

Настройка Зпроху для чайников

 bozza.ru/art-94.html

[главная](#) - [Статьи](#) - [Microsoft Windows](#) - [Прокси-сервер Зпроху](#)

Дата обновления: 18.03.2020

Теги: [Зпроху](#) [Настройка прокси](#)

Пара ремарок перед началом чтения:

- 1. Несмотря на то, что на сайте Зпроху.ру гордо отображается сообщение о том, что антивирусы могут воспринимать Зпроху как вирус, я с таким не сталкивался. Возможно, это относится к исходникам и компиляции, а не к бинарникам. Все же современные антивирусы могут собрать статистику, что файл с такой-то контрольной суммой скачан 100500 раз и не начинают голосить почему зря. Кто знает...*
- 2. Несмотря на то, что оригинал статьи не новый (в оригинале речь идет про версию 0.5.2, а на текущий момент стабильная версия 0.8.13), все подойдет и будет работать. Начинайте читать и совсем скоро вы увидите, что этот проху действительно может быть простым.*

Что это вообще такое?

Зпроху - это прокси сервер (проху, socks, порЗпроху) для Windows, Linux, Unix.

Исходники открыты, бинарники скомпилированы, инструкции по компиляции указаны. Много лет развития. Что еще надо? Вперед!

Проще некуда

Для затравки, если уж "для чайников", да простят меня читатели, то вот простой конфиг Зпроху.cfg:

```
auth none  
log  
проху
```

Куда уж проще... Вообще, Зпроху хорош тем, что он может быть и очень простым и лаконичным. Да и запуск его не сложен вовсе.

Скачали с сайта архив, распаковали в c:\Зпроху, создали текстовый файл Зпроху.cfg и положили его в папку с exe-файлом, запустили cmd:

```
C:\Users\>cd "C:\3proxy\bin"  
C:\3proxy\bin>3proxy.exe 3proxy.cfg  
200318221136.662 3128 00000 - 0.0.0.0:0 0.0.0.0:0 0 0 0 Accepting connections  
[8996/8360]
```

Все, ничего не устанавливая, мы запустили проху сервер, который принимает соединения на порт 3128. Можно прописать даже у себя в браузере, 127.0.0.1 3128 и все уже начинает работать, вы сразу увидите, как в консоли побегут строчки лога.

В этом конфиге:

- **log** без параметров означает выводить лог на stdout (на экран, в общем)
- **proxy** - запускаем именно прокси, а не socks, на стандартном порту 3128
- **auth none** - не запрашивать авторизацию, открытый прокси.

Оригинал с правками

Автор статьи: Kurmaeff Halit
halit_at_mail_dot_ru

Хочу поделиться своим опытом по настройке замечательного прокси-сервера от Заразы - 3proxy. Долгое время пользовался другим интересным прокси-сервером ES Proxy <http://esproxy.org.ua> от Георгия Павленко - маленького (всего один ехе-шник на 300 кБ), довольно простого в настройке и очень нетребовательного к ресурсам - по крайней мере, при почти сотне пользователей он умудрялся работать на старинном P133, почти не затыкаясь. И только вот это <почти> - иногда все же проксик самопроизвольно пожирал 100% процессора, хотя и продолжал частично или полностью справляться со своими обязанностями - а также то, что автор уже почти 2 года как не подает никаких признаков жизни, то есть продукт замер в развитии - привели к поискам другого подходящего сервера. Скажу сразу - прокси-сервера с красивыми GUI меня привлекали мало, я сторонник принципа <Мне чтобы ехать>, а красивый интерфейс - это, конечно, хорошо, но не в ущерб производительности. Перепробовав не от хорошей жизни 5-6 разных продуктов (среди них Usergate, Proxy+, Lan4net....) и не оставшись довольным получаемыми результатами (должен заметить, что известный продукт от Майкрософт я решительно отмел ввиду его высокой стоимости), решил попробовать таки 3проху, тем паче слышал в основном хорошие отклики. Если кто и жаловался на него, то только на сложность настройки, в чем, должен скачать, немного забегаю вперед, я и имел возможность убедиться.

Найдя в интернете сайт разработчика, почитал про возможности - и возрадовался. Тут тебе и любые лимиты на каждого пользователя, и ограничение скорости, и FTP, и SOCKS, и порт-мапперы, и даже работа как под Windows, так и под любыми никсами. Более того, есть и некоторые уникальные возможности - вроде ведения журналов через ODBC или задание произвольного формата логов, так что логи можно будет <скормить> любому лог-анализатору. В общем, всё, что может понадобиться системному администратору, чтобы и пользователей удовлетворить,

и не пускать их же куда не нужно... Не хватает разве что кэширования и контент-фильтра да еще пары мелочей, но это все обещается уже в скором будущем, да и по содержанию форума, оперативным ответам на вопросы пользователей и содержимому changelog'a версии 0.6 видно, что работа над совершенствованием сервера идет полным ходом.

Скачав дистрибутив (260 кБ! в стабильной на момента написания статьи версии 0.5.2), я погрузился в изучение образца конфигурационного файла, который прилагался в комплекте. По ходу изучения возникло немало вопросов. С того же сайта пришлось скачать и распечатать, чтобы неспешно изучить, FAQ и HowTo. Когда и эти материалы были изучены, оказалось, что вопросов меньше не стало. Попытки запустить прокси были первоначально безуспешными, пришлось почитать материалы конференции, где и был найден образец очень простого конфигурационного файла, построенного по принципу <разрешить все и всем>. Кстати, вот он (версия 0):

```
auth none  
log  
proxy
```

Все гениальное, как говорится, просто. А моя ошибка заключалась, кстати, как раз в том, что я пытался сразу написать рабочий конфиг со всеми лимитами и т.д.

Важно! Конфиг файл в общем случае должен находиться в одной папке с исполняемым файлом 3proxy.exe (по умолчанию в папке bin). Иначе надо запускать исполняемый файл с указанием пути к конфигу. В 90% случаях этот конфиг будет работать. Выше мы уже прокомментировали этот конфиг.

- В первой строчке указывается, что авторизация не требуется, соответственно, доступ получают все юзеры без исключения. Если вы попытаетесь ввести после этой команды какие-то ограничения доступа, то они не сработают, о чем и сообщается в FAQ.
- Во второй строке задаем ведение лога, при этом не указываем путь к файлу, следовательно, лог будет выводиться на экран.
- И, наконец, запускаем сам прокси на стандартном порту 3128.

Многие параметры опущены, соответственно, для них прокси возьмет значения по умолчанию. Теперь можете на всех пользовательских компьютерах настроить ваш любимый браузер - не знаю, как у вас, а у меня это почетное место уже лет 6 занимает Opera - в его сетевых настройках поставить галочку "использовать прокси", а затем для всех протоколов указать IP вашего прокси-сервера и порт 3128. Если вам не нужен учет и контроль, мало интересует безопасность сети, а трафик у вас безлимитный, то можно этим и ограничиться, по крайней мере на первое время. Можно даже удалить вторую строчку, она не обязательна.

В начале конфига можно было бы еще поставить строчку `service`, для того, чтобы сервер запускался как служба, а не отдельным приложением. В этом случае его надо будет предварительно установить как службу командой:

```
3proxy --install
```

Но не спешите этого делать! Дело в том, что будучи запущен приложением, 3проху будет сразу выдавать на экран сообщения об обрабатываемых запросах и ошибках и вам не придется смотреть текстовые логи. Причем, что особенно радует, с указанием кода ошибки и номера строки в конфиге, где эта ошибка присутствует. А на стадии обучения это очень помогает.

Да, и еще одно важное замечание насчет конфига - **никаких лишних пробелов, табуляций и т.п. знаков в начале строки**, иначе строка конфига будет считаться закоментированной и значит, не будет обрабатываться.

Техническое задание

Опишем предполагаемую ситуацию. Надо "пустить интернет" на 3 компьютера с IP адресами `192.168.0.2`, `192.168.0.3`, `192.168.0.4`, в качестве прокси-сервера выступает компьютер с двумя сетевыми картами, внутренним IP-адресом `192.168.0.1` и "внешним", то есть полученным от провайдера адресом `211.80.11.12`. От провайдера же получены адреса DNS-серверов: `213.153.100.1` и `213.153.101.1`. У провайдера доступен SMTP-сервер `smtp.provider.ru`.

На компьютере с адресами `192.168.0.2-192.168.0.4` работают пользователи Иванов, Петров, Сидоров соответственно.

Установка 3проху

Прокси-сервер установим в папку `c:\3proxy`, хотя установим - это громко сказано, для начала его можно просто распаковать из zip-архива в заданную папку, пусть это будет `c:\3proxy` (про инсталляцию службы было сказано выше).

Конфиг

Теперь последовательно займемся дальнейшим улучшением нашего замечательного конфига.

Для начала укажем внутренний:

```
internal 192.168.0.1
```

- вы ведь понимаете, что здесь должен стоять IP сетевой карты ВАШЕГО прокси сервера, настроенной на локальную сеть. Эта команда задает именно тот IP адрес в локальной сети, к которому будут обращаться клиенты и который, соответственно, будет прописан в браузерах в качестве прокси-сервера

и внешний интерфейсы для прокси-сервера:

```
external 211.80.11.12
```

- а этот IP адрес вы должны были получить от провайдера, как правило, в случае выделенной линии это так называемый "реальный" IP, который должен быть доступен из интернета (хотя это не обязательно).

Необходимое замечание - командами `external` и `internal` мы задаем внутренние и внешние интерфейсы, так сказать, глобально, на весь конфиг и на все службы. Если есть необходимость для отдельных служб задавать свои интерфейсы, то это можно сделать с помощью параметров службы `-e` и `-i`, например, аналогом совокупности команд:

```
internal 192.168.0.1
external 211.80.11.12
proxy
```

будет команда

```
proxy -i192.168.0.1 -e211.80.11.12
```

Теперь для ускорения процесса разрешения имен хостов укажем DNS-сервера - лучше всего указать DNS-сервера вашего провайдера - и запустим кэширование DNS:

```
nserver 213.153.100.1
nserver 213.153.101.1
nscache 65536
```

где вместо 65536 может быть любое, желательно представляющее степень числа 2 и достаточно большое. Итак, на данном этапе наш конфиг выглядит следующим образом (версия 1):

```
internal 192.168.0.1
external 211.80.11.12
nserver 213.153.100.1
nserver 213.153.101.1
nscache 65536
log
auth none
proxy
```

Кажется, нам еще не хватает полноценного логирования всех доступов и ошибок, ведь на экране лог очень быстро пролетает! Поменяем команду `log` на более расширенную

```
log c:\3proxy\logs\3proxy.log D
```

Таким образом, мы даем команду сохранять логи в папке `c:3proxylogs` и каждый день начинать новый лог (опция `D`). Еще раз напомним, что на стадии тестирования лучше все же сделать вывод логов на экран - так удобнее искать ошибки. И еще добавим команду

```
rotate 30
```

для того, чтобы в папке с логами сохранялись лишь последние 30 файлов, а то логи так быстро растут, а диск не резиновый!

Вспоминаем, что 3проху позволяет нам видоизменить формат логов, чтобы можно было их обрабатывать любым лог-анализатором (кстати, могу посоветовать скачать с сайта www.wrsru.ru довольно приличный и, что немаловажно, бесплатный лог-анализатор Wrsru нашего отечественного программиста. Тем более что автор анализатора любезно пошел навстречу моей просьбе и в последней версии этого анализатора (1.65) появилась, наконец, так сказать, оптимизированная поддержка 3проху - именно для логов приведенного ниже формата)

```
logformat "L%C - %U [%d/%o/%Y:%H:%M:%S %z] ""%T"" %E %I %O %N/%R:%r"
```

В этом случае логи примут вид:

```
192.168.0.4 - ivanov - [02/Jun/2006:11:13:56 +0500]  
"GET http://svalka.biz/uploads/posts/1147559374_00.jpg HTTP/1.0" 000 17497 457  
PROXY/http://svalka.biz:3128
```

- где идут по порядку: IP рабочей станции - имя пользователя - число, месяц, год и время - вид запроса - URL - номер ошибки (если 000 - значит, ошибок нет) - принято байт - отправлено байт - сервис - хост - порт, по которому клиент связался с сервисом.

Ну и наконец, если вы хотите еще больше сэкономить места на диске под логи, то их можно архивировать с помощью любого доступного архиватора, например, вот так:

```
archiver "c:\Program Files\WinRAR\rar.exe" rar a -df -inul %A %F
```

(подсказку для опций командной строки WinRAR можно найти в его же Help'e)

Можно собирать продвинутую версию (версия 2) нашего конфига:

```

internal 192.168.0.1
external 211.80.11.12
nserver 213.153.100.1
nserver 213.153.101.1
nscache 65536
logformat "L%C - %U [%d/%o/%Y:%H:%M:%S %z] ""%T"" %E %I %O %N/%R:%r"
log c:\3proxy\logs\3proxy.log D
rotate 30
archiver "c:\Program Files\WinRAR\rar.exe" rar a -df -inul %A %F
auth none
proxy

```

Все лучше и лучше! Продолжаем. Подумаем о том, что пользователей много, а канал, может, и не очень толстый, хорошо, если хотя бы 128 кб, а если диалап на 33.6? Начинаем резать, товарищи!

```
bandlimin 240000 * 192.168.0.2,192.168.0.3
```

- разрешаем качать компьютерам с IP-адресами **192.168.0.2** и **192.168.0.3** со скоростью 24 кбит в секунду, причем это 24 кбит приходятся не на каждый из этих компьютеров, а на оба в совокупности, т.е. если оба будут к примеру, качать одновременно файлы с достаточно быстрых сайтов, то каждому придется только по 12 кбит в секунду (заметьте, килобит, а не килобайт, если надо пересчитать в килобайты, разделите числа на 8)

```
bandlimin 48000 * 192.168.0.4
```

- а этому счастливчику единолично скорость 48 кбит в секунду

```
bandlimout 24000 *
```

- исходящую скорость тоже ограничим всем до 24 килобит в секунду (сейчас эта команда работает только в версии devel 0.6) Ну и наконец, если вы хотите, чтобы эти жесткие ограничения не касались, к примеру, скачивания почты, то снимите ограничения на порт 110:

```
nobandlimin * * * 110
```

и не забудьте поставить эту команду ПЕРЕД прочими командами ограничения скорости - конфиг обрабатывается последовательно до первого удовлетворяющего условия.

Скорости скачивания-закачивания мы уже задали, продолжаем закручивать гайки. В этом нам помогут команды **deny** и **allow**

```
deny * 192.168.200.4 * 110 * 1-5 18:00:00-23:59:59,00:00:00-08:00:00
```

- запрещаем получение почты пользователю с IP .4 в нерабочее время по будням

```
allow * 192.168.200.4 * * * 1-5 09:00:00-18:00:00
```

- разрешаем доступ в интернет пользователю с IP .4 исключительно с понедельника по пятницу с 9 часов утра до 6 вечера, но зато разрешены запросы на любые порты, то есть можно и с FTP-серверов качать, и HTTPS, к примеру.

```
allow * 192.168.0.2,192.168.0.3 * 80,443 * 1-7 00:00:00-23:59:59
```

- а здесь разрешаем только WEB-серфинг, зато круглосуточно и любой день недели

Кстати вспоминаем (вы ведь уже медитировали над man, FAQ и HowTO, не правда ли?), что команда

```
auth none
```

собственно никакую авторизацию не дает, т.к. отключает любую авторизацию совсем и команды allow и deny после нее теряют всякий смысл. Пора перейти на что-нибудь более подходящее, в нашем случае это авторизация по IP:

```
auth iponly
```

Снова собираем очередную версию конфига с учетом вышенанписанного (версия 3):

```
internal 192.168.0.1
external 211.80.11.12
nserver 213.153.100.1
nserver 213.153.101.1
nscache 65536
logformat "L%C - %U [%d/%o/%Y:%H:%M:%S %z] ""%T"" %E %I %O %N/%R:%r"
log c:\3proxy\logs\3proxy.log D
rotate 30
archiver "c:\Program Files\WinRAR\rar.exe" rar a -df -inul %A %F
nobandlimin * * * 110
bandlimin 24000 * 192.168.0.2,192.168.0.3
bandlimin 48000 * 192.168.0.4
bandlimout 24000 *
auth iponly
deny * 192.168.200.4 * 110 * 1-5 18:00:00-23:59:59,00:00:00-08:00:00
allow * 192.168.200.4 * * * 1-5 09:00:00-18:00:00
allow * 192.168.0.2,192.168.0.3 * 80,443 * 1-7 00:00:00-23:59:59
proxy
```

Продолжим наше увлекательное занятие. Учет и контроль - вот один из основных наших принципов. А какой же учет без счетчиков? Они определяются командами counter, countin и nocountin.

```
counter "c:\3proxy\3proxy.3cf"
```

- этим мы задаем, где у нас будет хранить общий файл хранения всех всех счетчиков. Он представляет собой файл в двоичном формате, причем он всегда один на всех.

Эту команду можно было бы задать в формате

```
counter "c:\3proxy\3proxy.3cf" D "C:\3proxy\traftraf"
```


- тогда каждый день (частота задается в данном случае параметром **D**(ежедневно) - на месте **D** могли быть **W**(еженедельно), **M**(ежемесячно) или **H**(ежечасно)) в папке **C:\3proxy raf raf** будет генерироваться файл отчета в текстовом формате, который позволит нам при необходимости посмотреть статистику за любой день. Посмотрите получаемые отчеты в блокноте, там все понятно.

```
countin "1/Иванов" M 50 * 192.168.0.2 * * *
countin "2/Петров" M 80 * 192.168.0.3 * * *
countin "3/Сидоров" M 30 * 192.168.0.4 * * *
```

- а эти команды задают разные месячные лимиты входящего (счетчик на исходящий трафик, кстати, автор программы тоже обещает сделать) трафика в мегабайтах на компьютеры с IP-адресами **192.168.0.2-192.168.0.4**, причем трафик в нашем случае считается на всех портах и для всех протоколов. Важно - номера счетчиков (в нашем случае 1,2,3) не должны совпадать. Имена, вернее, фамилии пользователей интернета, идущие после косой черты - приведены для удобства, именно они будут читаться на странице статистики, но это позже... Команда **nocountin**, соответственно, позволяет не считать трафик по заданному ряду параметров. Например, командой

```
nocountin * 192.168.0.4 * 110 *
```

помещенной ПЕРЕД счетчиками **countin**, можно отключить подсчет трафика для компьютера с IP .4 при получении почты по 110 порту.

Снова собираем очередную версию конфига с учетом вышенанписанного (версия 4):

```
internal 192.168.0.1
external 211.80.11.12
nserver 213.153.100.1
nserver 213.153.101.1
nscache 65536
logformat "L%C - %U [%d/%o/%Y:%H:%M:%S %z] \"%T\" %E %I %O %N/%R:%r"
log c:\3proxy\logs\3proxy.log D
rotate 30
archiver "c:\Program Files\WinRAR\rar.exe" rar a -df -inul %A %F
nobandlimin * * * 110
bandlimin 24000 * 192.168.0.2,192.168.0.3
bandlimin 48000 * 192.168.0.4
bandlimout 24000 *
counter "c:\3proxy\3proxy.3cf" D "C:\3proxy\traftraf"
nocountin * 192.168.0.4 * 110 *
countin "1/Иванов" M 50 * 192.168.0.2 * * *
countin "2/Петров" M 80 * 192.168.0.3 * * *
countin "3/Сидоров" M 30 * 192.168.0.4 * * *
auth iponly
deny * 192.168.200.4 * 110 * 1-5 18:00:00-23:59:59,00:00:00-08:00:00
allow * 192.168.200.4 * * * 1-5 09:00:00-18:00:00
allow * 192.168.0.2,192.168.0.3 * 80,443 * 1-7 00:00:00-23:59:59
proxy
```

Ну вот, мы уже почти все настроили, но ведь интернет - это не только WEB-серфинг и FTP. Электронная почта! Вот ей мы и займемся. Для начала настроим получение почты. Для этого в составе 3проху имеется свой pop3-прокси:

pop3p

В этом случае надо будет настроить ваши мэйл-клиенты. Если, к примеру, для ящика vasya.pupkin@mail.ru в настройках моего любимого TheBat! в настройках доставки почты (закладка Транспорт) раньше стояли такие параметры:

pop3-сервер: pop.mail.ru
пользователь: vasya.pupkin

то чтобы получать почту через прокси, надо будет их поменять на следующие:

pop3-сервер: 192.168.0.1
пользователь: vasya.pupkin@pop.mail.ru

- внимание, именно последовательность "имя пользователя"@"POP3-сервер вашего ящика", а не ваш адрес.

Для большей наглядности рассмотрим еще вариант настройки клиента:

pop3-сервер: mail.example.com
пользователь: vasya.pupkin@example.com

Тогда для получения почты через прокси пришлось бы поменять настройки клиента на следующие:

pop3-сервер: 192.168.0.1
пользователь: vasya.pupkin@example.com@mail.example.com

Что же касается отправки почты, то разработчик нашего замечательного прокси-сервера рекомендует воспользоваться для этого портмаппингом:

```
tcppm -i192.168.0.1 25 smtp.provider.ru 25
```

то есть, не мудрствуя лукаво, мы просто все запросы по порту 25, по которому, собственно, и отправляются почтовые сообщения, перекидываем на 25-ый же порт почтового сервера провайдера.

В настройках вашего почтового клиента в этом случае для упомянутого выше ящика в общем случае поменяется запись только для SMTP-сервера:

SMTP-сервер: 192.168.0.1

Должен заметить, что ловкий прием с портмаппингом можно было бы использовать и для получения почты. В этом случае используется команда:

```
tcppm -i192.168.0.1 2110 pop.mail.ru 110
```

Есть в этом варианте одна неприятная сторона - если пользователи будут пользоваться ящиками на разных почтовых серверах, благо что их много развелось (yandex, rambler, newmail...), то для каждого почтового сервера придется завести новую запись портмаппинга в нашем конфиге, к примеру:

```
tcppm -i192.168.0.1 3110 pop.yandex.ru 110
```

Впрочем, вряд ли их число даже в большом офисе превысит 10-15.

В этом случае параметры ящика в мэйл-клиенте будут следующие:

pop3-сервер: 192.168.0.1

пользователь: vasya.purkin

и дополнительно надо будет поменять порт на 2110 и не забыть дать пользователям разрешение на этот нестандартный порт:

```
allow * 192.168.0.2,192.168.0.3 * 2110,25,110 * 1-5 00:00:00-20:00:00
```

Какой из способов получения почты выбирать - дело вкуса, мне более привычен портмаппинг, его и будем использовать.

Собираем очередную версию конфига с получением почты через pop3r (версия 5):

```

internal 192.168.0.1
external 211.80.11.12
nserver 213.153.100.1
nserver 213.153.101.1
nscache 65536
logformat "%L%C - %U [%d/%o/%Y:%H:%M:%S %z] \"%T\" %E %I %O %N/%R:%r"
log c:\3proxy\logs\3proxy.log D
rotate 30
archiver "c:\Program Files\WinRAR\rar.exe" rar a -df -inul %A %F
nobandlimin * * * 110
bandlimin 24000 * 192.168.0.2,192.168.0.3
bandlimin 48000 * 192.168.0.4
bandlimout 24000 *
counter "c:\3proxy\3proxy.3cf" D "C:\3proxy\traftraf"
nocountin * 192.168.0.4 * 110 *
countin "1/Иванов" M 50 * 192.168.0.2 * * *
countin "2/Петров" M 80 * 192.168.0.3 * * *
countin "3/Сидоров" M 30 * 192.168.0.4 * * *

auth iponly
allow * 192.168.200.4 * * * 1-5 09:00:00-18:00:00
allow * 192.168.0.2,192.168.0.3 * 80,443 * 1-7 00:00:00-23:59:59
proxy

# ПОЧТА
flush
auth iponly
deny * 192.168.200.4 * 110 * 1-5 18:00:00-23:59:59,00:00:00-08:00:00
allow * 192.168.0.2,192.168.0.3,192.168.0.4 * 25,110 * 1-5 00:00:00-20:00:00
tcppm -i192.168.0.1 25 smtp.provider.ru 25
pop3p

```

в случае получения почты через портмаппинг секция конфига, идущая после комментария # ПОЧТА выглядела бы немного по другому:

```

# ПОЧТА
flush
auth iponly
deny * 192.168.200.4 * 110 * 1-5 18:00:00-23:59:59,00:00:00-08:00:00
allow * 192.168.0.2,192.168.0.3,192.168.0.4 * 25,110 * 1-5 00:00:00-20:00:00
tcppm -i192.168.0.1 25 smtp.provider.ru 25
tcppm -i192.168.0.1 2110 pop.mail.ru 110
tcppm -i192.168.0.1 3110 pop.yandex.ru 110

```

Вы, наверное, заметили новую команду **flush**. Мы ее применили для сброса заданного выше списка доступа, чтобы можно было изменить разрешения доступа для пользователей. Замечу, что сброс не действует на ограничение скорости, которое действует на все службы. И не забываем после его применения задавать способ авторизации.

И наконец, рассмотрим важную составляющую прокси-сервера - WEB-интерфейс администрирования. Для доступа к нему надо запустить команду

```
admin
```

Для доступа к интерфейсу пропишите в адресной строке браузера следующий адрес: <http://192.168.0.1:80>, не забыв, конечно, дать разрешение пользователю на использование стандартного для службы [admin](#) порта 80. В открывшемся окне можно будет получить доступ к информации о максимально допустимом трафике и его текущем значении, посмотреть конфиг целиком и даже перезапустить прокси-сервер. Поэтому разрешения на эту службу раздавайте крайне внимательно! Впрочем, у службы [admin](#) есть ключ `-s`, который не дает делать пользователю ничего недозволенного. Считаю своим долгом упомянуть о команде `writable`, которая, будучи размещена в самом начале конфига, дает возможность не только читать конфиг через WEB-интерфейс, но и менять его! Автор программы предлагает дважды подумать, прежде чем включать эту опцию. Что касается конфига, есть еще одна замечательная команда, мимо которой я не могу пройти. Это команда `monitor`, которая позволяет прокси-серверу следить за файлом конфигурации и в случае обнаружения в ней изменений незамедлительно (по моим наблюдениям, через 2-3 секунды после сохранения) загружать новую конфигурацию. Просто отличная возможность, которой мне так не хватало в ES Proxy!

```
monitor "c:\3proxy\bin\3proxy.cfg"
```

Разумеется, команда `монитор` может работать не только с основным конфигом прокси-сервера. Дело в том, что в конфиг можно включать отдельные текстовые файлы со списками пользователей, сетей, паролей и т.д. Для этого используется макрос `$`. Например, чтобы не записывать в основной конфиг большой список IP адресов компьютеров, имеющих доступ к почте, мы можем создать текстовый файл [IP_list.txt](#), в него внести через запятую (без пробелов!) все нужные нам IP, тогда команда, задающая список доступа к почте и выглядящая так:

```
allow * 192.168.0.2,192.168.0.3,192.168.0.4 * 2110,25,110 * 1-5  
00:00:00-20:00:00
```

может быть заменена командой

```
allow * "$c:\3proxy\IP_list.txt" * 2110,25,110 * 1-5 00:00:00-20:00:00
```

Тогда мы можем задать слежение и за списком адресов из файла [IP_list.txt](#)

Да, чуть не забыли еще две службы - FTP-прокси и SOCKS-прокси, запускаемые командами

```
ftppr  
socks
```

По умолчанию они устанавливаются на порты 21 и 1080 соответственно. Служба [ftppr](#) позволяет запустить "настоящий" FTP-прокси, с поддержкой всего стандартного набора команд протокола FTP, не забудьте в настройках вашего FTP-клиента (в моем случае это FAR) указать прокси-сервер [192.168.0.1](#) на порту 21. А

служба SOCKS позволит вам пользоваться, к примеру, вашей любимой аськой, соответственно в ее настройках надо будет указать службу SOCKS5, сервер **192.168.0.1** и порт **1080**.

Осталось привести получившийся рабочий вариант конфига (версия 6):

```
service
internal 192.168.0.1
external 211.80.11.12
nserver 213.153.100.1
nserver 213.153.101.1
nscache 65536
monitor "c:\3proxy\bin\3proxy.cfg"
monitor "c:\3proxy\IP_list.txt"
logformat "L%C - %U [%d/%o/%Y:%H:%M:%S %z] ""%T"" %E %I %O %N/%R:%r"
log c:\3proxy\logs\3proxy.log D
rotate 30
archiver "c:\Program Files\WinRAR\rar.exe" rar a -df -inul %A %F
nobandlimin * * * 110
bandlimin 24000 * 192.168.0.2,192.168.0.3
bandlimin 48000 * 192.168.0.4
bandlimout 24000 *
counter "c:\3proxy\3proxy.3cf" D "C:\3proxy\traftraf"
nocountin * 192.168.0.4 * 110 *
countin "1/Иванов" M 50 * 192.168.0.2 * * *
countin "2/Петров" M 80 * 192.168.0.3 * * *
countin "3/Сидоров" M 30 * 192.168.0.4 * * *

auth iponly
allow * 192.168.200.4 * * * 1-5 09:00:00-18:00:00
allow * 192.168.0.2,192.168.0.3 * 80,443 * 1-7 00:00:00-23:59:59
proxy

# ПОЧТА
flush
auth iponly
deny * 192.168.200.4 * 110 * 1-5 18:00:00-23:59:59,00:00:00-08:00:00
allow * 192.168.0.2,192.168.0.3,192.168.0.4 * 25,110 * 1-5 00:00:00-20:00:00
tcppm -i192.168.0.1 25 smtp.provider.ru 25
tcppm -i192.168.0.1 2110 pop.mail.ru 110
tcppm -i192.168.0.1 3110 pop.yandex.ru 110
pop3p

flush
auth iponly
allow * 192.168.0.2,192.168.0.3 * 21,1080 * 1-5 00:00:00-20:00:00
ftppr
socks

flush
auth iponly
allow * 192.168.0.4 * 80 * 1-5 00:00:00-20:00:00
admin
```

Здесь мы запускаем прокси-сервер уже не как приложение, а как службу, для этого используется команда `service`. Напоминаю, что установка службы производится командой из командной строки Windows либо любого файл-менеджера:

```
3proxy --install
```

Службы не обязательно должны висеть на стандартных портах, любую из них можно сконфигурировать на другой порт опцией `-pНОМЕР_ПОРТА`, например:

```
proxy -p8080
ftppr -p2121
admin -p8081
```

Хочу прояснить на примерах еще один момент, который я сам понял не сразу и понимание которого, возможно, может вызвать проблемы и у вас. Это использование авторизации `auth strong`. Предположим, что нам хочется, что доступ к WEB-администрированию был доступен только с компьютера `192.168.0.4`, причем даже в этом случае запрашивался пароль доступа. В этом случае соответствующая секция конфига будет выглядеть так:

```
flush
auth strong
allow Administrator 192.168.0.4 * 80 * 1-5 00:00:00-20:00:00
admin
```

И разумеется, прежде чем пользоваться авторизацией `strong`, надо завести пользователя, в нашем случае это `Administrator`:

```
users Administrator:CL:password
```

Теперь при попытке доступа на страницу WEB-администрирования с компьютера Сидорова будет запрашиваться пароль, а на других компьютера доступ будет полностью закрыт. Если же третья строчка в приведенной выше секции конфига будет выглядеть так:

```
allow Administrator * * 80 * 1-5 00:00:00-20:00:00
```

то доступ к администрированию можно будет получить с любого компьютера при предъявлении пароля для пользователя `Administrator`.

Из приведенных примеров становится ясно, что авторизация `strong` предполагает проверку по наиболее полному списку параметров: IP рабочей станции, имя и пароль пользователя, список разрешенных хостов, портов и т.д., в отличие от авторизации `iponly`, когда имя пользователя не проверяется - но звездочку вместо имени пользователя в команде `allow` мы все же вынуждены ставить для совместимости!

Для удобства дальнейшего пользования конфиг можно разбить на логические разделы и пополнить комментариями. Я это делаю с помощью символа `#`. Напомню, что любая строка конфига, начинающаяся с пробела или с символа `#`,

воспринимается сервером как комментарий. В нашем случае окончательно "вылизанный" конфиг может выглядеть так (версия 7):


```
#####
# НАЧАЛО КОНФИГА
#####

service
internal 192.168.0.1
external 211.80.11.12
nserver 213.153.100.1
nserver 213.153.101.1
nscache 65536
monitor "c:\3proxy\bin\3proxy.cfg"
monitor "c:\3proxy\IP_list.txt"
logformat "L%C - %U [%d/%o/%Y:%H:%M:%S %z] ""%T"" %E %I %O %N/%R:%r"
log c:\3proxy\logs3\proxy.log D
rotate 30
archiver "c:\Program Files\WinRAR\rar.exe" rar a -df -inul %A %F

#####
# ЗАДАЕМ ПОЛЬЗОВАТЕЛЕЙ, ОГРАНИЧЕНИЯ СКОРОСТИ И СЧЕТЧИКИ
#####

users Administrator:CL:password
nobandlimin * * * 110
bandlimin 24000 * 192.168.0.2,192.168.0.3
bandlimin 48000 * 192.168.0.4
bandlimout 24000 *
counter "c:\3proxy\3proxy.3cf" D "C:\3proxy\traftraf"
nocountin * 192.168.0.4 * 110 *
countin "1/Иванов" M 50 * 192.168.0.2 * * *
countin "2/Петров" M 80 * 192.168.0.3 * * *
countin "3/Сидоров" M 30 * 192.168.0.4 * * *

#####
# РАЗДАЕМ WEB
#####

auth iponly
allow * 192.168.0.4 * * * 1-5 09:00:00-18:00:00
allow * 192.168.0.2,192.168.0.3 * 80,443 * 1-7 00:00:00-23:59:59
proxy

#####
# РАЗДАЕМ ПОЧТУ
#####

flush
auth iponly
deny * 192.168.200.4 * 110 * 1-5 18:00:00-23:59:59,00:00:00-08:00:00
allow * "$c:\3proxy\IP_list.txt" * 3110,25,110 * 1-5 00:00:00-20:00:00
tcppm -i192.168.0.1 25 smtp.provider.ru 25
tcppm -i192.168.0.1 2110 pop.mail.ru 110
tcppm -i192.168.0.1 3110 pop.yandex.ru 110

#####
# РАЗДАЕМ FTP И АСЬКУ
```

```
#####

flush
auth iponly
allow * 192.168.0.2,192.168.0.3 * 2110,25,110 * 1-5 00:00:00-20:00:00
ftppr
socks

#####
# АДМИНИСТРАТИВНЫЙ ДОСТУП
#####

flush
auth iponly
allow Administrator 192.168.0.4 * 80 * 1-5 00:00:00-20:00:00
admin

end

#####
# КОНЕЦ КОНФИГА
#####
```

Согласитесь, что так конфиг намного информативнее и легче в восприятии.

В конце поставили необязательную команду end. Все, что находится после нее, сервером не обрабатывается. Команда end - это своего рода знак глобального, до конца конфига, комментария. Бывает иногда удобно отключить большую часть конфига, просто поставив перед ней эту команду.

Необходимое послесловие

Приведенный выше конфиг довольно близок к тому, что я использую в офисе, за исключением количества пользователей и некоторых специфичных моментов.

Должен признаться, что мы не рассмотрели и трети всех возможных настроек сервера. Есть еще много разных команд, которые вам могут понадобиться - *parent*, *nsrecord*, *fakeresolve*, *dialer*, *maxconn*, *system*, *include*, которые остались не рассмотренными вообще, да и разнообразие всевозможных сочетаний рассмотренных команд намного богаче. Но я и не ставил себе задачу дать более-менее полное описание конфига, ограничившись более скромной - рассмотреть на ряде примеров, построенных как матрешка - от самого маленького и простого до достаточно сложного и "правильного" - конфиг, чтобы на их основе понять основные принципы. Несмотря на кажущуюся на первый взгляд сложность логики конфига, надеюсь, в процессе чтения этого текста вы уже убедились, что в нем все довольно строго и логично. Опять же автор программы обещает к релизу сделать конфиг более читаемым и понятным.

Важно понять, как работает обработчик конфига. Очень просто - в порядке следования команд в конфиге. Если серверу надо определить, дать ли разрешение на тот или иной запрос - он ищет первое удовлетворяющее правило из списка доступа клиентов (ACL) и действует в соответствии с ним, даже если после него есть другие правила. И еще одно - все разрешения должны быть заданы в конфиге ДО запуска соответствующей службы.