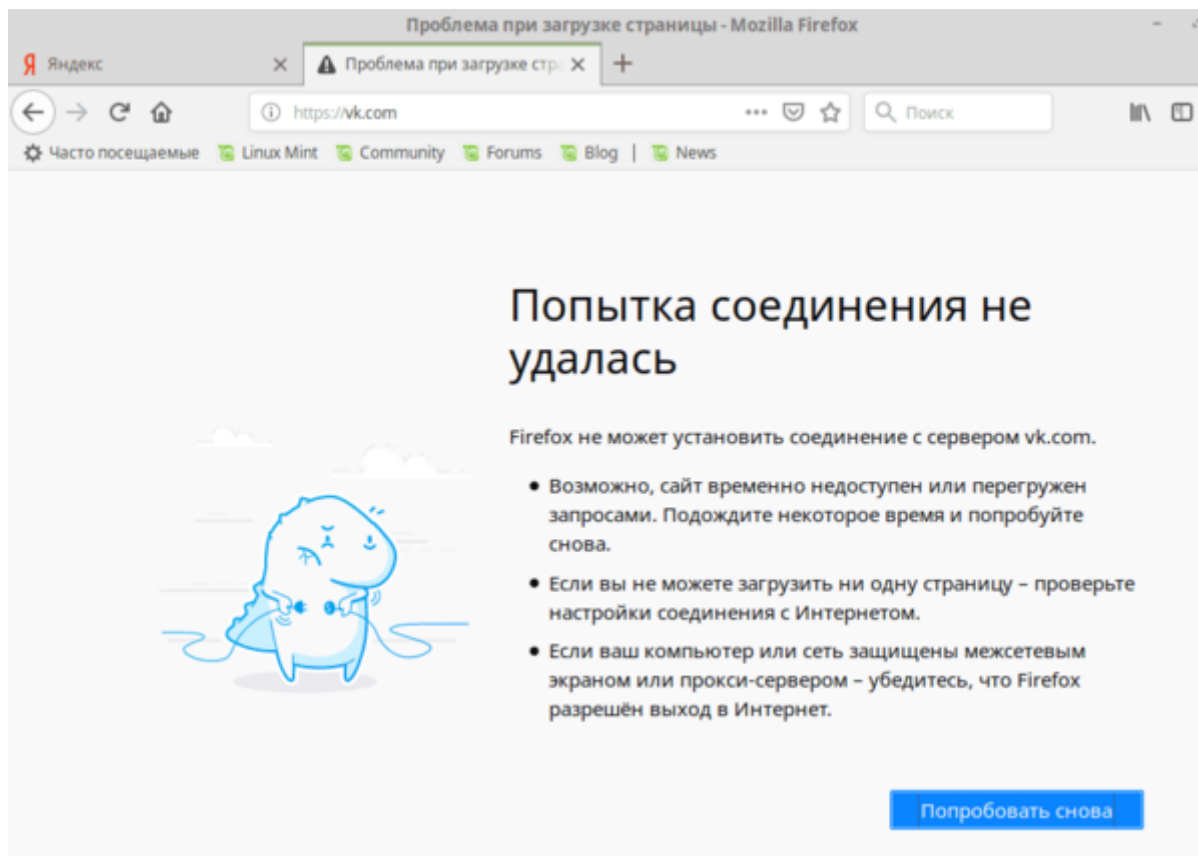
Данное правило должно располагаться самым первым в цепочке FORWARD, выше FastTrack.

Теперь попробуем посетить запрещенный сайт:



Страничка не грузится, однако браузер не понимает в чем дело и продолжает попытки ее грузить (т.е. в заголовке вкладки постоянно крутится колесико), это нагружает ресурсы ПК и не вносит ясности пользователю. Так происходит из-за того, что мы просто убиваем пакеты - действие **drop**, и отправитель не понимает почему нет ответа. Поэтому для внутренних ресурсов лучше использовать действие **reject**, которое отправляет назад пакет с сообщением что ресурс недоступен.

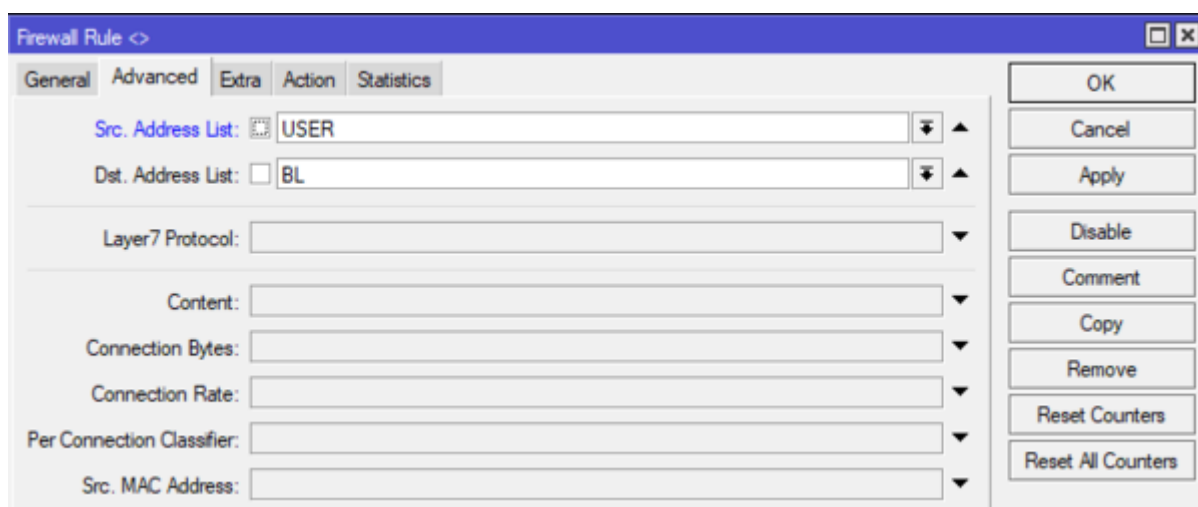
После замены действия при повторной попытке посетить ресурс мы сразу увидим сообщение о его недоступности:



Быстро добавить правило через командную строку можно так:

```
/ip firewall filter
add action=reject chain=forward dst-address-list=BL in-interface=bridge1 reject-
with=icmp-network-unreachable src-address-list=!USER
```

Теперь немного изменим задачу, применим черный список только к группе USER. Для этого немного изменим условия на закладке **Advanced**, а именно укажем **Src. Address List - USER** без восклицательного знака, в итоге условие будет читаться как: если источник в группе USER и назначение в группе BL.



Или в командной строке:

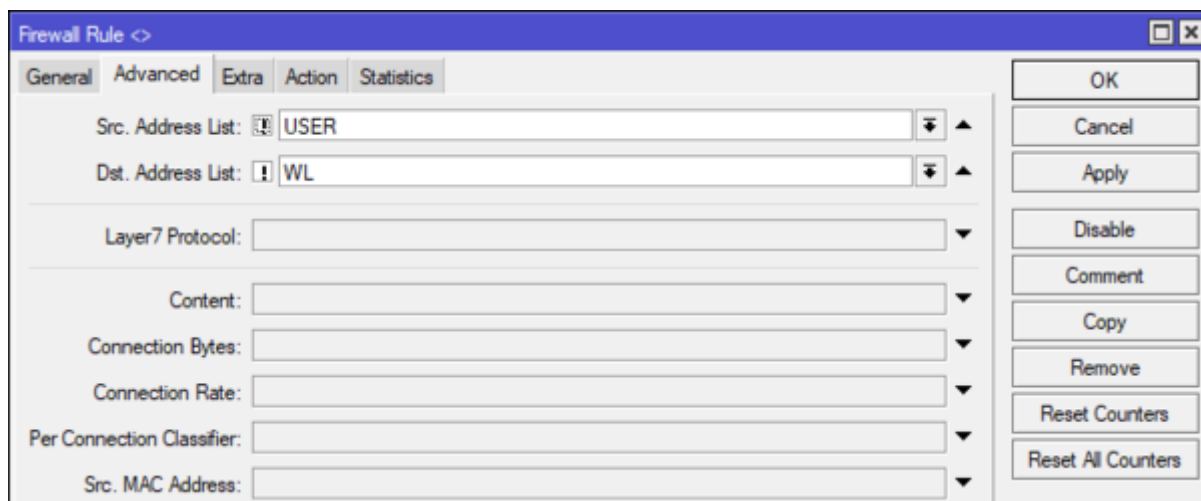

```
/ip firewall filter
add action=reject chain=forward dst-address-list=BL in-interface=bridge1 reject-
with=icmp-network-unreachable src-address-list=USER
```

Таким образом фильтрация по черным спискам не представляет особых сложностей. Все упирается в эти самые списки, которые нужно составлять и поддерживать в актуальном состоянии. Загрузить в роутер готовые списки из интернета также не очень хорошая идея, потому как каждый пакет будет проверяться на вхождение в список, что может вызвать серьезную нагрузку на роутер, при том, что подавляющее большинство адресов из этого списка ваши пользователи могут никогда не посещать. Поэтому следует трезво оценивать собственные ресурсы и возможности и применять списки там, где это действительно нужно.

Белые списки

На первый взгляд организация доступа в сеть по белым спискам ничем принципиально не отличается от черных, однако это не так, выше мы уже говорили почему и далее покажем это на примерах. А пока реализуем схему с доступом по белым спискам для всех, кроме группы USER.

Снова перейдем в **IP - Firewall - Filter Rules** и создадим новое правило. На закладке **General** также укажем **Chain - forward** и **In. Interface - bridge1**, на **Advanced** указываем **Src. Address List - !USER** и **Dst. Address List - !WL**:



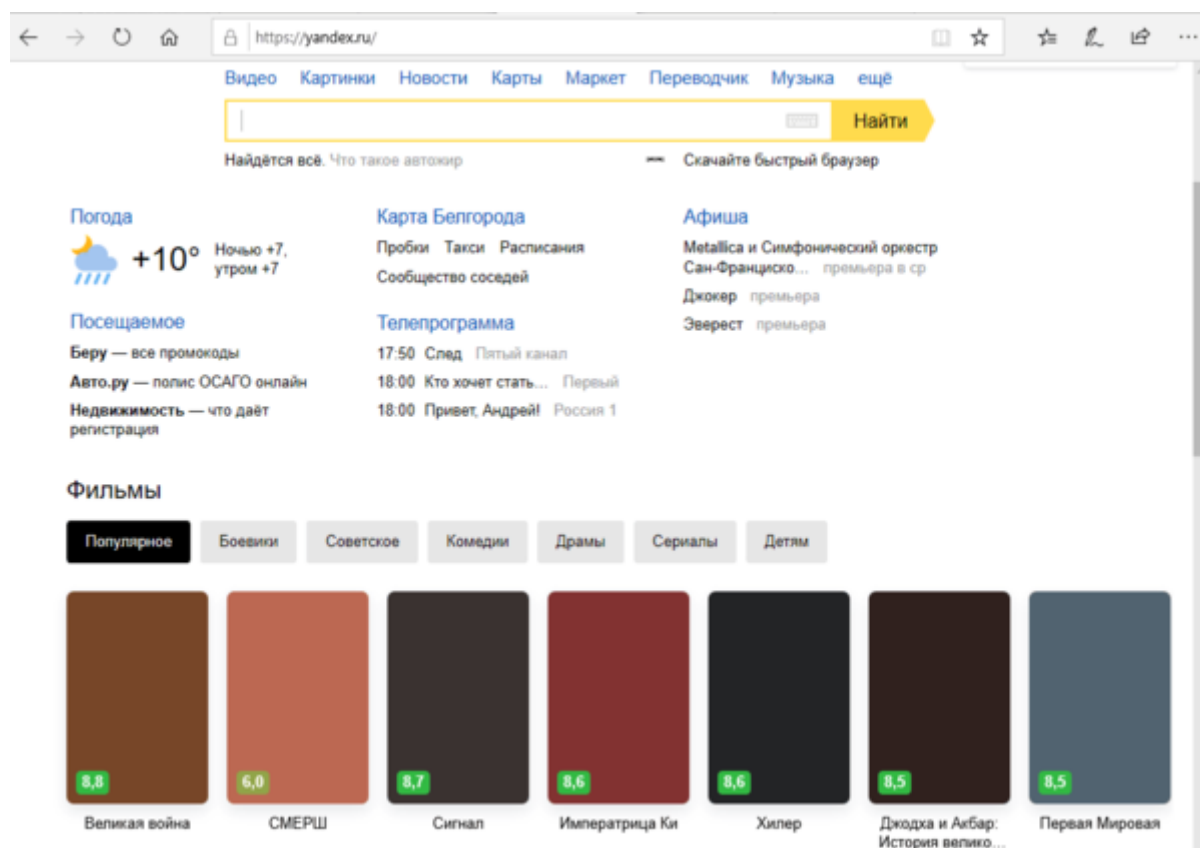
И на закладке **Action** указываем действие **reject**. Таким образом данное правило будет блокировать все соединения, если адрес отправителя не входит в группу USER и адрес назначения не входит в белый список WL.

Аналогичное действие через консоль:

```
/ip firewall filter
add action=reject chain=forward dst-address-list=!WL in-interface=bridge1 reject-
with=icmp-network-unreachable src-address-list=!USER
```

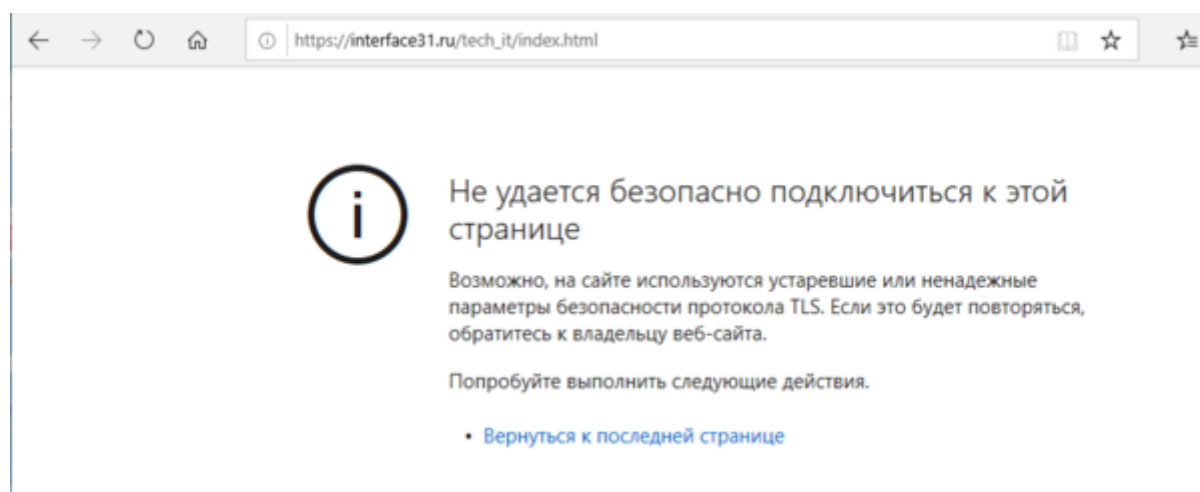
Данное правило также следует располагать первым в цепочке FORWARD.

Добавим к разрешенным несколько адресов, в нашем случае **yandex.ru** и **interface31.ru** и попробуем открыть один из них. Яндекс открывается, но выглядит довольно непривычно.



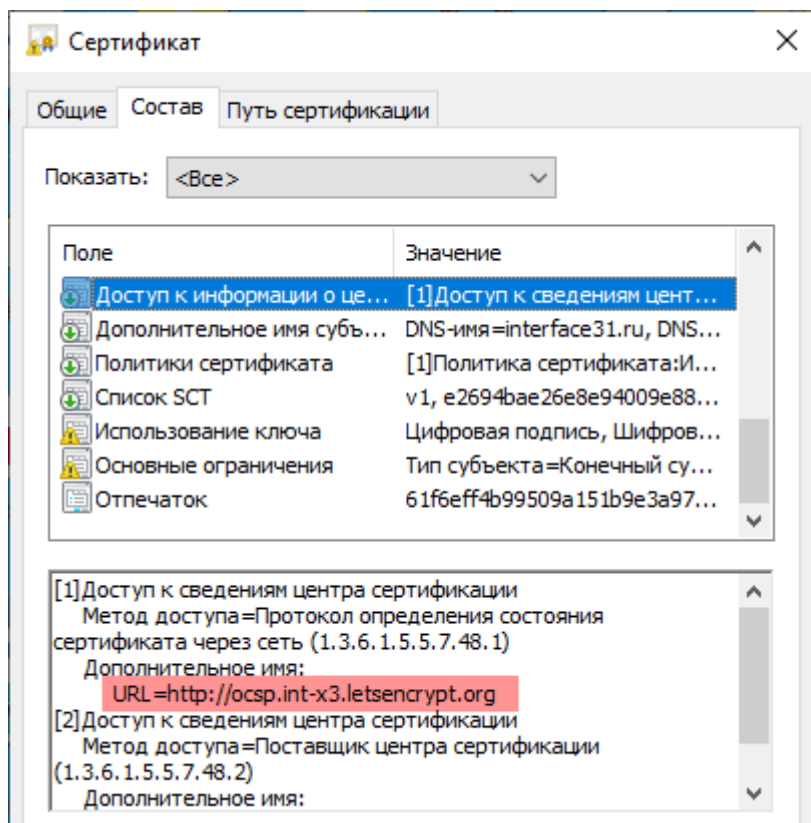
Многие картинки, которые располагаются на иных серверах, включая сервера самого Яндекса, но имеющего другие IP-адреса просто не подгружаются. Хотя никаких фатальных последствий это не несет, как поисковик Яндекс работает. А вот в почту войти уже не получится, для этого придется разрешить как минимум **mail.yandex.ru** и **passport.yandex.ru**.

Теперь попробуем открыть наш сайт. А вот тут первый неприятный сюрприз:



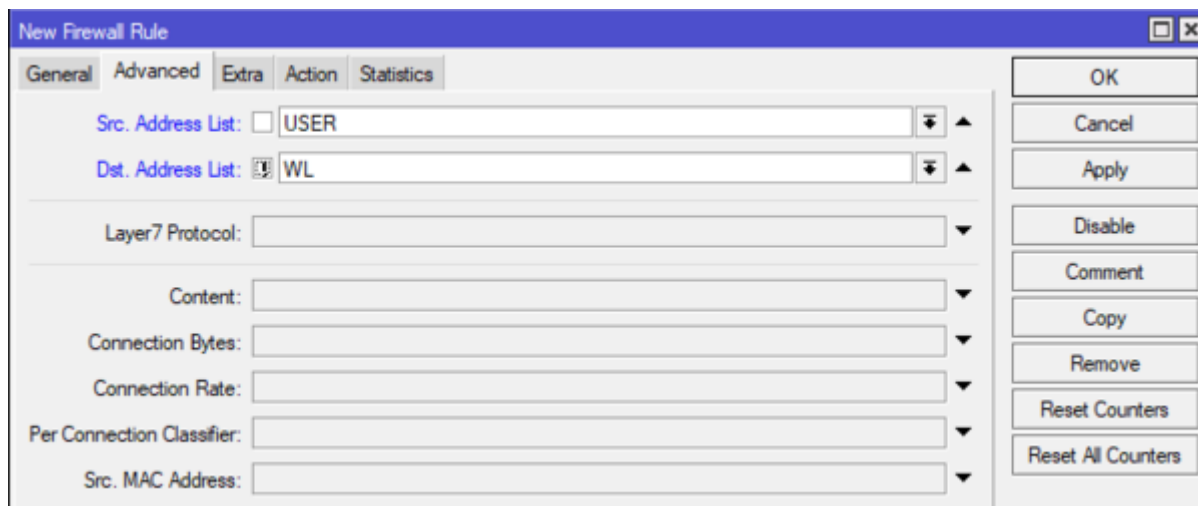
Что это значит? Браузер не может проверить подлинность сертификата, а так как наш сайт использует HSTS, то доступ к нему будет невозможен, потому как подобные действия могут указывать на атаку с понижением степени защиты, чему HSTS должен препятствовать.

Для того, чтобы браузер смог проверить сертификат нам нужно разрешить доступ к сведениям центра сертификации, адреса нужных узлов можно найти в самом сертификате сайта:



В нашем случае оказалось достаточно добавить узел **ocsp.int-x3.letsencrypt.org**, сайт загрузился, но без комментариев, так как они реализованы на стороннем ресурсе и разрешение доступа к нему не решило проблемы. В этом случае вам придется либо заниматься долгим и кропотливым выяснением необходимых для работы ресурсов и занесением их в белый список, либо отказываться от части функционала сайтов.

Чтобы применить белый список только к участникам группы немного изменим правило: в **Advanced** указываем **Src. Address List - USER**, т.е. без восклицательного знака. Теперь логика правила изменится и будут блокироваться все соединения для группы USER, кроме тех, которые разрешены белым списком.



Либо в командной строке:

```
/ip firewall filter
add action=reject chain=forward dst-address-list=!WL in-interface=bridge1 reject-
with=icmp-network-unreachable src-address-list=USER
```

Как видим, технически организовать доступ по белым спискам не так уж сложно, гораздо сложнее обеспечить полноценную работу разрешенных сайтов, что требует достаточно долгой и кропотливой работы по выявлению и добавлению в список связанных ресурсов.

Layer 7 protocol

Layer 7 protocol - это методика поиска определенных вхождений в ICMP/TCP/UDP потоках при помощи регулярных выражений. На первый взгляд достаточно интересная возможность, существенно расширяющая степень контроля над проходящим трафиком, но есть один существенный недостаток. Как уже понятно из названия, данный вид фильтрации работает на прикладном (L7) уровне, т.е. полностью обрабатывается CPU и даже при небольшом количестве правил способен создать сильную нагрузку на оборудование, особенно старые (не ARM) модели.

Использовать L7 для блокировки сайтов не рекомендуют сами разработчики Mikrotik, справедливо замечая, что в большинстве случаев это не будет работать так, как задумано, но при этом вы будете впустую растрачивать вычислительные ресурсы роутера. На наш взгляд использовать L7 для задач, связанных с доступом к сайтам вообще бессмысленно. Современный трафик в подавляющем большинстве зашифрованный и различного рода конструкции для анализа URL просто не будут работать, а управлять доступом на основе доменного имени вполне можно и на L3 (чем мы занимались выше).

По этой же самой причине не будут работать многие размещенные в интернете инструкции, где трафик фильтровался по содержимому, типам файлов или потоков, использовал параметры запросов и т.д. и т.п. Хотя мы до сих пор встречаем статьи, в которых по L7 пытаются блокировать соцсети или Youtube, мотивируя это

большим числом адресов, использованием CDN, поддоменов и т.д. и т.п. Однако все это не выдерживает никакой критики, соцсети и видеохостинги прекрасно блокируются по доменному имени.

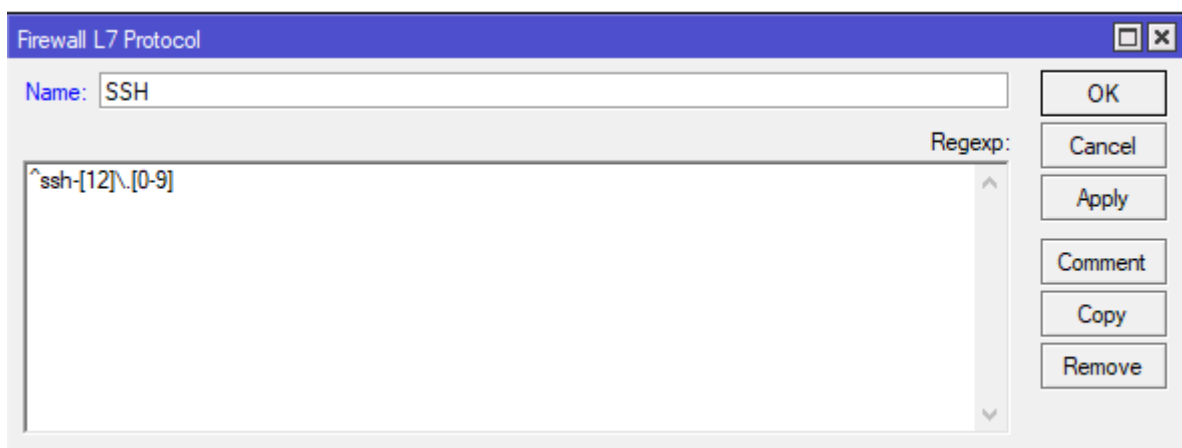
Мы не рекомендуем использовать L7 во всех тех случаях, когда задачу можно решить иным образом, применяя его только для решения специфичных задач. Например, выявления и блокировки какого-либо вида трафика.

Поставим для примера следующую задачу: заблокировать возможность установления SSH-соединений для клиентов сети. Решение в лоб - заблокировать исходящие соединения на 22 порт не принесет успеха, так как SSH-сервер может работать на произвольном порту. Поэтому нужно при помощи специальных паттернов определить наличие именно SSH-трафика и каким-то образом его заблокировать.

Где брать паттерны? Опытные пользователи могут запустить сетевой сканер (tcpdump, Wireshark) и проанализировать доступное содержимое пакетов и на основании полученной информации составить регулярное выражение. Либо воспользоваться сайтом l7-filter.sourceforge.net, однако большая часть паттернов оттуда работать не будет. Во-первых, сайт достаточно старый, последний раз обновлялся в 2009 году, во-вторых, очень многие протоколы перестали использоваться в открытом виде, а используют SSL-шифрование. В этом случае вы просто увидите SSL-поток, заблокировать который бессмысленно, так как вы заблокируете практически весь интернет.

Для решения нашей задачи сначала перейдем в IP - Firewall - Layer 7 protocol и создадим новый фильтр: в поле Name напомним произвольное имя, в нашем случае SSH, а в поле Regexp внесем регулярное выражение паттерна:

```
^ssh-[12]\.[0-9]
```

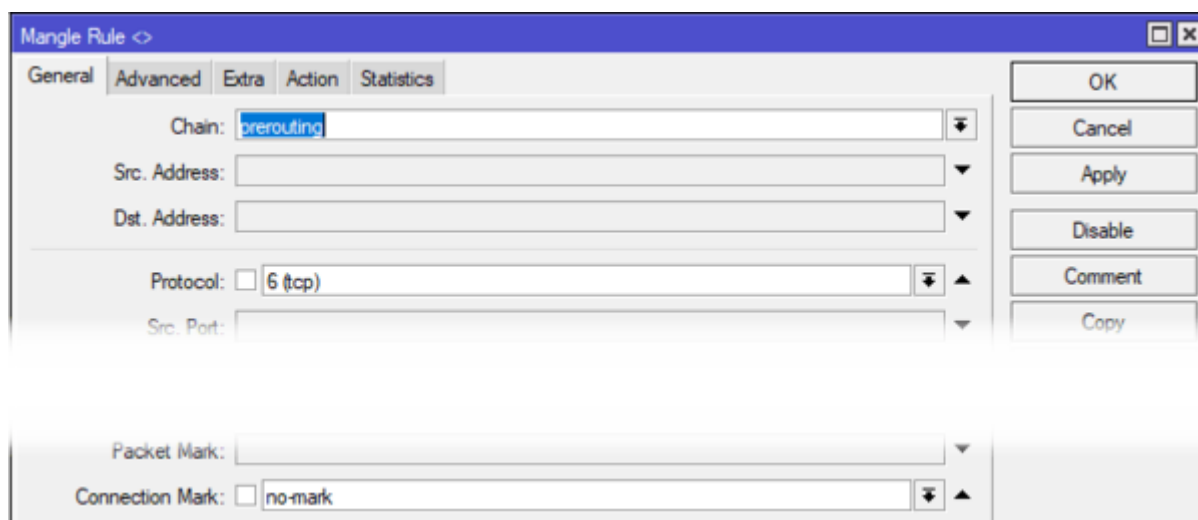


Также можно выполнить команду в терминале:

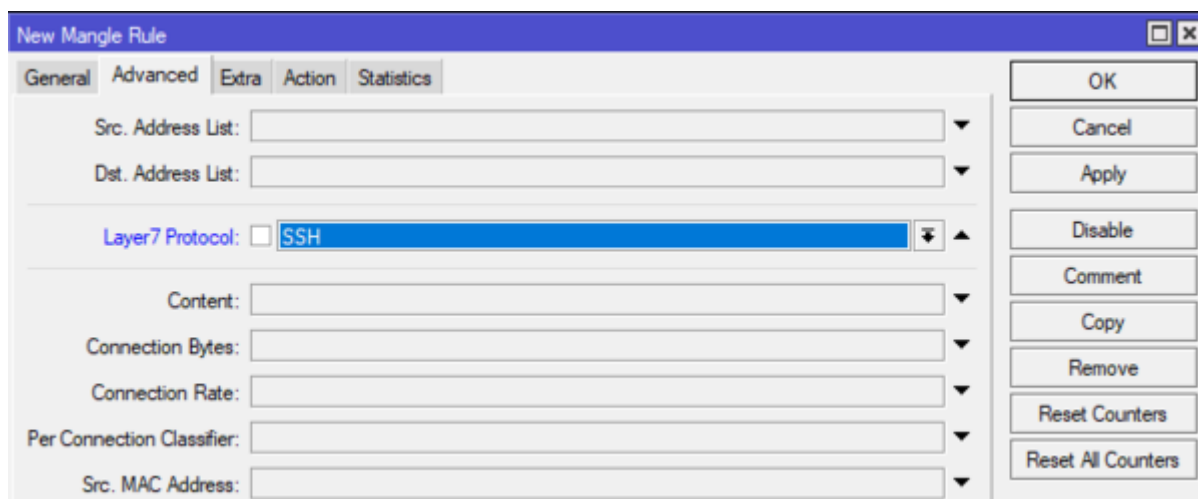
```
/ip firewall layer7-protocol  
add name=SSH regexp="^ssh-[12]\.[0-9]"
```

Что делать дальше? Самое очевидное решение - использовать данный фильтр в правилах брандмауэра является примером того, как делать не надо. В этом случае через L7 фильтр будет проходить **каждый** пакет, что вызовет сильную нагрузку на CPU роутера.

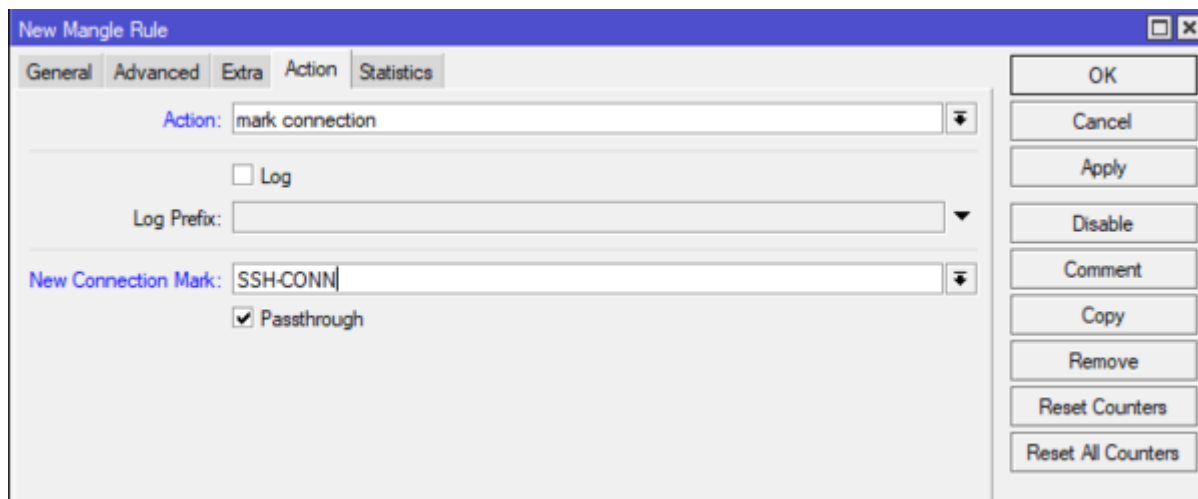
Поэтому мы пойдем другим путем и на основании L7 фильтра будем **маркировать соединения**, которых гораздо меньше, чем пакетов. Перейдем в **IP - Firewall - Mangle** и создадим новое правило: на закладке **General** выставляем **Chain - prerouting**, **Protocol - tcp** и **Connection Mark - no mark**:



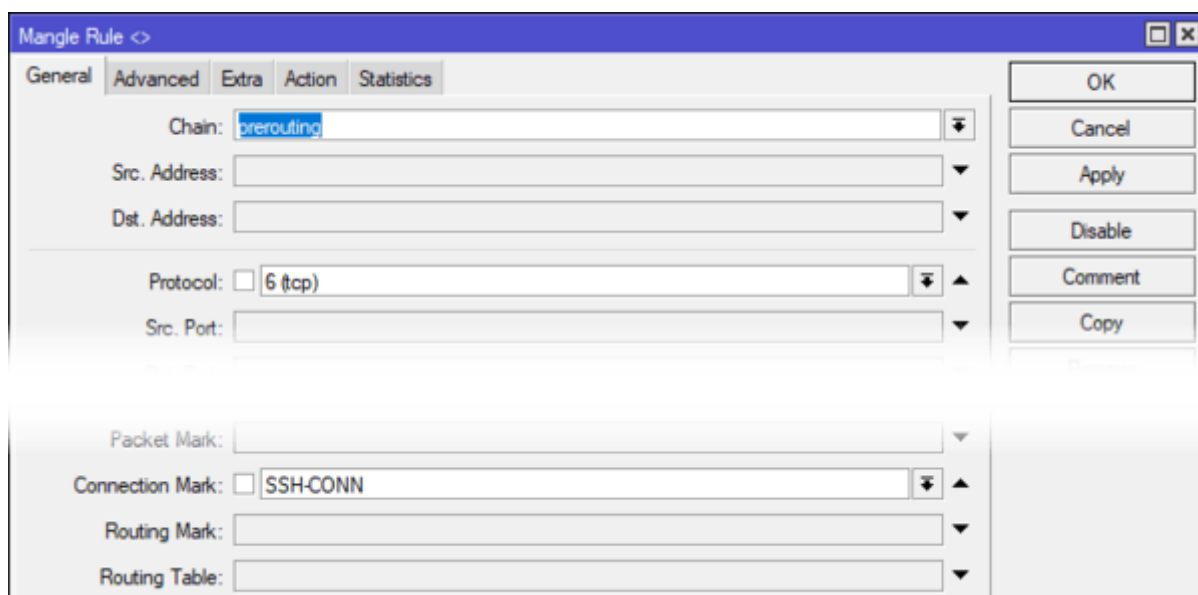
На закладке **Advanced** указываем использование созданного нами фильтра **Layer 7 Protocol - SSH**:



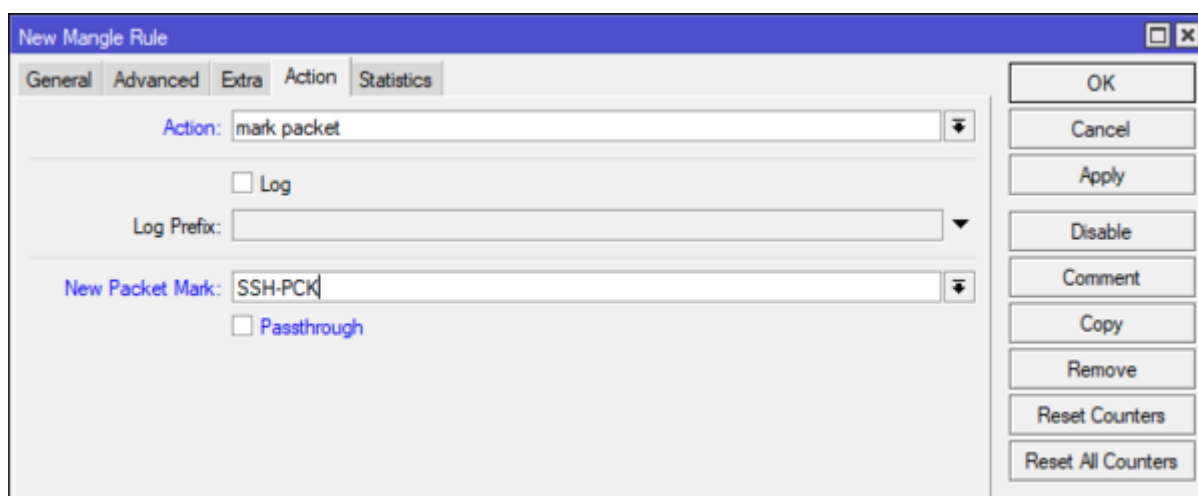
В **Action** указываем действие **mark-connection**, задаем марку соединения **New Connection Mark - SSH-CONN** и обязательно ставим флаг **Passthrough** для прохождения пакета далее по цепочке:



Затем добавим еще одно правило: **General - Chain - prerouting, Protocol - tcp и Connection Mark - SSH-CONN**:



А в действиях добавим **mark packet, New Packet Mark - SSH-PCK** и снимем флаг **Passthrough**:

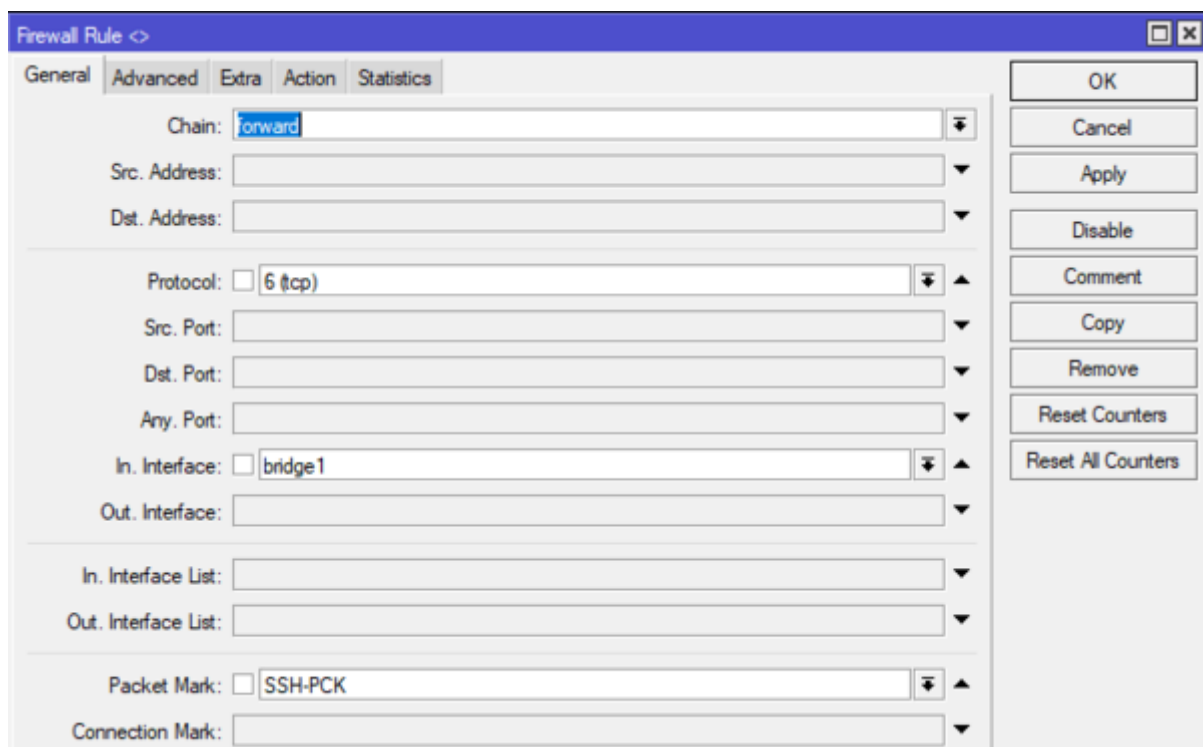


Все тоже самое быстро делается в командной строке:

```
/ip firewall mangle
add action=mark-connection chain=prerouting connection-mark=no-mark layer7-
protocol=SSH new-connection-mark=SSH-CONN passthrough=yes protocol=tcp
add action=mark-packet chain=prerouting connection-mark=SSH-CONN new-packet-
mark=SSH-PCK passthrough=no protocol=tcp
```

Многие читатели не работают с брандмауэром дальше таблицы Filter, поэтому что, что мы сейчас сделали в Mangle может показаться им какой-то особой магией. Коротко поясним наши действия. Первое правило проверяет все немаркированные соединения и те из них, которые сосуществуют фильтру L7, т.е. SSH-соединения получают метку SSH-CONN и продолжают движение по цепочке. Следующее правило проверяет соединения и все пакеты соединений, промаркированных как SSH-CONN снабжает меткой SSH-PCK.

Таким образом мы пометили все пакеты, относящиеся к SSH-соединениям, но L7 фильтр мы используем только для соединений, не нагружая роутер проверкой каждого пакета. Теперь запретим транзит таких пакетов, для этого вернемся в **IP - Firewall - Filter Rules** и создадим правило, на закладке **General** которого укажем: **Chain - forward, Protocol - tcp, In Interface - bridge1** и **Packet Mark - SSH-PCK**:



На закладке **Action** ставим действие **drop**. То же самое в консоли:

```
/ip firewall filter
add action=drop chain=forward in-interface=bridge1 packet-mark=SSH-PCK
protocol=tcp
```

Ставим это правило также в начало цепочки FORWARD и если вы все сделали правильно, то установить SSH-соединение из вашей сети больше никому не удастся.


```
andrey@andrey-mint ~
Файл  Правка  Вид  Поиск  Терминал  Справка
andrey@andrey-mint ~ $ ssh [redacted].ru
ssh_exchange_identification: read: Connection timed out
andrey@andrey-mint ~ $
```

Следует понимать, что выше был лишь пример того, как можно использовать Layer 7 protocol на Mikrotik, в реальной ситуации следует несколько раз подумать и прибегать к возможностям L7 только тогда, когда все остальные варианты исчерпаны. Также старайтесь как можно более подробно описывать условия, для правил использующих L7 фильтры, чтобы максимально уменьшить нагрузку на процессор роутера.

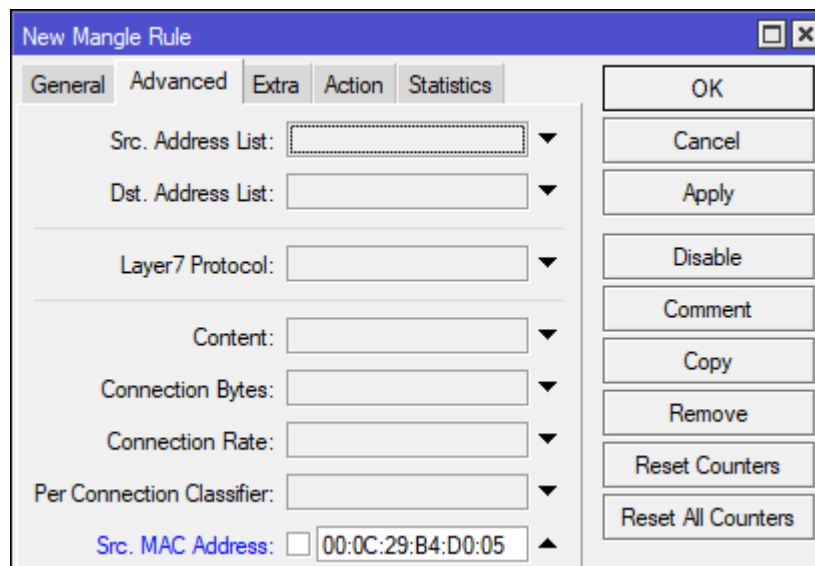
Фильтрация по MAC-адресам

Наши читатели с завидным постоянством спрашивают нас как можно организовать фильтрацию по MAC-адресам. Мы уже говорили и повторим еще раз, что считаем такую фильтрацию не самым оптимальным способом, потому что для идентификации следует использовать более высокоуровневые параметры: пользователя или IP-адрес. Но если сильно хочется, то почему бы и нет.

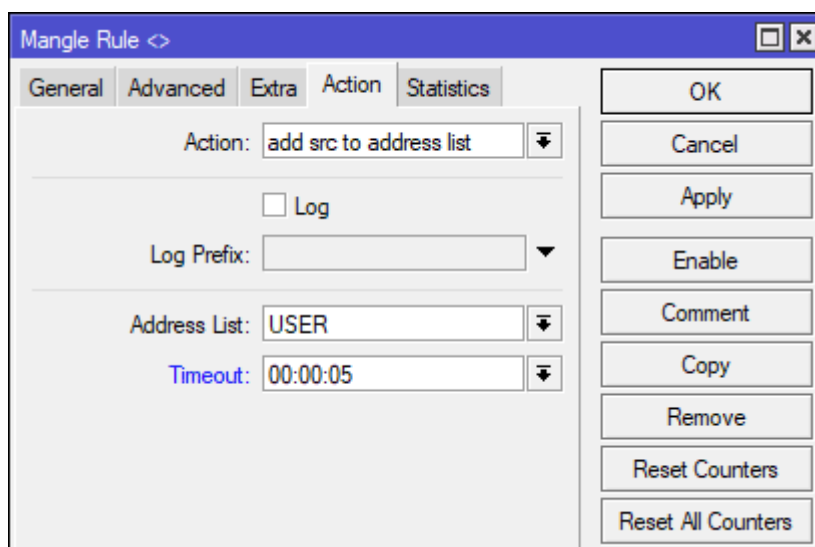
Среди условий в правилах брандмауэра есть опция MAC-адреса, но в одном правиле можно указать только один адрес, т.е. для каждого MAC вам придется создать свою копию правила, что увеличит нагрузку на устройство и сделает набор правил трудночитаемым.

В тоже время MAC-адрес нам нужен для одной единственной цели - идентифицировать пользователя, что мы также можем сделать и по IP-адресу, для этого нам нужно будет преобразовать MAC в IP, который уже можно добавить в один из списков и использовать представленные нами выше правила. В этом нам снова поможет таблица Mangle.

Откроем **IP - Firewall - Mangle** и добавим правило, на закладке **General** укажем **Chain - prerouting**, **In Interface - bridge1**, на **Advanced** в поле **Src. MAC Address** укажем MAC-адрес нужного устройства.



И на закладке **Action** добавим действие **add src to address list**, где в поле **Address List** укажем требуемый список пользователей, в нашем случае **USER**, а в поле **Timeout** укажите требуемое время жизни записи, это нужно для того, чтобы запись обновилась при смене обладателем MAC IP-адреса. На скриншоте мы, в тестовых целях, использовали 5 секунд, в реальной жизни руководствуйтесь здравым смыслом и выбирайте более высокие значения.



Это же правило в командной строке:

```
/ip firewall mangle
add action=add-src-to-address-list address-list=USER address-list-timeout=5s
chain=prerouting in-interface=bridge1 src-mac-address=00:0C:29:B9:FF:2E
```

Теперь первый пришедший с данного устройства пакет добавит его IP-адрес в указанный нами список, тем самым связав его с текущим MAC на время указанное в **Timeout**. Для каждого следующего устройства необходимо создать подобное правило, также не забывайте снабжать каждое из них комментарием, чтобы впоследствии вам и вашим коллегам было понятно о каком именно устройстве идет речь.

Заключение

Как видим возможности RouterOS позволяют решать достаточно сложные задачи используя даже недорогие роутеры. Но следует понимать ограничения всех вышеперечисленных методов, осознавая их достоинства и недостатки. А также соотносить свои требования с возможностями оборудования. Если понимать и принимать во внимание эти факторы, то фильтрация по спискам на Mikrotik будет эффективным инструментом в руках администратора. В противном случае вы получите только разочарование и иные негативные последствия. Поэтому пожелаем вам благоразумия и напомним: хороший администратор выбирает для каждой задачи наиболее подходящий инструмент, что является признаком профессионализма. А фанатизм еще никого до добра не доводил.

Онлайн-курс по MikroTik

Научиться настраивать MikroTik с нуля или систематизировать уже имеющиеся знания можно на [углубленном курсе по администрированию MikroTik](#). Автор курса, сертифицированный тренер MikroTik Дмитрий Скоромнов, лично проверяет лабораторные работы и контролирует прогресс каждого своего студента. В три раза больше информации, чем в вендорской программе MTCNA, более 20 часов практики и доступ навсегда.
