TORифицируем FreeBSD

Впервые, когда я столкнулся с анонимной сетью Tor (2007 год), то потратил не один день, чтобы настроить ее и добиться максимально возможной анонимности в Интернет. Существует огромное количество статей, обзоров, инструкций по настройке Tor'a, но все они разбросаны в разных уголках сети. Хранить все эти чужие обрывки мыслей в конце концов надоело и было принято решение написать статью, основанную на личном опыте использования и анализа ошибок, которые возникали в ходе экспериментов.

Я не претендую на новизну предоставленной информации, а всего лишь хочу собрать ее как можно больше и в одном месте. Окончательной версии этого материала не существует, и он, по мере возможности, будет дополняется различными советами и рекомендациями. Вся информация, изложенная ниже, была проверена на ОС FreeBSD 8.0, но так же может быть использована на любой другой ОС с незначительными изменениями. Все настройки, приведенные ниже, - это результат, который, в какой-то мере, меня устраивает. Поэтому если кому-то захочется использовать вместо Privoxy какой-то другой web-фильтр, фаервол, браузер, etc — это только приветствуется, так как каждый человек те или иные задачи решает по разному.

Установка и Настройка

Для работы в анонимной сети нам понадобится непосредственно сам Tor и Privoxy - web-прокси с расширенными возможностями фильтрования запросов, обеспечивающий конфиденциальность, модифицирующий содержимое web-страниц, осуществляющий управление закладками cookie, контролирующий доступ к web-контенту через прокси и удаляющий рекламу, баннеры, всплывающие окна, etc.

Заходим как root и выполняем команды:

```
#cd /ust/ports/security/tor
#make install clean
#cd ../../www/privoxy
#make install clean
#tor —version
…
```

Tor version 0.2.1.24
…

После завершения установки (в принципе проблем с ней возникнуть не должно), переходим к настройке.

Вообще по умолчанию конфиг Tor'a пригоден для работы, если конечно пользователь не находится за http(s)-прокси или не ограничен фаерволом. Для конкретного случая, естественно, придётся повозиться. Я добавил всего лишь один параметр который принудительно заставляет tor работать в качестве клиента, так как в случае, когда он обнаруживает что стоит на машине с широким каналом, то оптимизирует своё поведение к организации middleman-сервера, поэтому Tor будет работать как клиент чуть с превышенными запросами. Как я понял, это уменьшает количество обращений к корневым серверам и экономит трафик.

Редактируем конфигурационный файл tor'a:

#cd /usr/local/etc/tor/
#mv torrc.sample torrc
#chmod +w torrc
#echo «ClientOnly 1»>>torrc

Так же есть возможность настройки выхода в сеть через http(s)-прокси, выстраивать цепочки входа/выхода в сеть через посредников (proxy, socks, vpn, другие анонимные сети), регулировать временной период для синхронизации с корневыми серверами, настраивать политики ограничения, скрытые сервисы, Tor-сервер и многое другое.

Для любителей GUI существуют графические оболочки Tork и Vidalia. Последняя устанавливается по умолчанию и находится в /usr/local/bin/vidalia. Клиент Tor'a работает с правами пользователя _tor и, что бы запустить графический интерфейс для него с правами другого пользователя, необходимо отредактировать /usr/local/etc/tor/torrc. Раскомментируйте строчки:

ControlPort 9051– на этом порту Tor будет принимать подключения для управления Tor-сервером, то еесть можно подключиться удаленно для конфигурации Tor'a. В первую очередь опция важна для тех, кто использует графические оболочки;

HashedControlPassword - хеш пароля для доступа и конфигурации Tor-сервера.

Для генерации хэш значения необходимо выполнить команду:

#tor —hash-password My_P4s5w0rD

```
toshiba# tor --hash-password My-P4s5w0rD
Apr 03 11:31:45.781 [notice] Tor v0.2.1.24. This is experimental software. Do no
t rely on it for strong anonymity. (Running on FreeBSD i386)
Apr 03 11:31:45.782 [warn] You are running Tor as root. You don't need to, and y
ou probably shouldn't.
16:7ACD74C20677E9DF60AAF1DD4E66C9BD718CE885695D503FE8D3122AE7
```

Далее этот хэш надо вставить напротив последней раскомментированной строки, сохранить изменения и выполнить команду kill -HUP pid процесса (перечитать конфигурационный файл) либо перезагрузить систему. Если честно, я не вижу смысла в графической оболочке для Tor'a, так как тонко сконфигурировать его через неё всё равно не получится.

Теперь переходим к конфигурации web-фильтра:

#privoxy —version
Privoxy version 3.0.16
#cd /usr/local/etc/privoxy/
#chmod +w config
#vi config

Вообще этот дефолтовый конфиг с незначительными изменениями так же подходит, но всё же на некоторых моментах я заострю внимание.

Forward-socks4a / 127.0.0.1:9050 . - обязательный параметр, который необходимо прописать.

Эта строка определяет, что перенаправление необходимо применять ко всем адресам (об этом говорит символ '/'), при этом используя в качестве socks-прокси tor(127.0.0.1:9050). Версия 4a выбрана не случайно, так как версия именно этого протокола позволяет резолвить DNS имена через прокси. Точка в конце означает что запросы не будут перенаправляться к другому http-прокси. Необходимо проследить, что бы в конфиге privoxy строки, отвечающие за перенаправление (forward) были все закомментированы, кроме вышеописанной.

Listen-address 127.0.0.1:8118 (по умолчанию) — принимать клиентские запросы «адрес:порт».

Хочу лишь отметить, если опустить ip-адрес, то privoxy будет «слушать» на всех интерфейсах, что существенно снижает общую безопасность ОС.

Toggle 1 (по умолчанию) - если отключить (установить в ноль; смотрите enable-remote-toggle), то Privoxy будет работать как нейтральный прокси сервер, то есть блокировка рекламы, фильтрация web-страниц, etc работать не будут. При внесении каких-либо изменений в фильтры полезно последующие тестирование и сравнение результатов «до» и «после».

Enable-remote-toggle 0 (по умолчанию) — позволяет включать/отключать privoxy через web-браузер. В последних версиях эта возможность отключена. Я считаю ее лишней.

Enable-edit-actions 0 (по умолчанию) — позволяет редактировать файлы действий (о них позже) через web-браузер. Не знаю кому как, а мне удобней в vi редактировать с последующим перечитыванием конфигурационного файла (kill -HUP pid процесса). Как и предыдущий параметр, он отключён по умолчанию.
Выше я привёл наиболее важные, на мой взгляд, опции. Конечно, не исключено, что кому-то понадобится настроить списки ACL, списки доверенных хостов, размер буфера для фильтрации содержимого, и т.п.

Теперь переходим к action-файлам. Они указывают privoxy как фильтровать поступающий web-контент: блокировать, отфильтровать, изменить. Нас, прежде всего, будут интересовать действия hide-*, с помощью которых можно подделать http-заголовок. Он передаёт на сервер

информацию, которая снижает общую анонимность, что крайне нежелательно.

#vi /usr/local/etc/privoxy/match-all.action

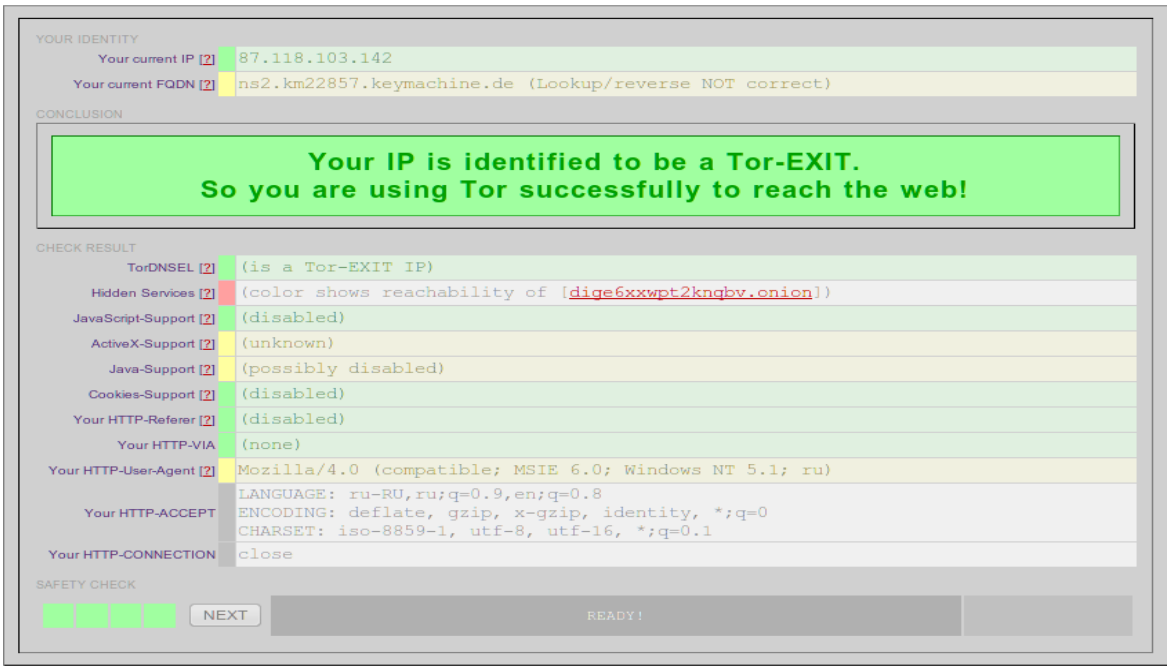
hide-referrer{forge} — скрывает информацию о том, по какой ссылке пользователь перешёл на сайт. В качестве значения, указанного в фигурных скобках, можно написать любое, будь то {http://www.google.com/} или {my-security-referer}. Так же можно вообще вырезать этот заголовок (параметр {block}), но, на мой взгляд, это снижает анонимность, так как идентифицировать такого пользователя в пределах хотя бы одного сайта не так уж сложно. Значение forge представляет все таким образом, будто пользователь перешёл с главной страницы сервера, которому направлялся запрос. Разработчики рекомендуют последний параметр. Это связано с web-сервером, который может на некорректный referer «отдать» не полный контент (изображения, баннеры). На мой взгляд, это не так страшно, и даже в некоторых ситуациях повышает степень анонимности. Например, можно сёрфить по сайту и перейти на такую страницу, на которую с главной страницы сайта перейти невозможно, сразу станет ясно что пользователь подделал http-заголовок. Хотя если поставить какой-нибудь поисковик, то, например, с главной страницы гугла перейти на сайт www.exampl.com так же не реально, поэтому я решил «прислушаться» к мнению разработчиков.

hide-user-agent{Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;)} - заменяет значение заголовка User-Agent на заданное. В принципе можно написать все, что душе угодно, но опять же, лучше маскироваться под общий поток пользователей, которые в основном используют ОС Windows XP, SP2(3), Vista, Seven и браузер IE6(7,8), Opera, Firefox. Данное значение выбрано не случайно, почему это так, объясняется ниже в разделе «Тонкая настройка браузера».

hide-accept-language{ru-RU,ru;q=0.9,en;q=0.8} — заменяет значение заголовка HTTP-ACCEPT на заданное. В этом заголовке три поля: LANGUAGE, ENCODING, CHARSET. Этот параметр относится к первому. Последнее можно подделать средствами браузера (об этом позже).

hide-from-header{block} — блокирует передачу электронного почтового адреса web-серверу. Вместо block можно написать любое значение, но этого делать не стоит по понятным причинам.

change-x-forwarded-for{block} - удаляет любые существующие заголовки HTTP, вида "X-Forwarded-for:" из клиентских запросов и не позволяет добавлять новые. В ранних версиях программы он назывался hide-forwarded-for-headers и в качестве своего значения ни чего не принимал.



Вообще Privoxy - очень функциональное приложение. Есть возможность создавать свои фильтры, применить их ко всем или определённым сайтам, контролировать поведение cookie, всплывающих окон, баннеров, создавать доверенные зоны и многое другое.

Важно отметить, не смотря на то, что эта программа работает с протоколом https, она не может фильтровать его пакеты, так как трафик уже поступает в зашифрованном виде. Это несет в себе угрозу снижения степени анонимности и сводит на нет все предыдущие усилия, поэтому необходимо быть аккуратными при посещении сайтов, обеспечивающих защищенное соединение.

Можно, конечно, воспользоваться другими web-фильтрами, которые умеют обрабатывать зашифрованный web-контент или воспользоваться встроенными средствами браузера(минусы этого способа в разделе «Тонкая настройка браузера»). В качестве альтернативы Privoxy можно воспользоваться Proxomitron'ом. Большой минус последней, заключается в том, что его разработка завершилась в 2003 году. Можно обойтись вообще без web-фильтра, если использовать для сёрфинга браузер Mozilla Firefox с его многочисленными плагинами, многие из которых, кстати, написаны специально для анонимного серфинга Интенет через Tor. Один из таких плагинов называется Torbutton и позволяет делать то, чего не умеет Privoxy. Также можно воспользоваться уже настроенным пакетом типа TorBrowser, в который входят Firefox с набором плагинов и polipo. Он не требует настройки и после распаковки можно сразу приступать к работе. Возможно, читатель проявит инициативу и сам соберет нужный ему софт. Я лишь привёл примеры альтернативных решений, которыми можно воспользоваться.

Ещё одна не приятная ситуация, на которую стоит обратить внимание, заключается в резолве DNS имён. TOR работает на транспортном уровне с TCP, а DNS в большенстве случаев работает с UDP, поэтому может случиться так, что трафик идёт через tor, а запросы DNS в обход. Это сильный удар по анонимности, которого стоит избегать. Не случайно разработчики выбрали протокол socks4a, который умеет транслировать запросы DNS через себя. Хотя socks5 тоже

умеет это делать, он устроен таким образом, что трансляция этих запросов не обязательна, а в версии 4a она происходит принудительно. Не надо путать протоколы socks4 и socks4a: первый вообще не умеет транслировать запросы DNS, поэтому его в связке с tor использовать не рекомендуется.

Решений проблем с DNS несколько. Можно воспользоваться собственным DNS-сервером или же обращаться к ресурсам Интернет по их сетевому адресу. Последний вариант не удобен для повседневной деятельности пользователя, а первый сводится к установке и настройке собственного демона, что так же требует определённых знаний. Существует ещё один способ который не требует особых усилий. Он заключается в редактировании файла /etc/resolv.conf в котором обычно хранятся записи о серверах имён провайдера. Необходимо закомментировать все строки, которые там присутствуют, и добавить фиктивный DNS-сервер. В качестве такого сервера может выступать 127.0.0.1 (localhost) и резолв происходит где-то в сети Tor (на exit-nod'e) без участия провайдера и утечек информации. На первый взгляд этот способ выглядит вполне работоспособным но (!) существует опасность «нарваться» на DNS-spoofing, поэтому необходимо быть предельно внимательным. В таких ситуациях я бы порекомендовал все ресурсы, которые имеют для пользователя определённую ценность внести в файл /etc/hosts. Это защитит от атак подобного рода, т.к по умолчанию ОС обращается при резолве имени сначала в файл hosts, а уже потом к DNS-серверам. Возникает сразу вопрос: а как же безопасно преобразовать имя понятное человеку в ip-адрес? Очень просто. В состав tor'a входит утилита под названием tor-resolve которая в качестве параметра принимает имя ресурса, а возвращает сетевой адрес.

```
toshiba# tor-resolve www.defec.ru
95.131.29.1
```

Итак, от одной беды защитились (хотя способ не претендует на изящность), а другая подкралась незаметно. Дело в том, что трафик, идущий от пользователя до самого последнего сервера (exit-nod) в сети Tor идёт зашифрованный. На exit-node он расшифровывается и идёт уже непосредственно к серверу, к которому обращается пользователь. И тут пользователь может подвергнуться MitM-атаке. Это атака - бич анонимной сети tor и защититься от неё очень и очень сложно. Приходиться либо надеяться на порядочность владельца exit-node’a, либо использовать протокол ssl или другие протоколы защищающие от подобных атак. При этом всё равно остаётся возможность анализа трафика, но уже с большими сложностями. Точнее перехватить трафик всё так же легко, а вот модифицировать или подменить информацию уже сложнее. Одна из таких атак осуществляется с помощью утилиты sslstrip. Суть этой атаки заключается в том, она позволяет перехватить SSL-соединений, основанную на том факте, что, как правило, перед началом взаимодействия по https и установкой SSL-соединения пользователи посещают некоторую обычную веб-страницу с помощью незащищенного http-соединения, где и нажимают заветную кнопку авторизации после заполнения необходимых полей. Защиты как таковой нет, т.к многие сайты практикуют редирект http=>https=>http, а если сервер поддерживает полностью защищенную версию сайта, то нужно заходить исключительно по https (с полным набором адреса https://адрес) и, конечно же, быть предельно внимательными и следить за сообщениями браузера при работе по защищенному протоколу. Есть возможность вычислить exit-node'ы, которые используют sslstrip с помощью утилиты torscanner которая входит в пакет tortunnel. В качестве параметра она принимает URL-адрес и соединяется со всеми выходящими узлами сети. Таким образом можно проследить, на каких узлах был переход с https на http. Далее внести эти узлы в качестве значения параметра ExcludeNodes и tor вовсе исключит их при построении цепочек нодов. Идея хорошая, но недостаток очевиден: атакующий может

появиться внезапно, перехватить трафик и исчезнуть.

У внимательного читателя возникнет вопрос: а что делать с программами, которые работают по TCP, но опционально не поддерживают прокси? Опять же, решений этой проблемы несколько. Можно воспользоваться утилитой torify, которая поставляется вместе с пакетом tor. Работать с ней просто:

#cd /usr/local/etc/tor
#mv tor- tsocks.conf.sample tor-tsocks.conf
#torify <программа>[аргументы]

```
toshiba# tor-resolve defec.ru
95.131.29.1
toshiba# torify telnet 95.131.29.1 80
Trying 95.131.29.1...
Connected to 95.131.29.1.
Escape character is '^]'.
```

В качестве аналога torify хочу предложить читателям proxychains. По умолчанию она уже настроена для работы с tor и не требует редактирования конфигурационного файла. Так же в последних версиях появилась проксификация DNS-запросов. Есть возможность выстраивать цепочки проски до входа в анонимную сеть. Пользоваться ею так же просто, как и вышеописанной утилитой. Процесс установки описывать не буду, в силу его простоты. Примечание: в отличие от proxychains, torify не умеет резолвить адреса, поэтому её необходимо использовать в связке с tor-resolve.

```
toshiba# proxychains telnet defec.ru 80
ProxyChains-3.1 (http://proxychains.sf.net)
|DNS-request| defec.ru
|S-chain|-<>-127.0.0.1:9050-<><>-4.2.2.2:53-<><>-OK
|DNS-response| defec.ru is 95.131.29.1
Trying 95.131.29.1...
|S-chain|-<>-127.0.0.1:9050-<><>-95.131.29.1:80-<><>-OK
Connected to 95.131.29.1.
Escape character is '^]'.
```

Чтобы tor и Privoxy запускались во время загрузки ОС как демоны, необходимо отредактировать /etc/rc.conf:

#vi /etc/rc.conf
tor_enable=YES
privoxy_enable=YES

Ещё необходимо проверить, что бы на rc-скриптах этих программ стоял бит выполнения:

#cd /usr/local/etc/rc.d/
#ls -la tor privoxy

```
toshiba# cd /usr/local/etc/rc.d/
toshiba# ls -la tor privoxy
-r-xr--r--  1 privoxy  privoxy  2104 14 мар 16:28 privoxy
-r-xr--r--  1 _tor     _tor     1261 27 мар 18:07 tor
```

Иначе выполняем chmod u+x tor privoxy.
Теперь необходимо перезагрузиться и посмотреть, что всё нормально запускается и работает без ошибок. Далее надо убедиться, что не запущено ничего лишнего, выполнив команду sockstat -46.

Настройка фаервола

В современной сети Интернет без средств защиты компьютера «обитать» стало опасно. Поэтому фаерволл уже стал постоянным атрибутом защиты компьютера, но его настройка в нашем случае явилась ещё и дополнительной надстройкой для Tor'a. Необходимо перестраховаться от нежелательных DNS-запросов в обход анонимной сети или программ, которые не настроены на работу с ней.

Итак, в качестве брандмауэра я выбрал ipfw. Считаю,что фаервол должен быть ядерный, это даёт дополнительную ступеньку защиты ОС, поэтому я пересобрал ядро с включёнными опциями ipfw. Для тестов или временных манипуляций с анонимностью вполне подойдёт и модуль, который можно загрузить с помощью kldload.

При работе клиента, необходимо получать информацию, которая даёт возможность для построения цепочек. По умолчанию сервера tor работают на портах 9001 и 9030, но разработчики рекомендуют использовать порты 80 и 443 (если не заняты) настраивать сервер на них. Это связано с политиками безопасности корпоративных сетей, в которых разрешён лишь необходимый трафик и необходимо для того, чтобы пользователи, которые ограничены ими, могли так же пользоваться средствами анонимности. Из этого следует вывод, что необходимо разрешить исходящий/входящий трафик на эти порты:

ipfw add allow tcp from ip-адрес to any 80,443,8080,9001,9030 via сетевой интерфес
add allow tcp from any 80,443,8080,9001,9030 to ip-адрес via сетевой интерфейс

В принципе этого достаточно, что бы клиент заработал, но(!) необходимо следить, чтобы приложения, работающие с этими портами(80,443), были настроены на работу с Tor. Дополнительная преграда уже выставлена ранее (фиктивный DNS сервер плюс запрет на исходящие/входящие на DNS), поэтому опасаться не надо, если вдруг пользователь ненастроенным браузером попытаться зайти на сайт.

Правила, разрешающие обращаться с 127.0.0.1 Tor и Privoxy не нужны, т.к по умолчанию будет выполняться /etc/rc.firewall в котором будут реализованы следующие правила:

ipfw add 100 pass all from any to any via lo0
ipfw add 200 deny all from any to 127.0.0.0/8
ipfw add 300 deny ip from 127.0.0.0/8 to any

Их назначение — пропускать весь локальный трафик на кольцевом интерфейсе и предотвращать попытки обращения к внутренним адресам машины из внешнего мира. Если этих правил не будет, перестанет работать множество внутренних сервисов — RPC, X11,etc.

Тонкая настройка браузера

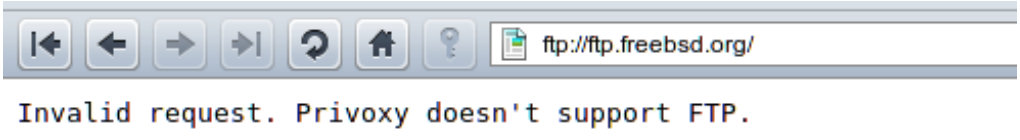
Я пользуюсь web-браузером Opera 10.10. На момент написания статьи эта была последняя доступная версия из портов. У каждого пользователя свои предпочтения по части выбора инструмента серфинга по Сети, поэтому если кто-то из читателей пользуется другим браузером - в этом нет ничего страшного. Разработчики рекомендуют использовать Mozilla Firefox, тем более для него написано множество плагинов, которые повышают анонимность.

Сразу хочу отметить, что для достижения приемлемого уровня анонимности придётся отказаться от Java, JavaScript, Flash, ActiveX (для пользователей Windows), различных плагинов, etc, так как они несут угрозу раскрытия анонимности. Тонко настроить браузер через графическое меню не представляется возможным. Многие настройки скрыты от глаз пользователя, поэтому придётся настраивать через about:config(opera:config). Ниже я приведу наиболее важные, на мой взгляд, настройки.

В адресной строке набираем about:config:

Блок [Cache]
Cache HTTPS After Sessions=0(выкл)-при выходе сохранять HTTPS-странице в кэше.
Блок [Clear Private Data Dialog]
CheckFlags=1023 — запомнить для последующего использования:cookie,историю посещения страниц,кэш,etc(Инструменты=>Удалить личные данные)
Блок [Disk Cache]
Empty On Exit=1-очищать кэш при выходе из Opera
Блок [Extensions]
Ask Flash Download=0-запрос о загрузке Flash
Plugins=0-включить плагины
Scripting=0-Включить JavaScript/ECMAScript
Блок [Java]
Enabled=0-разрешить Java на web-страницах
Блок [Multimedia]
Play Background Sound=0-Если на рисунке есть звуки, то проигрывать их
Блок [Network]
В этом блоке возможны так же настройки HTTP Accept ,HTTP Accept Charset,HTTP Accept Language
Блок [Personal info]
Если что то есть, всё удаляем
Блок [Proxy]
HTTP Server,HTTPS Server,FTP Server,Gopher ServerбWAIS Server=127.0.0.1:8118
Use FTP,Use GOPHER,Use HTTP,Use HTTPS,Use WAIS=1(ставим галочки)
No Proxy Servers=127.0.0.1

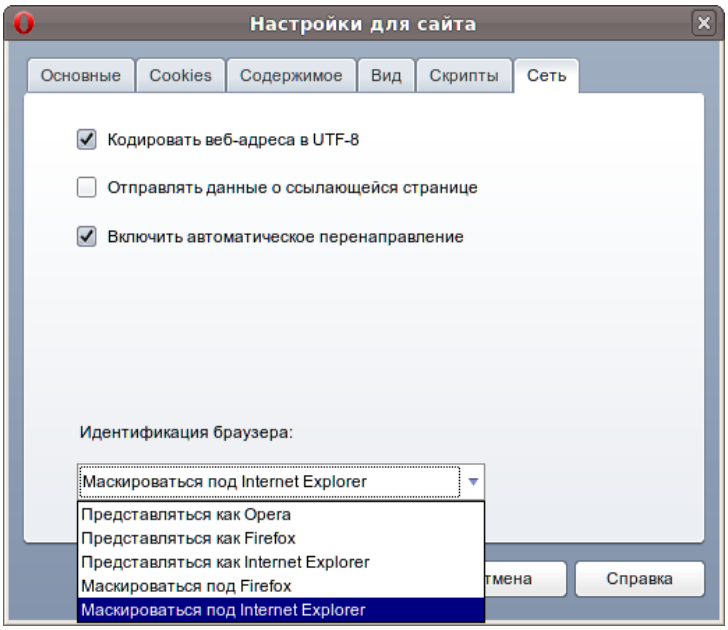
Примечание: хотя Privoxy не поддерживает FTP, все же следует настроить браузер на работу через прокси для этого протокола, и при попытке набрать ftp:// URL-адрес, браузер отобразит ошибку, но информация не будет послана в сеть.



Блок [SecurityPrefs]
Password Lifetime=1-Время хранения мастер-пароля в памяти, в минутах.
Блок [Special]
JavaScript AppCodeName,JavaScript AppName,JavaScript IE AppName,JavaScript Opera AppName-какое имя должно быть найдено свойством DOM/JS, которое отыскивает кодовое название браузера; я выставил везде «Microsoft Internet Explorer»
Блок [User Agent]
Spoof UserAgent ID=5 - если 5,то идентифицируется как Internet Explorer, и не упоминает об Opera
Блок [User Prefs]
Accept Cookies Session Only - удалять Cookie при выходе(см. Enable Cookie)
Allow script to change status =0Разрешить JavaScript изменять содержимое строк состояния Opera
Allow script to hide address =0 Разрешить JavaScript открывать вкладки без полного поля адреса
Allow script to lower window =0 Разрешить JavaScript сворачивать страницы
Allow script to move window=0 Разрешить JavaScript менять положение страниц
Allow script to raise window=0 Разрешить JavaScript активировать свернутые вкладки или окна
Allow script to receive right clicks=0 Разрешить JavaScript обрабатывать щелчки правой кнопки
Allow script to resize window=0 Разрешить JavaScript изменять размеры страниц и окон
Примечание: По умолчанию в браузере будет отключён JavaScript, но многие сайты без него работать не будут и для доверенных всё же придётся его включать (Инструменты=>Быстрые настройки=>Настройки для сайта или F12=>Настройки для сайта), поэтому пока что отключаем всё.
Check For New Opera=0-Сохранять запись о еженедельной проверке новых обновлений
Client Pull=0-Разрешить перенаправление HTTP
Client Refresh=0 Разрешить автоматическое перенаправление
Enable Cookies=0-Отклонять все Cookie
Примечание: По умолчанию cookie будут отключены, но работа с большинством сайтов из-за этого будет невозможна, поэтому для доверенных всё же придётся их включать (Инструменты=>Быстрые настройки=>Настройки для сайта или F12=>Настройки для сайта)
Enable Referrer=0-Разрешить логирование рефереров; разрешить веб-сайтам знать с какого сайта пришёл пользователь.
Enable Wand=0-Включить менеджер паролей
Global History File-История Opera записывается в этот файл(стираем путевое имя)
Home URL=http://torcheck.xenobite.eu-домашняя страница. Советую поставить потому что, перед началом работы в Интернете, после запуска браузера, вначале стоит убедиться в своей анонимности.
Max Direct History Lines=0 Максимальное число набранных URL
Max Global History Lines=0 Максимальное число записей в общей истории
Save Password Protected Pages=0-Разрешить сохранение защищенных паролем страниц в сеансе
Script Spoof=5-идентефицироваться в сценариях как Microsoft Internet Explorer.
User JavaScript =0 Включить или выключить использование UserJS.
User JavaScript on HTTPS =0 Разрешить UserJS на защищенных серверах.
Warn Insecure Form=1-Предупредить перед опубликованием данных формы на небезопасных страницах.

Вернёмся к теме про модификацию заголовка User Agent. Как видно из выше перечисленных настроек, есть возможность встроенными средствами программы изменить некоторую информацию, посылаемую серверу web-браузером. Например «Http Accept Charset», Privoxy не умеет изменять, а вот с помощью браузера это предоставляется возможным. Как уже известно,

Privoxy не умеет фильтровать защищённый трафик, поэтому ничего не остаётся кроме как воспользоваться встроенными средствами. Переходим в меню Инструменты =>Настройки =>Дополнительно=>Содержимое=>Настройки для сайтов=>Добавить. В появившемся окне можно настроить для конкретного сайта cookie, javascript, всплывающие окна, etc. Сложностей в конфигурации возникнуть не должно и настройки задаваемые пользователем будут действовать только в пределах одного конкретного сайта. Хочу остановиться лишь на вкладке «Сеть» и выборе идентификации браузера. Так как изначально в моём примере я представлялся как «Microsoft Internet Explorer», то выбрал из списка «Маскироваться под Internet Explorer». Значения «Представляться как ...» оставляют некоторую информацию о настоящем браузере.



Вроде бы всё настроено и работает, но уже есть списки заголовков, по которым можно определить настоящий браузер и некоторые «умные» скрипты проверки на анонимность используют их. По моим наблюдениям, Opera с выставленным значением «Маскироваться под Internet Explorer», идентифицируется такими скриптами как MSIE6 (так же выводится название ОС) с дополнительным полем «Браузер модифицирован» и тегами, которые, по его мнению, были изменены. В моём случае это теги ОС и язык (ru). «Это не есть хорошо» - подумал я и решил поэксперементировать. В результате этих эксперементов всё таки нашёлся способ запутать «умный» тест на анонимность.

Исходная строка, определяющая браузер, маскирующийся под IE средствами Opera, выглядит так:

Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;(тут пробел)ru

Браузер
MSIE v 6.0
Браузер модифицирован
да
Тэги модификации
ru
OS
Windows XP

Удалив тег модификации «ru» и, вставив данную строку в качестве значения параметра hide-user-agent в файл действий web-фильтра, получил ответ, что мой браузер не модифицирован и является настоящим.

СОБРАННАЯ ИНФОРМАЦИЯ
Переданный адрес
173.14.249.26
Браузер
MSIE v 6.0
OS
Windows XP
Найденный адрес
173.14.249.26
Имя хоста
173-14-249-26-washington.hfc.comcastbusiness.net
Почтовый сервер
не найден

Хорошо, но не совсем по той причине, что защищённый трафик Privoxy фильтровать не умеет, и с первого взгляда видно, что строка идентифицирующая браузер по http, отличается от той, что получена по https.

По http:

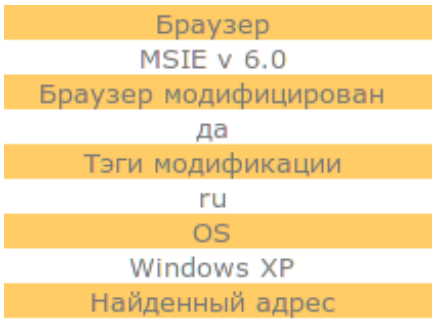
Your HTTP-User-Agent [?]	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;)

по https:

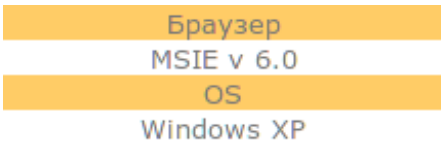
Your HTTP-User-Agent [?]	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; ru)

Согласитесь, не очень «красивая» ситуация, хоть и не сильно заметно. Поигравшись ещё немного с этим тэгом, я всё таки нашёл способ, который, на первый взгляд, ничем не отличается от исходной строки, но всё таки сбивает с толку скрипт проверки. Между символом “;” и тегом модификации “ru” стоит пробел, если его удалить, то браузер и ОС определяются как немодифицированные.

До модификации(Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;<тут пробел>ru):



После модификации(Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;ru):

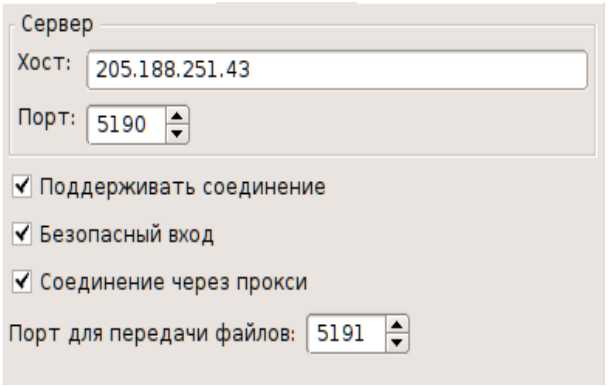


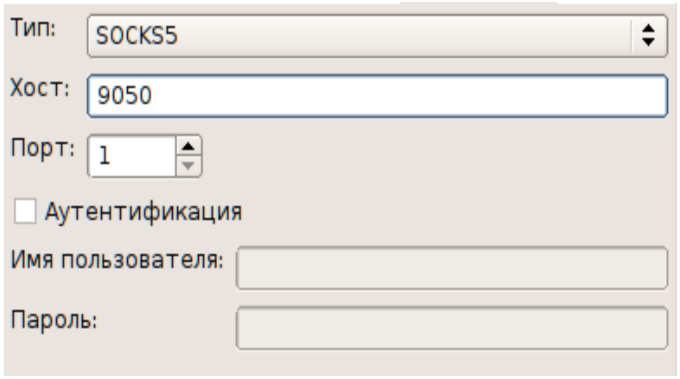
Получается что, абсолютно две одинаковых, на первый взгляд, строки, интерпретируются в зависимости от протокола(http/https) по-разному. К сожалению, это лишь создаёт иллюзию, что скрипт проверки «работает неправильно». Минусы использования встроенных средств браузера очевидны, к тому же для каждого https сайта придется производить настройки в «...=>Содержимое=>Настройки сайтов». У читателя может возникнуть вопрос: зачем так «заморачиваться»? Проще в файле действий выставить что-нибудь «пораспространённей» на данный момент, а с https ситуация и без этого трудноразрешимая, пусть остаётся всё как есть. Моё мнение на этот счёт такое: оставить «как есть» не вариант, тем более, когда имеется возможность сделать чуть-чуть лучше. К тому же я пользовался всего лишь одним тестом на анонимность, работающим по https, и в этом случае он идентифицировал браузер «как надо» без манипуляций, описанных выше (маскировка была произведена средствами браузера).Это несомненно радует, но если атакующий осведомлён в этих мелочах, то этот вариант «работать» не будет.

Программы сетевого общения

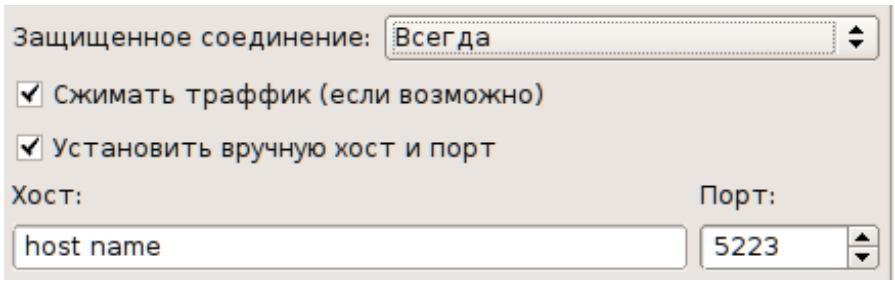
Для ICQ и Jabber я использую qutIM (версия 0.2) - бесплатный open-source многопротокольный (ICQ, Jabber/GTalk/Ya.Online/LiveJournal.com, Mail.Ru, IRC) клиент для общения. Сразу хочу отметить, что придется отказаться от различных плагинов показа погоды и т.п. дополнений. Для того, чтобы настроиться на работу с Tor'ом необходимо сконфигурировать для каждого протокола, соединение через прокси сервер:
Настройки=>Учётные записи=>Выбираем протокол=>Редактировать=>Сеть ставим галочку «Соединение через прокси».

Для тех, кто сомневается в утечке DNS запросов, в поле «Хост» можно прописать ip-адрес сервера. Безопасно преобразовать DNS имя, поможет утилита tor-resolve. На вкладке «Прокси» выбираем «Тип»=>HTTP/SOCKS5, «Хост»=>127.0.0.1, «Порт»=>8118/9050(в зависимости от типа выбранного прокси).





Для Jabber'a все почти аналогично, только на вкладке «Ресурс» следует вписать нужный ресурс. На вкладке «Соединение» выставить из выпадающего списка «Защищёное соединение»=>Всегда, поставить галочку «Установить в ручную хост и порт», в поле «Хост» вписать адрес сервера и 5223(SSL) порт.



Есть вариант настроить соединение через прокси глобально (Настройки=>Глобальный прокси), но предпочтительней, на мой взгляд, первый вариант, т.к есть возможность выставить вместо DNS имени сетевой адрес и, в случае с Jabber, принудительно указать сервер и порт.

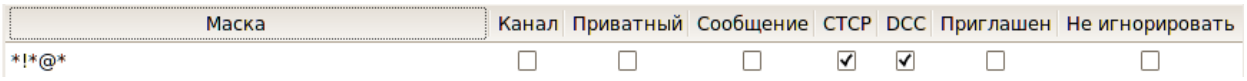
Примечание: пользоваться ICQ через Tor небезопасно, т.к с большой вероятностью (особенно если красивый UIN) номер может быть угнан. Предпочтительней пользоваться ICQ через Jabber с помощью транспортов, т.к трафик идёт по SSL и расшифровывается непосредственно на сервере предоставляющий транспорт, но(!) опять же существует опасность потерять UIN. На эту тему читал много материалов и пришёл к выводу, что трафик, идущий напрямую через Tor=>ICQ, рассматривать не стоит вообще, в связи с участившимися случаями недобросовестности владельцев exit-node'ов. Второй способ конечно же лучше, но опять же возникает вопрос: а можно ли доверять владельцу Jabber-сервера? Конечно же, перед тем как использовать транспорт, необходимо найти о нём отзывы пользователей, желательно на нескольких ресурсах, но это не даёт гарантии, что номер не будет угнан. UIN может быть потерян в случае, когда соединение идёт напрямую с серверами ICQ, т.к на месте администратора серверов провайдера, может сидеть такой же недобросовестный администратор как и владелец «снифающего» exit-node'a или Jabber-сервера. По этому поводу ничего советовать не буду и оставляю право выбора за читателем.

Для IRC я пользуюсь Xchat (версия 2.8.6) - кроссплатформенный клиент для этого протокола. Тут так же всё просто, только хочу отметить некоторые моменты. При настройке соединения через прокси «Настройка=>Установки=>Сеть=>Настройки Сети» в выпадающем списке «Использовать прокси для…» ставим «Все соединения». «...=>Сеть=>Передача файла» в списке

«Автоматически принимать файлы» выбрать «Нет». По возможности необходимо пользоваться только защищённым соединением. Благо сейчас все IRC-серверы поддерживают SSL(Xchat=>Список сетей=>Править). К сожалению этот протокол устроен так, что с помощью CTCP запросов возможно узнать о пользователе некоторую информацию. Необходимо поставить запрет на такие запросы. «Настройка=>Дополнительно=>Ответы CTCP» и в появившемся окне удаляем все ответы и нажимаем «Записать». Далее в окно ввода сообщения пишем:

/ignore *!*@* CTCP
!@* добавлен в список игнорирования,

переходим «Окно=>Список игнорирования» и ставим галочки «CTCP» и «DCC»



Для ещё большей безопасности отключаем показ версии:

/set irc_hide_version on

По желанию, для всех выше перечисленных протоколов можно отключить логи. Для qutIM в «Настройки=>История, для Xchat Настройка=>Установка=>Протоколирование», и по возможности отказаться от услуг передачи файлов. В завершении хочу порекомендовать пользоваться дополнительными средствами шифрования. Существуют плагины, которые шифруют сообщения, и прочитать их может только адресат, т.е. тот у которого есть «ключ». Примеры таких плагинов: GPG Crypt и qutIM Coder для qutIM, xchat-mircryption для Xchat.

Настойка Email

Для отправки/получения электронной почты я использую web-интерфес почтового сервера. Причины, которые заставили меня это делать несколько. Первая и, наверное, самая главная заключается в том, что многие программы получения/отправки почты очень тяжело настроить на работу с Tor, т.к. многие сервера сети просто блокируют почтовый трафик. Вторая заключается в дописывание этими программами «лишней» информации в заголовок письма, что крайне нежелательно. Об этом писал ЗАРАЗА в своей статье «Утечка данных через служебную информацию и сетевой протокол в клиентском приложении». В случае с web-интерфейсом дополнительный фильтр в виде Privoxy (или другого web-фильтра) уже выставлен и настраивать ничего не надо. Третья, более индивидуальная. Я получаю не много корреспонденции, поэтому в установке и настройке дополнительного ПО не вижу смысла. В дополнение хотелось бы порекомендовать использование дополнительных средств шифрования, которые должны увеличить безопасность электронной переписки.

Настройка «остального» ПО

Под «остальным» ПО подразумевается множество сетевых приложений, которые имеют место быть. Я лишь описал необходимый минимум, который понадобится пользователю во время работы. Многие программы, такие как lynx, links, wget и т.д. также требуют тонкой настройки. Они поддерживают множество параметров, которые непременно помогут в более безопасном и анонимном их использовании. Это не удивительно, т.к. эти приложения работают по протоколу

HTTP. Вспомните о тонкой настройке браузера или загляните в мануалы по этим программам и всё станет ясно. Конечно же, это относится не только к протоколу HTTP, но и ко многим другим программам, работающим на транспортном уровне с TCP.

Многие программы имеют глобальные файлы конфигурации, т.е. не важно под каким пользователем будут их использовать. При запуске они будут принимать конфигурацию установленную в этих файлах. Примеры таких файлов вышеописанных программ находятся в /usr/local/etc/lynx.cfg и /usr/local/etc/wgetrc. Лучше всего глобально настроить их сразу на максимальную степень конфиденциальности, а если необходимо сделать какие-то индивидуальные настройки, их можно переопределить в пользовательских файлах конфигурации - .lynxrc и .wgetrc. Это даёт дополнительную прегради от факта, что пользователь случайным образом выйдет в сеть под другим пользователем, но уже не анонимно. Существуют так же переменные окружения http_proxy/HTTP_PROXY/и т.д., которые также лучше всего занести в глобальный конфигурационный файл своего командного интерпретатора, а уже потом, по необходимости, переопределить в пользовательском файле.

Приложения, которые не работают с TCP не просто использовать вместе с Tor. К таким приложениям относятся: ping, host, dig, nslookup, traceroute и т.п. Вместо них можно воспользоваться онлайн-утилитами, которых великое множество во всемирной паутине или же найти альтернативу. В замену стандартным утилитам ping/traceroute можно воспользоваться hping'ом. Она может «пинговать» хосты не только при помощи ICMP пакетов, которые часто отвергаются брандмауэрами, но также и при помощи TCP и UDP-пакетов. Эта утилита позволяет, при помощи TCP пакетов с плавно изменяющимся TTL, выяснить маршрут к хосту, даже если этого не смогла сделать программа traceroute. Принцип работы тот же, но TCP пакеты, мягко говоря, реже уничтожаются брандмауэрами. Можно написать на стандартные утилиты алиасы которые будут заменяться hping'ом. Необходимо добавить в файл /etc/csh.cshrc (или глобальный конфигурационный файл другого командного интерпретатора) строки:

alias ping proxychains hping -p 80 -S
alias traceroute proxychains hping -p 80 -S —traceroute

Примечание: этот вариант ping'a и tracerout'a будет работать только в том случае, если на проверяемом хосте стоит web-сервер, слушающий 80-й порт(флаг -p 80). Если используется другой командный интерпретатор, то формат записи алиасов может быть другой. В bash, например, необходимо добавить в /etc/profile строки:

alias ping='proxychains hping -p 80 -S'
alias traceroute='proxychains hping -p 80 -S —traceroute'

Таким же образом можно создать алиас на любую необходимую программу и связать её с torify/proxychains, что бы не набирать постоянно torify/proxychains <программа>.

На обычные пинги ресурс не отвечает:

```
toshiba# ping forum.web-hack.ru
PING forum.web-hack.ru (81.177.3.229): 56 data bytes
^C
--- forum.web-hack.ru ping statistics ---
2 packets transmitted, 0 packets received, 100.0% packet loss
```

а если воспользоваться hping'ом, то видно, что ресурс функционирует в нормальном состоянии.

```
toshiba# proxychains hping -p 80 -S -c 4 forum.web-hack.ru
ProxyChains-3.1 (http://proxychains.sf.net)
|DNS-request| forum.web-hack.ru
|S-chain|-<>-127.0.0.1:9050-<><>-4.2.2.2:53-<><>-OK
|DNS-response| forum.web-hack.ru is 81.177.3.229
HPING forum.web-hack.ru (tun0 81.177.3.229): S set, 40 headers + 0 data bytes
len=44 ip=81.177.3.229 ttl=57 DF id=37718 sport=80 flags=SA seq=0 win=65535 rtt=
16.6 ms
len=44 ip=81.177.3.229 ttl=57 DF id=37931 sport=80 flags=SA seq=1 win=65535 rtt=
15.9 ms
len=44 ip=81.177.3.229 ttl=57 DF id=38094 sport=80 flags=SA seq=2 win=65535 rtt=
16.4 ms
len=44 ip=81.177.3.229 ttl=57 DF id=38277 sport=80 flags=SA seq=3 win=65535 rtt=
16.0 ms

--- forum.web-hack.ru hping statistic ---
4 packets tramitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 15.9/16.2/16.6 ms
```

Примечание: некоторые утилиты из стандартного набора все таки удалось заставить работать с Tor'ом. Например, утилита whois прекрасно работает в связке с proxychains.

После того как всё настроено и работает без ошибок обязательно необходимо проверить все настроенные приложения на предмет утечек DNS. Достаточно воспользоваться любым анализатором пакетов, которых великое множество. В базовую систему FreeBSD входит tcpdump-утилита, позволяющая перехватывать и анализировать сетевой трафик, проходящий через компьютер, на котором запущена данная программа. Она позволяет отфильтровать трафик на определённый порт.

#tcpdump -q -i <сетевой интерфейс> port 53

Первый флаг установлен для вывода минимума информации; второй указывает, какой «слушать» интерфейс; третий определяет, какие пакеты «захватывать» (на 53 порту по умолчанию обычно принимает пользовательские запросы DNS сервер). Если всё настроено правильно, то снифер не должен захватить ни одного пакета.

Рекомендации по использованию Tor

Тут хотелось бы отметить некоторые рекомендации, которые помогут читателю не раскрыть свою анонимность и оставаться бдительным.

- ➔ По возможности используйте для анонимной и «не анонимной» работы два разных подключения к Интернет, два разных компьютера, две учётные записи (было бы хорошо =)). Как правило, под одной и той же учетной записью используется одинаковое ПО и, например, если пользователь зашёл на сайт не анонимно с включёными cookie, а потом посетил этот же сайт с включенными опциями анонимности, то его можно будет идентифицировать по cookie, которые были переданы в прошлый сеанс работы. Если все таки приходится работать под одной учётной записью установите дополнительное ПО для работы с разными степенями конфиденциальности.

- ➔ Перед тем как использовать ту или иную утилиту, попробуйте найти способы ее «анонимизации», почитайте отзывы пользователей, зайдите на сайт производителя ПО и ознакомьтесь со способами ее тонкой настройки.
- ➔ После настройки ПО обязательно протестируйте его работу в связке со снифером.
- ➔ По возможности отключайте автоматическое обновление или проверку новых версий ПО.
- ➔ Не используйте одну и туже учётную запись на форумах, сайтах, чатах и т.п. когда для разного вида серфинга (анонимный/неанонимный).
- ➔ Будте внимательны к сообщениям браузера когда работаете по защищённому протоколу.
- ➔ Следите за новостями об уязвимостях, найденных в Tor и новых атаках, которые были произведены на эту сеть. Возможно, там будут описаны варианты защиты.
- ➔ Следите регулярно за выходами новых версий Tor'a, в них может быть исправлена критическая уязвимость безопасности.
- ➔ Так же на некоторых ресурсах «для параноиков» советуют следить за тем, что мы пишем в Сети, так как хороший лингвист может определить, что два разных, на первый взгляд, пользователя это есть один и тот же человек. Выводы делаем сами ;).

Это конечно же не полный перечень того, что можно посоветовать для увеличения степени работы во всемирной паутине. Возможно, читатель дополнит этот список своими рекомендациями.

Послесловие

Я не считаю Tor панацеей от всех проблем, но если прикинуть соотношение цены и качества, то это вполне работающий инструмент, тем более, если необходимо быть анонимным один-два раза, при этом не затрагивая финансовую сторону. Есть множество плюсов и минусов, которые заставляют задуматься об использование данного средства как инструмента для сокрытия реальной информации о себе. В Сети на эту тему существует множество обсуждений, которые имеют как сторонников, так противников данной сети. Моё мнение: вполне реально использовать Tor для анонимного сёрфинга, но для более «серьёзных» дел (например для банковских транкзаций) он подходит с большой натяжкой и используется на свой страх и риск. Несомненно, идеальной анонимности не существует, а близкая к тому, только та, которая создана своими руками (например, взломанный сервер с установленным соответствующим ПО). Выбор остаётся за конечным пользователем который решает «Быть или не быть».

P.S. Буду рад любой обоснованной критике, которая, так или иначе, покажет полезность этого материала.