Установка и настройка ОС Whonix

Whonix – это дистрибутив, который базируется на Debian и при этом состоит из двух частей. Когда я говорю «из двух частей», я имею в виду, что для работы Whonix необходимы две виртуальные машины. Первая – это шлюз (Whonix Gateway), который работает только через Tor и Торифицирует абсолютно весь траффик, а вторая – полностью изолированная рабочая станция (Whonix Workstation), настроенная таким образом, что подключается только к шлюзу и берет интернет только оттуда.

Таким образом, абсолютно все приложения, запущенные на рабочей станции пускают свой траффик через Tor, потому что Workstation берет интернет с Gateway. При этом Workstation не знает свой реальный IP адрес, и если рабочую станцию взломают, злоумышленник так и не сможет узнать ваш реальный IP адрес.

Из особенностей Whonix можно выделить также то, что каждое предустановленные в нем приложение работает на отдельном Socks-порту, что означает, что для каждого такого приложения создается отдельная цепочка из узлов Tor. Whonix отлично защищен от утечек DNS. У Whonix хорошая защита от идентификации пользователя при помощи так называемого Fingerprinting.

Ходит миф, что Tails является самой анонимной ОС, которую когда-либо придумывали. Хочу сказать, что Whonix ничем не хуже в этом плане, а в некоторых моментах даже значительно превосходит.

Tails и Whonix. Оба дистрибутива (или ОС, называйте как хотите) создавались с уклоном в анонимность и безопасность пользователей. И Whonix, и Tails действительно очень хороши и прекрасно выполняют свои функции.

Если не углубляться до технических особенностей каждого из

дистрибутивов, то для типичного пользователя принципиальной разницей между ними будет то, что Tails предназначен для использования с USB-FLash накопителе (не рекомендуется ставить на виртуалку), а Whonix в большенстве случаев ставится на виртуалку. А также то, что Tails является «Amnestic», то есть все забывает после перезагрузки, а Whonix нет. Еще одним не мало важным отличием является то, что Whonix более гибок в настройке в принципе, а в настройках различных «Цепочек Анонимностей» в частности, тут тебе и VPN -> Tor, и VPN -> Tor -> VPN, и VPN — Tor -> Proxy/SSH и еще можно много чего придумать. В Tails, такого сделать нельзя. В Tails весь траффик идет сразу через Tor, и, например, поставить VPN перед Tor уже не получится, только после.

Tails очень удобен, если нужно быстренько воткнуть Флешку, зайти в Интернет, что-то где-то написать и выключить. Для продолжительной работы в Интернете его очень не удобно использовать, чего не скажешь о Whonix.

Конечно можно Tails настроить таким образом, что он не будет забывать нужные вам установленные приложения, конфигурационные файлы, всякие настройки и так далее, но зачем делать такие костыли?

Способы установки Whonix

1) Whonix на вируальной машине. В этой статье будет разобран именно этот способ. Это самый простой и распространенный способ поставить себе Whonix. Программой виртуализации у нас будет VirtualBox. В качестве Host OS может использоваться чуть ли не любая операционная система.

2) Qubes-Whonix. Это второй, тоже довольно распространенный способ пустить траффик через Whonix. В качестве хостовой операционной системы используется так называемый Qubes OS, а Whonix-Gateway ставится как виртуалка через встроенные средства виртуализации в Qubes OS.

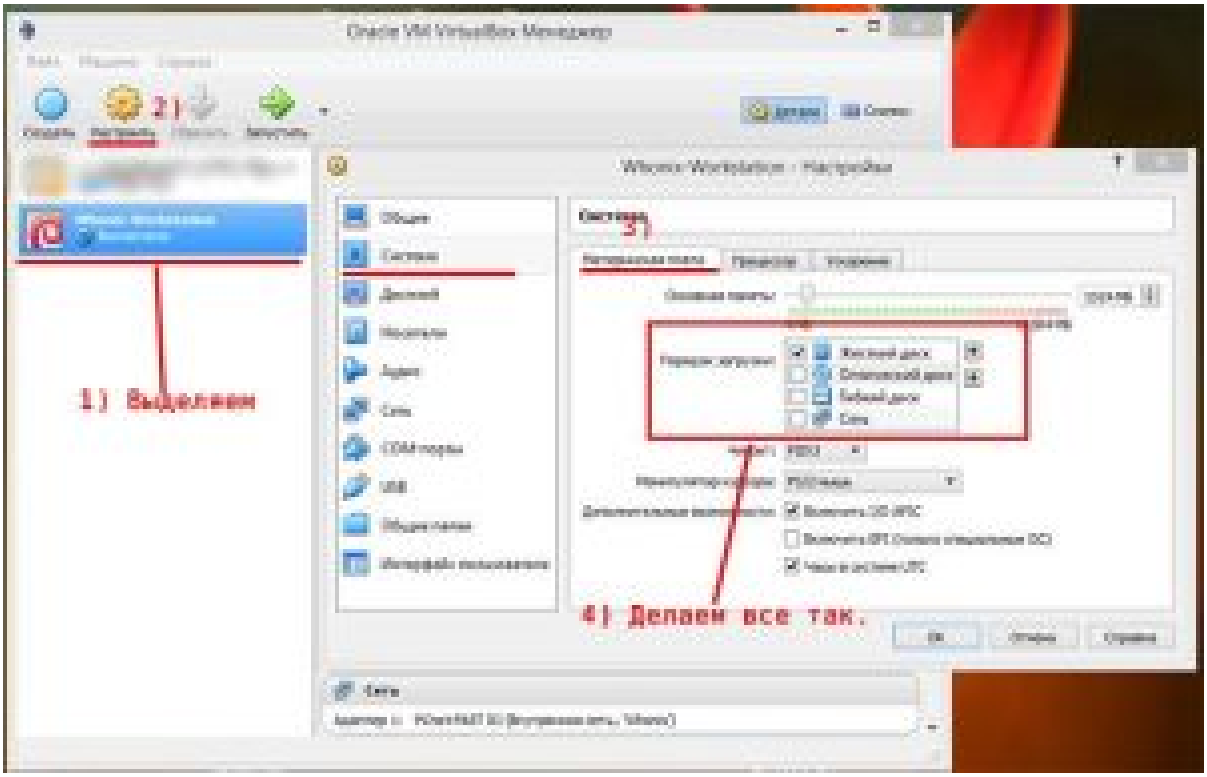
3) Виртуализация KVM. Третий способ ставить Whonix. Используется виртуализатор qemu-kvm или подобное.

4) Возможно физическое разделение виртуальных машин Whonix Workstation и Whonix Gateway

Как установить Whonix

Для начала нужно скачать VirtualBox. Теперь идем на сайт Whonix и скачиваем два образа «.ova» – Whonix Gateway и Whonix Workstation. С этой страницы скачиваем образы: Образы для VirtualBox. По ссылкам «Download Whonix-Gateway» и «Download Whonix-Workstation».

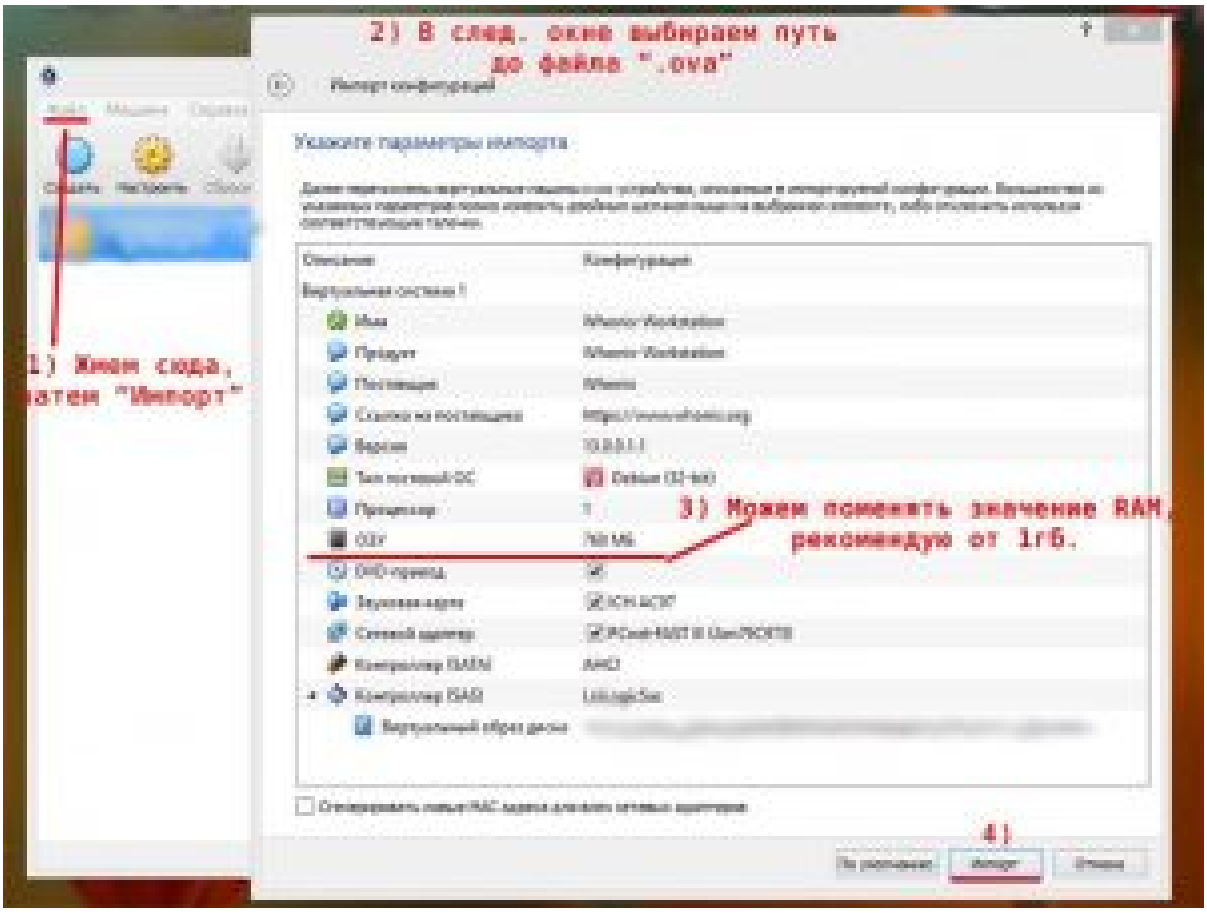
Открываем виртуалбокс, жмем File -> Import Appliance. Выбираем скаченный образ Whonix Gateway, жмем «Далее». Появится окошко с настройками, все оставляем по-умолчанию, кроме графы RAM, эту графу по желанию можно изменить. Жмем «Import» и ждем.



Точно такую же операцию проделываем для Whonix Workstation. Лично я для Whonix Workstation выделяю обычно не меньше 1024mb RAM (если есть возможность, выделяйте побольше гига), а для

первого запуска Whonix Gateway оставляю по-умолчанию (768mb).

Теперь у нас в нашем виртуалбоксе появились две виртуальные машины, но мы их пока не запускаем. Немного настроим. Жмем на виртуалку, потом «Настройки», переходим на вкладку «System». Тут мы должны поменять порядок загрузки и снять лишние галочки. Выбираем «Hard Disk» и стрелочками перемещаем его на первую строку, «Optical» – на вторую строку. Снимаем галочки с «Floppy» и «Optical», оставляем только на «Hard Disk». Теперь переходим во вкладку «Storage», там на Контроллере ставим галочку «Use Host I/O Cache». Точно такую же операцию проделываем и со второй виртуалкой.



Как запустить Whonix

Запускаем GW (Gateway) и WS (Workstation). Порядок запуска таков: сначала Gateway, затем Workstation.

Перед нами сразу появится окно, дважды жмем «Understood» (но

сначала читаем!). На GW должно быть отмечено «Iam Ready to Enable Tor», ждем «Next» несколько раз, потом «Yes. Automatically install updates from the Whonix team», жмем «Next», выбираем «Whonix Stable Repository» (рекомендую выбирать именно это) и далее до конца. На WS все аналогично. Теперь на обеих виртуалках автоматически должен запуститься так называемый whonixcheck.

Когда whonixcheck прошел, он покажет «Warning», где будет ругаться на то, что система не обновлена.

На обеих виртуалках открываем эмулятор терминала Konsole с ярлыка на рабочей столе.

На обеих VM вводим

```
sudo apt-get update && sudo apt-get upgrade -y && sudo apt-get dist-upgrade -y
```

Спросит пароль.

Стандартный логин – user

Пароль от user – changeme

Пароль от root – changeme

Позже мы поменяем пароль. Вводим пароль -> сидим с открытым ртом и ждем пока закончится обновление, процесс не быстрый. Обновление может прерваться по неизвестным причинам, если это случилось, повторно вводите команду выше, процесс продолжится.

Если все прошло окей, то повторно запускаем whonixcheck – вводим в терминале whonixcheck или запускаем с ярлыка на рабочем столе WhonixCheck. Лично я предпочитаю делать whonixcheck в терминале. Смотрим, чтоб не было «Warning». Если есть «Warning» в том же месте что и в прошлый запуск, то возвращаемся на абзац выше и обновляем систему. Если нет «Warning», то идем дальше.

Теперь на Workstation нам нужно скачать/обновить Tor Браузер.

Запукаем Tor Browser Downloader с ярлыка на раб. столе или вводим update-torbrowser в терминале. Начнется процесс, вам будет предложено выбрать версию Тор Браузера для скачивания, 6.0.7 и остальные. Версия 6.0.7 – стабильная версия, все остальные версии – тестинг или девелоперс. Я ставлю 6.0.7, вы ставите что хотите (лучше стабильную). В случае если программа была запущена с ярлыка на раб. столе, отметьте нужную версию кружочком (радиобаттон), в случае запуска скрипта через терминал, введете нужную версию вручную, как только предложат это сделать (в моем случае я ввел «6.0.7») и жмете интер. После скачивания, вам будет предложено установить его, жмете кнопку ‘Yes’ или вводите «y» в терминале и жмете интер. Версия «6.0.7» актуальна на момент написания статьи.

Меняем стандартный пароль. Открываем терминал, вводим

sudo -i

Водим пароль, теперь мы работаем из под root. Меняем пароль, сначала для root

passwd root

Дважды вводим пароль. Теперь пароль от root поменялся. Теперь для user

passwd user

Дважды вводим пароль. Теперь пароль от user поменялся. Теперь вводим exit.

Желательно чтобы пароль от root и от user были разные. Данную операцию можно проделать на обеих виртуалках.

Обновляем локали. Вводим

sudo dpkg-reconfigure locales

Появится графическое окошко. Навигация вверх-вниз – стрелочками на клаве. Листаем вниз и ищем <ru_RU.UTF-8 UTF-8>. Отмечаем по нажатию на пробел. Через нажатие на Tab

переключаемся на <Ok>. Теперь нам предложат выбрать язык в системе по умолчанию, здесь на ваше усмотрение. Я оставлю английский (чего и вам рекомендую), поэтому выбираю <en_US.UTF-8>, а вы можете выбрать <ru_RU.UTF-8>.

Ставим некоторый софт/пакеты, которые могут вам понадобиться

```
sudo apt-get install htop git openvpn openssl nmap psi-plus etherape openssh-client
```

htop – консольный task manager

git – git

nmap – сканер портов

psi-plus – Jabber клиент. Вместо него может быть pidgin

etherape – для графического мониторинга сетевого траффика

openssh-client – чтоб подрубаться по ssh куда-нибудь

Еще ставим данную группу пакетов

```
sudo apt-get install build-essential pkg-config make automake autoconf
```

Работа с Whonix Gateway

Лично я запуская GW в консольном режиме, потому что во-первых GUI жрет не мало RAM, во-вторых GUI по большому счету не нужен GW – все операции можно провести через консоль, все ярлыки на раб. столе спокойно заменяются командами в консоли.

Итак, для того чтобы запускать GW в консольном режиме, нужно выделить виртуалке поменьше RAM. По-умолчанию, минимально значение RAM, которое должно быть выделено для GW, чтобы он запускался в GUI режиме – 480mb. Значит, если выделено меньше 480mb, то GW запустится в консольном режиме.

Идем в виртуалбокс, выбираем GW -> жмем настройки -> переходим

на вкладку «System» -> перемещаем ползунок. Выбираем значение меньше 480.

Но что делать, если мы хотим выделить GW побольше RAM, скажем, 512 или оставить 768, но при этом запускать в консольном режиме? Редактируем файл /etc/rads.d/30_default.conf

sudo nano /etc/rads.d/30_default.conf

Ищем строчку «rads_minimum_ram». Эта строчка сообщает системе, какое минимальное количество RAM должно быть выделено, чтобы машина запускалась в GUI режиме. Как видите, по умолчанию стоит 480. Берем и меняем значение на любое другое, скажем, 1024 -> сохраняем -> выходим. Идем в настройки вирт. машины Gateway (выше) и выделяем сколько нужно RAM и в пределах 1024mb – будет запускаться в консольном режиме.

Ярлыки на Whonix Gateway

Грубо говоря, ярлыки условно можно разделить на пять групп:

1) Ярлыки, для управления Tor (Stop Tor, Reload Tor, Restart Tor)

2) Ярлыки, для управления Firewall (Global Firewall Settings, User Firewall Settings, Reload Firewall)

3) Ярлыки, для редактирования конфигурационных файлов Tor (Tor User Config, Tor Examples, Tor Data)

4) Ярлыки Whonix (WhonixCheck, WhonixSetup, Whonix Repository)

5) Остальные ярлыки – приложения (Konsole, Arm)

Stop Tor – остановить службу Tor. После выполнения пропадет интернет и приложения на Whonix Workstation работать не будут. Заменяется следующей командой в консоли

sudo service tor@default stop

Reload Tor – перезагружает службу Tor. При выполнении

перечитывает конфигурационные файлы и перезагружает цепочку Tor для приложений на Whonix Workstation. Заменяется следующей командой в консоли

sudo service tor@default reload

Restart Tor – перезапускает службу Tor. Трудно сейчас будет объяснить чем отличается от Reload Tor. Сначала служба Tor останавливается, затем запускается по новой. Если вам нужно перезагрузить цепочку, то используйте Tor Reload. Заменяется следующей командой в консоли

sudo service tor@default restart

Global Firewall Settings – открывает глобальные настройки Firewall по адресу /etc/whonix_firewall.d/30_default.conf в текстовом редакторе kwrite. Если вы не знаете, зачем это нужно, то данный ярлык не используйте. Заменяется следующей командой в консоли

sudo nano /etc/whonix_firewall.d/30_default.conf

User Firewall Settings – открывает пользовательские настройки Firewall по адресу /etc/whonix_firewall.d/50_user.conf в текстовом редакторе kwrite. Если вы не знаете, зачем это нужно, то данный ярлык не используйте. Заменяется следующей командой в консоли

sudo nano /etc/whonix_firewall.d/50_user.conf

Reload Firewall – перезагружает Firewall, если вдруг вы внесли изменения в /etc/whonix_firewall.d/30_default.conf или /etc/whonix_firewall.d/50_user.conf. Заменяется следующей командой в консоли

sudo whonix_firewall

Tor User Config – открывает основной конфигурационный файл /etc/tor/torrc в текстовом редакторе kwrite. Если вы не знаете ничего про конфигурационный файл torrc, зачем он нужен, что туда добавлять и как с ним работать, то неичего там не

меняйте. /etc/whonix_firewall.d/30_default.conf или /etc/whonix_firewall.d/50_user.conf. Заменяется следующей командой в консоли

sudo nano /etc/tor/torrc

Tor Examples – открывает файл /etc/tor/torrc.examples, который является примерно конфигурационного файла /etc/tor/torrc в текстовом редакторе в режиме «только для чтения». Это пример, в этом файле не нужно ничего редактировать. Если интересно – можете почитать. Заменяется следующей командой в консоли

nano /etc/tor/torrc.examples

Tor Data – открывает папку /var/lib/tor/ в файловом менеджере Dolphin. Заменяется следующей командой в консоли

cd /var/lib/tor/

WhonixCheck – запускает проверку. Нельзя запускать от рута.Заменяется следующей командой в консоли

whonixcheck

WhonixSetup – запускает Хуниксовский Setup Wizard. Заменяется следующей командой в консоли

sudo whonixsetup

Arm – мощный инструмент для мониторинга Tor. С помощью него можно контролировать Tor различным образом. Заменяется следующей командой в консоли

arm

Работаем с Jabber

Установка Psi+

sudo apt-get install psi-plus

Установка Pidgin

sudo apt-get install pidgin

Установка Gajim

sudo apt-get install gajim

Установка на примере клиента Psi+.

Ставим Psi+

sudo apt-get install psi-plus

Запускаем – открываем whiskermenu/applications menu, вводим в поиске Psi+ – открываем/переносим мышкой на раб. стол, появится ярлык.

Также запустить можно через терминал

psi-plus %U

Ярлык на раб. столе можно создать вручную

touch ~/Desktop/psi.desktop

Редактируем только что созданный файл

nano ~/Desktop/psi.desktop

или

mousepad ~/Desktop/psi.desktop

mousepad – это графический текстовый редактор, который идет вместе с xfce4. Если вы не ставили xfce4, то используйте вместо mousepad – редактор kwrite. Во всех командах мысленно «mousepad» меняйте на «kwrite»

И вводим туда следующее

```
[Desktop Entry]
# This is the spec version, *not* the application version
Version=1.0
Type=Application
Name=Psi+
GenericName=XMPP Client
```

```
Comment=Communicate over the XMPP network
Icon=psi-plus
Exec=psi-plus %U
MimeType=x-scheme-handler/xmpp;
Terminal=false
StartupWMClass=psi-plus
Categories=Network;InstantMessaging;Qt;
Keywords=XMPP;Jabber;Chat;InstantMessaging;
```

Если через nano редактируете, то вставляется текст с помощью ctrl+shift+v). Сохраняем -> закрываем. Все, ярлык готов.

Запускаем Psi+. Регистрация аккаунта. При запуске вылетет окно «Account Setup». Если хотите зарегистрировать аккаунт, то жмем «Registaer new account», если у вас уже есть аккаунт, жмем «Use existing account» (если вдруг никакого окна не вышло, то переходите к разделу «Добавление аккаунта» чуть ниже, а потом возвращайтесь сюда).

После нажатия «Registaer new account», вылетает окно регистрации нового аккаунта. В первой графе нужно ввести джаббер сервер. Например это могут быть следующие

```
exploit.im
zloy.im
swissjabber.ch
xmpp.jp
crypt.am
```

Я буду показывать регистрацию на примере сервера swissjabber.ch, хотя процесс регистрации на других серверах, почти ничем не отличается.

Вводим swissjabber.ch в первое поле, жмем «Next» -> появится окно, вводите ваш Username и Password.

Я ввожу Username: SupeDealer

Username не чувствителен к регистру, то есть SupeDealer = supedealer.

Жмем «Next» – выйдет окно, подтверждающее, что регистрация прошла успешно.

JID (Jabber IDentifier) – грубо говоря ваш адрес вида username@server.
В моем случае JID получился supedealer@swissjabber.ch

Затем выйдет окно с настройками аккаунта, состоящее из нескольких вкладок. Тут можно ничего не трогать, единственное что, переходим во вкладку «Misc.» -> там в «Resource» ставим «Manual».

Подключаемся -> на аккаунте правой кнопкой мыши -> Status -> Online.

Выскочит окно с ошибкой и предложеним написать о себе.

Во вкладке «General» заполните поле «Full Name» (пишем сюда то, как хотели бы, чтобы ваш аккаунт отображался у других пользователей), в поле Nickname тоже самое -> жмем «Publish».

Добавляем контакт

ПКМ на аккаунте -> «Add a contact» или в окне Psi+, вверху, есть иконка «человечек с плюсом», жмем на нее.

Во вновь появившимся окне напротив «XMPP Address» пишем JID человека, которого хотим добавить. Как видите, можно ввести Nickname (то, как он будет отображаться у вас в контактах) и группу (можно делать группы).

Я зарегистрирусь второй раз, мой JID будет supedealer@swissjabber.ch и добавлю свой первый аккаунт.

Напротив «XMPP Address» ввожу supedealer@swissjabber.ch, Nickname напишу «spb-dealer», «Group» – напишу «Drugs» -> жму -> выскочит окно с информацией о том, что этот контакт добавлен в ростер.

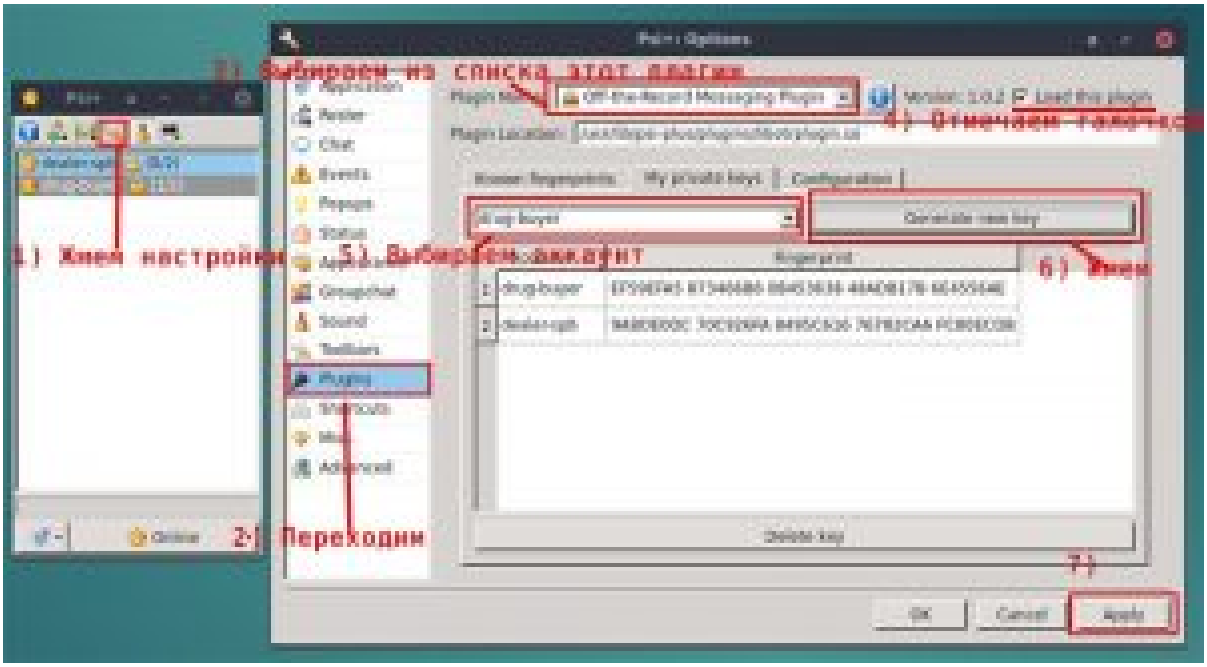
На нашем аккунте supedealer@swissjabber.ch появилось

уведомление о том, что кто-то нас хочет добавить. Выскочит системное сообщение, там жмем кнопку <Add/Auth>, тем самым мы принимаем заявку, так сказать, «в друзья». После вашей аутентификации собеседника, ему придет точно такое же системное сообщение.

Настроим шифрование OTR (Off-The-Record)

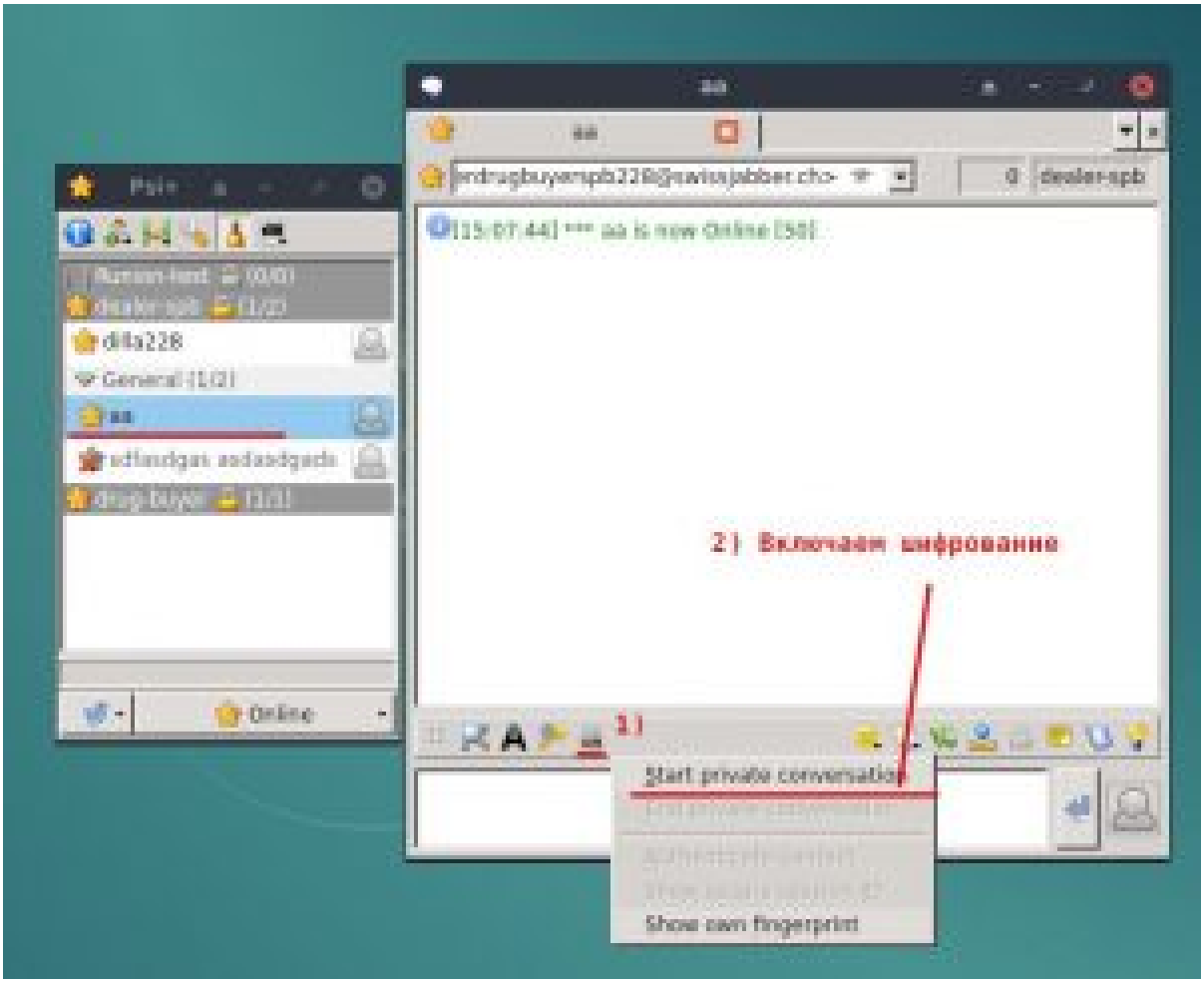
В главном окне Psi+ вверху есть иконка с ключем, жмем ее. Это настройки. Если что-то нужно настроить — настраивайте. Идем в раздел «Plugins» -> напротив «Plugin name» во всплывающем списке выбираем «Off-The-Record messaging Plugin» -> помечаем галочкой «Load this plugin». В появившимся ниже окне, переходим во вкладку «My private keys» -> там в выпадающем списке выбираем свой аккаунт -> жмем -> подтверждаем -> все.

Генерировать подобные ключи нужно для каждого вашего аккаунта. Не обязательно, но рекоменду перейти во вкладку «Configurations» и отметить радиобаттоном «Automatically start private messaging». Шифрование OTR настроено.



Врубаем шифрование в чате. В окне чата с собеседником будет иконка «замочка, на котором надпись OTR» -> жмем ее -> жмем «Start private messaging». Поскольку ранее, в настройках плагина мы поставили галочку на «Automatically start private

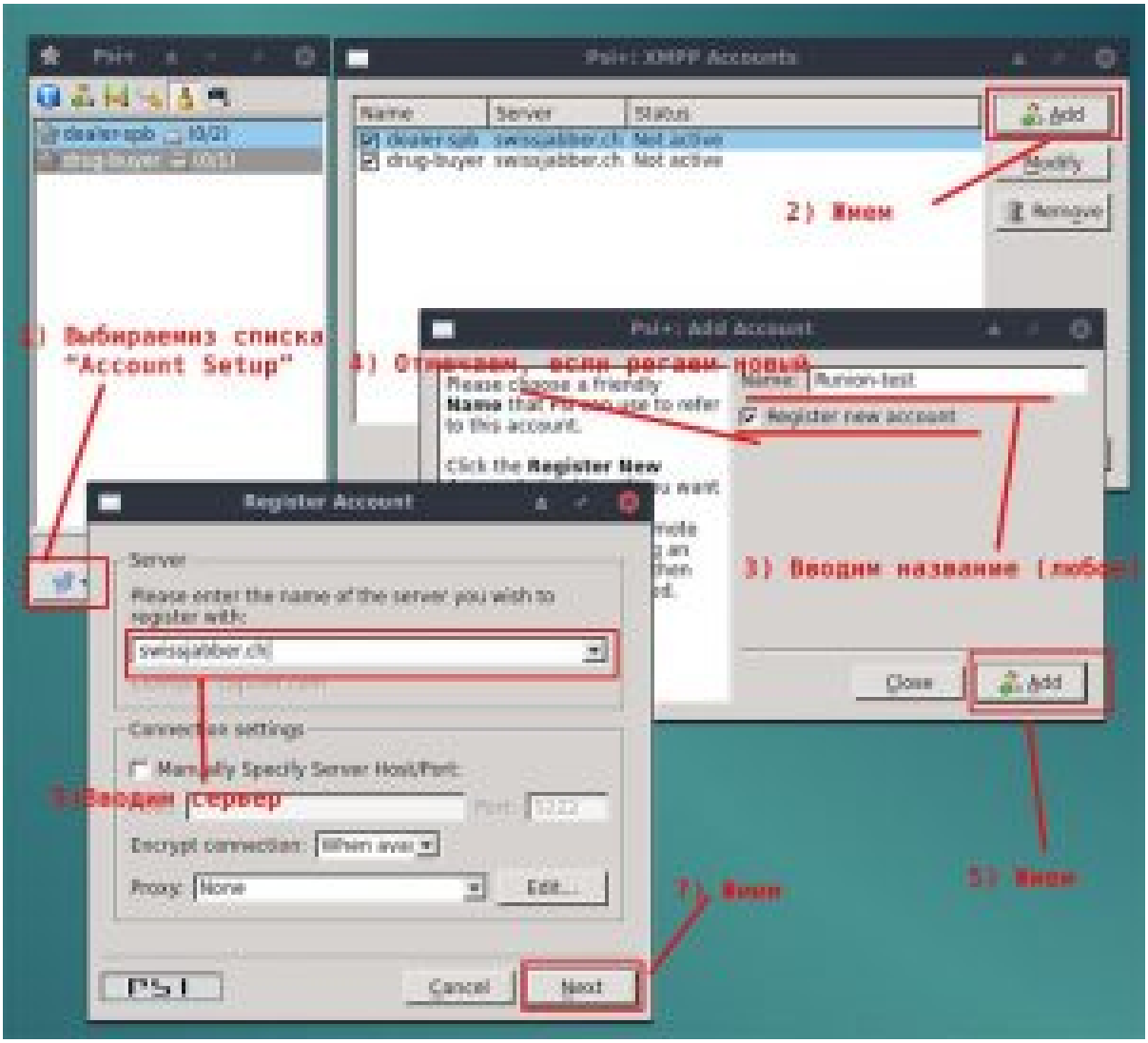
messaging», то шифрование начнется автоматически при отправке первого сообщения.



Добавление аккаунта

В главном окне Psi+, внизу слева есть кнопка, по нажатию на которую будет открыт выпадающий список. В этом списке ищем и жмем «Account Setup». Появится новое окно, в нем жмем <Add> -> появится окно добавления аккаунта, графа «Name» означает то, как будет выглядеть аккаунт у вас в окно Psi+, здесь может быть написано все что угодно. Если регистрируем новый аккаунт, то отмечаем галочкой «Register new account». Выскочит окно регистрации нового аккаунта, которое мы разбирали выше, в разделе регистрации.

А для того чтобы добавить существующий аккаунт, снимите галочку, потом вводите JID и пароль, жмете Save.



Советы по работе с Whonix

Не нужно скачивать и ставить Tor на Хуникс Воркстейшн (не надо делать apt-get install tor). На Воркстейшн стоит Tor Браузер. Им и пользуйтесь. Остальные приложения тоже получают интернет с Gateway, который полностью Торифицирует весь траффик.

Не рекомендую пытаться делать Tor -> Tor (Tor через Tor). Это лишнее. Такая связка вам ничего не дает.

Если вам на выходе НЕ нужен белый IP и вы ходите исключительно на httpS сайты или на сайты в пространстве .onion, то в качестве браузера используйте Tor Браузер.

Будьте осторожны, если выполняете apt-get autoremove, иногда по неведомым причинам, удаляются важные Хуниксовские пакеты

или конфиги, которые нужны для нормальной работы. Можно удалить так, что ничего нормально работать не будет, в том числе и интернет. И придется ставить Воркстейшн по новой.

Старайтесь ничего не удалять из того, что не ставили сами. Это может привести к плохим последствиям.

Лучше не меняйте репозитории со Stable на Testing для обновления системы. Придется делать apt-get dist-upgrade, которые вероятно снесет/обновит то, что не нужно и у вас все поламается, придется ставить Workstation по новой.

Если нужно поставить какой-то пакет, которого нет в Stable репозиториях, то ищем на оф. сайте программы, скачиваем и собираем вручную, или ищем «.deb» пакет на сайте Debian, скачиваем и ставим через dpkg -i.

Если совсем все плохо, и в вручную не собирается, и deb пакет не ставится (проблема с зависимостями), а скачивать пакеты-зависимости отдельно, а возможно и пакеты-зависимости от пакетов-зависимостей вам в падлу, то можно временно добавить Testing репозиторий.

Делайте так: добавляете Testing репозитории

sudo apt-get update

СТРОГО ТОЛЬКО ЭТУ КОМАНДУ, без upgrade и dist-upgrade, ставите нужный пакет через команду

sudo apt-get install <пакет>/testing

После того как поставился удаляем добавленный Testing репозиторий из файла и еще раз выполняем sudo apt-get update. Будьте внимательны.

Как добавить и удалить Testing репозитории

Прежде чем проводить манипуляции с файлом, сделаем бекап на всякий случай

```
sudo cp /etc/apt/sources.list.d/debian.list /etc/apt/sources.list.d/debian.list.bak
```

Для начала войдем в режим работы от root

```
sudo -i
```

При необходимости вводим пароль. Затем, для того чтобы добавить Testing репозитории, вводим

```
echo "deb http://ftp.us.debian.org/debian testing main contrib non-free" >> /etc/apt/sources.list.d/debian.list
```

Testing репозитории добавлены. Выходим из под root

```
exit
```

Обновляем репозитории

```
sudo apt-get update
```

Для того чтобы вернуться к первоначальному состоянию, нужно с помощью редактора nano отредактировать файл /etc/apt/sources.list.d/debian.list, удалив последнюю строчку (ту, которую мы добавили при помощи echo).

```
sudo nano /etc/apt/sources.list.d/debian.list
```

Удаляем последнюю строку. Или же восстанавливаем файл из созданного ранее бекапа

```
sudo cp /etc/apt/sources.list.d/debian.list.bak /etc/apt/sources.list.d/debian.list
```

И снова обновляем репозитории

```
sudo apt-get update
```

Пример

```
sudo apt-get install rofi/testing
```

Команда установит пакет rofi из Testing репозитория. Поставьте себе менеджер паролей keepassx. KeePassX – это некий сейф для хранения паролей к различный сайтам и сервисам.

```
sudo apt-get install keepassx
```

Если вы чувствуете, что где-то накосячили или у вас что-то работает не так, и никак не можете решить эту проблему, даже гугление не помогает, то сносите Workstation и ставьте по новой. В Виртуалбоксе выделяете Whonix Workstation -> жмете правой кнопкой мыши -> жмете удалить -> удалиьт все файлы.

Возвращаетесь в начала стать и ставите Workstation заново. При этом Gateway можете не трогать (если вы и там не на портачили).

Настраиваем Firefox

Настоятельно не рекомендую вам устанавливать какие-либо другие браузеры (Google Chrome, Chromium, Opera, Yandex Browser, Amigo и так далее). Используйте Tor Browser или Firefox.

В Whonix-Workstation предустановлен браузер Firefox ESR (Extended Support Release), далее просто Firefox.

А где же Iceweasel, спросите вы. А нет Iceweasel в Whonix. Приложение iceweasel является симлинком на firefox-esr

```
user@host:~$ which iceweasel
/usr/bin/iceweasel
user@host:~$ ls -l /usr/bin/iceweasel
lrwxrwxrwx 1 root root 30 Dec  1 00:48 /usr/bin/iceweasel ->
../lib/firefox-esr/firefox-esr
```

Поэтому настраивать будем Firefox. Для начала жмем на иконку «бургер» вверху справа -> жмем Preferences. Переходим во вкладку «Search», в «Default Search Engine» в выпадающем списке выбираем «DuckDuckGo». Гугл не будет искать, если запрос с IP Тора.

Переходим во вкладку «Privacy», В выпадающем списке выбираем «Use custom settings for history», ставим галочку на «Accect cookies from sites», напротив «Accept third-party cookies» в выпадающем списке выбираем «Never», напротив «Keep until» в выпадающем списке выбираем «I close Firefox», отмечаем

галочкой «Clear history when FireFox closes».

Теперь изменим конфиг. В адресной строке пишем «about:config», соглашаемся с тем, что будем аккуратны, жмем на кнопку, далее в адресной строке пишем следующие строки и меняем значения

```
media.peerconnection.enabled = false
geo.enabled = false
browser.send_pings = false
browser.safebrowsing.enabled = false
browser.safebrowsing.malware.enabled = false
browser.search.suggest.enabled = false
```

Далее

```
dom.battery.enabled = false
dom.enable_performance = false
dom.network.enabled = false
dom.storage.enabled = false
network.dns.disablePrefetch = true
network.http.sendSecureXSiteReferrer = false
network.prefetch-next = false
network.proxy.socks_remote_dns = true
```

Можно еще добавить

```
network.http.sendRefererHeader = 0
dom.storage.enabled = false
```

Реализация различных цепочек анонимности

VPN -> Tor -> Интернет

VPN в самом начале используется для того, чтобы скрыть сам факт использования Tor от вашего провайдера.

Самым простым вариантом будет использовать VPN на вашей хост. машине. Цепляем VPN на вашу хост. машину и работаем с Workstation. Траффик будет идти сначала через VPN, затем через Tor на Whonix-Gateway.

Для того чтобы использовать VPN (OpenVPN) на хост. машине, то:

1) Если у вас винда, скачиваете OpenVPN клиент.

2) Если у вас Linux, то ставите пакет openvpn

```
sudo apt-get install openvpn openssl
```

3) Если у вас Mac OS X, то ставите TunnelBlick.

Далее скачиваете конфигурационный файл и открываете при помощи вашего OpenVPN клиента.

VPN -> Tor -> VPN -> Интернет

Цепочка, которой хватает для 93% случаев. Второй VPN после Tor используется во-первых, для того, чтобы у вас был нормальный белый IP (если он вам нужен вообще, в зависимости от того, чем вы занимаетесь), а во-вторых спасает от прослушки на выходной ноде Тора, где, если вы заходите на сайт http (не https), то траффик шифроваться не будет.

VPN 1 цепляем на хост. машину, а VPN 2 должен быть на Workstation.

Как подключиться к VPN на Workstation. Как правило, OpenVPN на Workstation уже предустановлен, но можно перепроверить

```
sudo apt-get install openvpn openssl
```

Далее скачиваем конфигурационный файл «.ovpn» от вашего VPN-провайдера. Tor не может работать по UDP, поэтому ваш провайдер должен предоставлять вам конфигурационный файл, который работает с VPN-сервером по протоколу TCP. В конфиг. файле должна быть строчка «proto tcp» вместо «proto udp». Подключаемся к VPN

```
sudo openvpn --config <путь_до_.ovpn>
```

Сидим и работаем через настроенный Firefox. Tor Браузер по прежнему будет показывать IP Тора, потому что так устроено. Не пытайтесь ничего менять в Tor Браузере. Если нужна цепочка VPN

-> Tor -> VPN -> Интернет в браузере, то работаем через Firefox. Открываем Firefox -> заходим на сайт whoer.net -> смотрим свой IP. Если IP не Tor, значит все окей. Проверяем IPшник курлом

```
curl.anondist-orig https://check.torproject.org/ | grep "IP"
```

Теперь траффик почти всех приложений идет через VPN -> Tor -> Интернет. Почему я написал «почти всех приложений»? Потому что Whonix устроен таким образом, что некоторые приложения будут миновать этот VPN (2) и соеденяться сразу с Tor, траффик таких приложений будет VPN (1) -> Tor -> Интернет (при условии, что вы подключились к VPN на своей хост. машине).

Вот список этих приложений. Приложения, настроенные на работу таким образом через внутренние настройки

```
Tor Browser
HexChat
Mozilla Thunderbird with TorBirdy
Instant Messenger
sdwdate
whonixcheck
BitCoin
privoxy
polipo
Tor Browser Downloader by Whonix
Chat#Ricochet IM
Mixmaster
KDE application wide proxy settings
```

Приложения — uwt wrapped

```
apt-get
aptitude
gpg
ssh
git
wget
curl
mixmaster-update
```

Список uwt wrapped приложений

apt-get
aptitude
gpg
ssh
git
wget
curl
mixmaster-update

uwt wrapped приложения – это приложения немного модифицированные командой Whonix. Для каждого отдельного uwt wrapped приложения будет использоваться различная цепочка Tor. Например, у приложения curl и у приложения wget будут разные цепочки Tor с Гейтвея. Сравним работу приложений

curl https://check.torproject.org/ | grep "IP"

Данная команда вернет нам наш IP, полученный на сайте check.torproject.org результат выполнения

Your IP address appears to be: **111.111.111.11**

А теперь wget

wget -q -O- https://check.torproject.org/ | grep "IP"

Результат выполнени

Your IP address appears to be: **100.000.000.000**

Как видите, одна и таже команда вернула нам разные IP.

В системе остались «оригиналы» от этих приложений, которые называются как <название приложения>.anondist-orig

То есть, есть приложение curl, его оригинал называется curl.anondist-orig
Для wget – wget.anondist-orig

Можно сравнить работу оригинального приложения и приложения

модифицированного на примерах curl и wget.

Подключимся к VPN на Workstation. Мой VPN IP: 47.100.100.100

Проверяем наш IP через curl

curl https://check.torproject.org/ | grep "IP"

Рузльтат выполнения команды таков

Your IP address appears to be: **51.100.100.100**

IP показывает Торовский. Не VPN. А теперь тоже самое, только с командой curl.anondist-orig

curl.anondist-orig https://check.torproject.org/ | grep "IP"

Выдает мой VPN IP

Your IP address appears to be: **47.100.100.100**

Как видите, curl и curl.anondist-orig отличаются. Теперь тоже самое с wget.

wget -q -O- https://check.torproject.org/ | grep "IP"

Выдает IP Тора

Your IP address appears to be: **51.100.100.100**

Теперь wget.anondist-org:

wget.anondist-orig -q -O- https://check.torproject.org/ | grep "IP"

Выдает IP VPN

Your IP address appears to be: **47.100.100.100**

VPN -> Tor -> Proxy (HTTP/SOCKS/и т. д.) -> Интернет (через браузер Firefox)

Допустим мы хотим чтобы у нас был белый IP и при этом у нас есть Proxy (HTTP/SOCKS).

VPN на хост. машине, как в первой цепочке, а прокси ставим в настройки браузера Firefox.

Открываем Firefox -> справа сверху жмем на иконку «бургер» -> жмем «Preferences» -> в появившимся окне переходим во вкладку «Advanced» -> там переходим во вкладку «Network» -> жмем на <Settings>.

Появилось окно с настройками Proxy. Радиобаттаном отмечаем «Manual proxy configuration». Далее, если у вас HTTP прокси, то забиваем их в поле «HTTP Proxy», указав IP и порт, и отмечаем галочкой «Use this proxy server for all protocols».

Если у вас SOCKS4/SOCKS5 прокси, то забиваем их в поле «SOCKS Host», указав IP и порт, при этом отметив нужную версию SOCKS радиобаттоном. При желании можно отметить галочкой «Remote DNS».

Все, настроили Firefox на работу через прокси. Теперь наш траффик VPN -> Tor -> Proxy -> Интернет в Firefox.

Проверяем IP – заходим на whoer.net -> если IP прокси, то все ок.

Я разобрал настройку работы через прокси на примере браузера Firefox. Пустить таким образом траффик можно и с другим приложением, идем в настройки и разбираемся. Желательно не трогать приложения из списка выше.

VPN -> Tor -> SSH-Tunnel (средствами ssh.anondist-orig) -> Интернет (работаем через браузер Firefox)

Вы хотите белый IP на выходе и у вас есть доступ по ssh к какому-то серверу.
Через ssh есть возможность пробросить порт и сделать локальный прокси и настраивать нужное вам приложение на работу уже через этот прокси.

Итак, поскольку uwt wrapped ssh не позволит нам пробросить порт, будет использоваться ssh.anondit-orig (см. выше про uwt

wrapped).

ssh.anondist-orig -D 127.0.0.1:port username@ip

Где port – это любое число от 1024 до 65535. Желательно использовать какой-нибудь не стандартный (aka «экзотичекий») порт.

username – логин от сервера

ip – IP адрес сервера

Пример, в моем случае я ввожу

ssh.anondist-orig -D 127.0.0.1:17665 login@100.100.100.100

Ввели команду – вводим пароль и все. Локальная прокся создалась, при этом мы подключились по ssh к серверу. Теперь, для того что бы работать в Firefox через этот созданный прокси (наш ssh), проделываем следующие действия, как из раздела выше.

Открываем Firefox -> справа сверху жмем на иконку «бургер» -> жмем «Preferences» -> в появившимся окне переходим во вкладку «Advanced» -> там переходим во вкладку «Network» -> жмем на «Settings».

Появилось окно с настройками Proxy. Радиобаттаном отмечаем «Manual proxy configuration». В поле «SOCKS Host» забиваем ip «127.0.0.1», в поле «Port» вводим port (в моему случ. это 17665). При желании можно отметить галочкой «Remote DNS».

Все, настроили Firefox на работу через прокси, которую получили при помощи ssh. Теперь наш траффик VPN -> Tor -> SSH-Tunnel -> Интернет в Firefox. Проверяем IP – заходим на whoer.net -> если IP прокси, то все ок.

Пустить таким образом траффик можно и с другим приложением, идем в настройки нужного вам приложения и разбираемся. Желательно не трогать приложения из списка выше.

Проверяем IPшник курлом

```
curl.anondist-orig --socks5 127.0.0.1:17665 https://check.torproject.org/ | grep "IP"
```

Для того чтобы закрыть ssh соеденение – вводим в терминале

```
exit
```

VPN -> Tor -> SSH-Tunnel (средствами sshuttle) -> Интернет (весь траффик туннелируется)

Есть прекрасная программа, называется sshuttle. Думаю, не многие про нее слышали.

Репозиторий разработчика на Github

Оф. документация sshuttle

Это что-то вроде VPN, только работает с ssh. Не совсем VPN, но и не совсем обычный проброс портов. Программа заворачивает весь ваш IPv4 траффик в ssh-туннель. Траффик всех приложений. Делает она это при помощи правил в iptables.

При этом не требует админский прав на сервере, к которому есть доступ по ssh. На сервере должен быть установлен Python 2.7 или Python 3.5. В 98% случаев Python на серверах стоит.

Также программа поддерживает туннелирование DNS траффика, для этого добавляется опция –dns.

Поддерживает Linux и Mac OS X. Винду официально не поддерживает.

Если вы используете sshuttle для туннелирования всего траффика, то Top Браузер и uwt wrapped приложения из списка выше (curl, wget, ssh, git, apt-get и т. д.) не будет работать, как в случае с VPN (2) (где они его обходили и получалась цепочка VPN -> Tor -> Интернет).

Потому что sshuttle создает правила в iptables.

Остальные приложения функционируют нормально, в том числе и браузер Firefox, о примере которого дальше пойдет речь.

Если вы используете sshuttle и вам вдруг приспичило заюзать curl/git/apt-get или что-то еще из uwt wrapped приложений, то используйте их аналоги, которые я рассматривал выше (curl.anondist-orig, git.anondist-orig, apt-get.anondist-orig и т. д.).

Устанавливаем sshuttle

sudo apt-get install sshuttle

После установки ничего не нужно дополнительно конфигурировать. Запускается все следующим образом

sudo sshuttle -r username@ip:port 0.0.0.0/0 -vv

где username – логин к ssh серверу

ip – IP адрес сервера

port – порт, по которому доступен ssh на сервере, по-умолчанию это 22, если не менялся.

опция -vv – verbose mode.

Вводим пароль, сначала от user на Whonix, затем от пользователя на сервере.Теперь весь наш IPv4 траффик туннелирован.

Проверяем. Открываем Firefox, заходим на whoer.net, смотрим свой IP адрес. Радуемся. При этом, конечно же, в настройках прокси в Firefox, радиобаттоном должно быть отмечено «No proxy».

Для туннелирования еще и dns траффика, будет следующая команда

sudo sshuttle --dns -r username@ip:port 0.0.0.0/0 -vv

Проверяем IPшник курлом

curl.anondist-orig https://check.torproject.org/ | grep "IP"

VPN -> Tor -> Proxy (HTTP/SOCKS)/SSH-Tunnel -> Интернет в Tor Браузере (использование аддона FoxyProxy).

Для браузера Firefox, а соответственно и для Tor Браузера тоже (потому что Tor Браузер основан на Firefox) существует хороший аддон для работы с проксями – FoxyProxy Standart. FoxyProxy Standart по своему функционалу полностью заменяет стандартную фичу с настройками проксей в браузерах Firefox/Tor Browser.

Поскольку в настройках проксей в Tor Браузере уже прописано SOCKS5 127.0.0.1:9150, что не позволяет нам написать туда что-либо еще, то мы будем использовать этот аддон.

Устанавливаем аддон. Для начала идем по ссылке. Там жмем на синюю кнопку -> на сл. стр. жмем -> во всплывающем окне жмем -> после этого перезапускаем Tor Браузер.

Аддон позволяет добавлять сразу несколько проксей и менять их в пару нажатий, что довольно удобно.

Настраиваем прокси. В Tor Браузере, справа от адресной строки, появилась иконка «Лисы». Жмем на нее -> попадаем в окно с настройками. Интерфейс окна с настройками интуитивно понятен, но я опишу как настроить прокси. Итак, в окне жмем. Теперь отмечаем кружочком (радиобаттон) «Manual Proxy Configuration».

Напротив поля «Host or IP Address» вводим ip прокси. Напротив Port – порт. Если прокси SOCKS, отмечаем это галочкой, и выбираем версию (SOCKS5 или SOCKS4). При этом, если прокси у вас HTTP/HTTPS, так же есть возможность использовать Аутентификацию (чего нельзя в стандартных настройках).

Использовать Аутентификацию, если у вас SOCKS, к сожалению, нельзя (используйте proxychains или пробрасывайте порты с ssh).

В случае, если у вас уже есть Proxy (HTTP/SOCKS), то просто заполняем все поля, и жмем OK. В случае, если у вас есть

доступ по ssh к серверу, то сначала пробрасываем порт (делаем локальный прокси, смотрите реализацию цепочки «VPN -> Tor -> SSH-Tunnel (средствами ssh.anondist-orig) -> Интернет»), затем заполняем все поля.

Для того чтобы включить нужный вам прокси (а добавить вы могли несколько), нужно: нажать ПКМ на иконку с «лисой» -> выбрать «Use Proxy <прокси, который вы добавили> for all URLs». Иконка лисы станет другово цвета. Все будет работать, если ваш прокси не дохлый.

Проверяем, идем на whoer.net смотрим IP.

Congratulations. This browser is configured to use Proxy.

Для того чтобы отключить: ПКМ на иконку с «лисой», выбираем вариант «Completely Disable FoxyProxy»

VPN (1) -> Tor -> VPN (2) -> Proxy(HTTP/SOCKS)/SSH-Tunnel -> Интернет

Комбинируем следующие цепочки: «VPN (1) -> Tor -> VPN (2) -> Интнернет» и «VPN (1) -> Tor -> Proxy(HTTP/SOCKS) -> Интернет» или «VPN -> Tor -> SSH-Tunnel (средствами ssh.anondist-orig) -> Интернет»

VPN(1) – на хост. машину. VPN (2) – на Workstation, точно также, как делали цепочку «VPN(1) -> Tor -> VPN(2) -> Интернет».

Далее, настраиваем Firefox на работу через Proxy/SOCKS встроенными средствами в Firefox, как в разобранных цепочках выше («VPN -> Tor -> Proxy(HTTP/SOCKS/и т. д.) -> Интернет»). Это если у вас уже есть прокси/socks.

А если у вас есть доступ по ssh к серверу, то создаем локальный прокси и настраиваем на работу через него нужное вам приложение (выше было разобрано на примере Firefox, в цепочке «VPN -> Tor -> SSH-Tunnel (средствами ssh.anondist-orig) -> Интернет»).

В кратце, все выглядит примерно так:

1) VPN (1) цепляем на вашу хост. машину

2) VPN (2) ставим на Workstation.

3) Если у вас есть Proxy/SOCKS, то настраеваем работу приложение через них.

4) Если у вас есть доступ по ssh, то делаете локальный прокси, и настраиваете работу нужного приложния через него.

VPN (1) -> Tor -> VPN (2) -> SSH-Tunnel (sshuttle) -> Интернет.

Да, это тоже будет работать. На выходе будет IP с ssh. Комбинируем следующие цепочки: VPN (1) -> Tor -> VPN (2) -> Интнернет» и «VPN -> Tor -> SSH-Tunnel (средствами sshuttle) -> Интернет. Читайте выше.

Все выглядит примерно так:

1) VPN (1) цепляем на вашу хост. машину

2) VPN (2) ставим на Workstation (читайте выше).

3) Используем sshuttle. Прочитать вы можете выше («VPN -> Tor -> SSH-Tunnel (средствами sshuttle) -> Интернет»).

4) Работаем. При этом, как и было описано выше в разделе про sshuttle, не будут работать uwt wrapped приложения (можно заменить) и Тор Браузер в частности. Остальные приложения работают нормально, в том числе и браузер Firefox.

Proxy (HTTP/SOCKS) -> Tor -> Интернет

Здесь мы работаем на Whonix Gateway, смотрите не перепутайте виртуалки.

Если у вас вдруг нет VPN, который вы могли бы прицепить на хост. машину, то еще одним способом скрыть факт использования

Тор от провайдера является использование проксей и прописывание их в основной конфигурационный файл Тор (/etc/tor/torrc) на Whonix Gateway. Для реализации подобной цепочки, нам необходимы прокси HTTP/HTTPS/SOCKS4/SOCKS5. Подойдут прокси с аутентификацией. Будем работать на Gateway, копаться в кофигурационном файле /etc/tor/torrc.

Итак, для того чтобы наш траффик шел сначала через прокси, а затем через Тор, нам нужно в конфигурационном файле /etc/tor/torrc на Gateway прописать работу через прокси.

Как это сделать. Для начала на всякий случай сделаем бекап файла

```
sudo cp /etc/tor/torrc ~/torrc.orig
```

Теперь открываем /etc/tor/torrc

С ярлыка на Gateway (Tor User Config), или вводим

```
sudo nano /etc/tor/torrc
```

Первоначальный вид конфиг файла

```
# This file is part of Whonix
# Copyright (C) 2012 - 2013 adrelanos
# See the file COPYING for copying conditions.

# Use this file for your user customizations.
# Please see /etc/tor/torrc.examples for help, options, comments etc.

# Anything here will override Whonix's own Tor config customizations in
# /usr/share/tor/tor-service-defaults-torrc

# Enable Tor through whonixsetup or manually uncomment "DisableNetwork 0" by
# removing the # in front of it.
DisableNetwork 0
```

Теперь, в зависимости от того, какого у вас типа прокси, нам

нужно добавить одни из следующихъ строк.

SOCKS5 прокси

Если у вас прокси SOCKS5, то в конфиг. файл нужно добавить

Socks5Proxy host[:port]

host – ip или hostname сервера

port – порт, на котором слушает socks5

Пример

Socks5Proxy 185.100.100.100:12759

Если у вас SOCKS5 с аутентификацией, то добавить нужно эти три строки, строго в такой последовательности

Socks5Proxy host[:port]
Socks5ProxyUsername username
Socks5ProxyPassword password

username – ваш username для аутентификации SOCKS5

password – ваш пароль для аутентификации SOCKS5

Пример

Socks5Proxy 185.100.100.100:12759
Socks5ProxyUsername pprunion
Socks5ProxyPassword superPPrunion111

Таким образом, конфиг файл пришел к виду

```
# This file is part of Whonix
# Copyright (C) 2012 - 2013 adrelanos
# See the file COPYING for copying conditions.

# Use this file for your user customizations.
# Please see /etc/tor/torrc.examples for help, options, comments etc.

# Anything here will override Whonix's own Tor config
```

```
customizations in
# /usr/share/tor/tor-service-defaults-torrc

# Enable Tor through whonixsetup or manually uncomment
"DisableNetwork 0" by
# removing the # in front of it.
DisableNetwork 0
Socks5Proxy 185.100.100.100:12759
Socks5ProxyUsername pprunion
Socks5ProxyPassword superPPrunion111
```

Сохраняем конфиг файл -> перезапускаем Tor

```
sudo service tor@default restart
```

Если интернет на WS работает, значит все OK. Если нет, значит вы где-то напортачили или ваша прокся дохлая.

SOCKS4 прокси

Если у вас прокси SOCKS4, то в конфиг. файл нужно добавить

```
Socks4Proxy host[:port]
```

host – ip или hostname сервера

port – порт, на котором слушает socks4

Пример

```
Socks4Proxy 185.100.100.100:12759
```

SOCKS4 прокси не поддерживают аутентификацию.

Конфиг файл пришел к виду

```
# This file is part of Whonix
# Copyright (C) 2012 - 2013 adrelanos
# See the file COPYING for copying conditions.

# Use this file for your user customizations.
# Please see /etc/tor/torrc.examples for help, options,
comments etc.
```

```
# Anything here will override Whonix's own Tor config customizations in
# /usr/share/tor/tor-service-defaults-torrc

# Enable Tor through whonixsetup or manually uncomment "DisableNetwork 0" by
# removing the # in front of it.
DisableNetwork 0
Socks4Proxy 185.100.100.100:12759
```

Сохраняем конфиг файл -> перезапускаем Tor

```
sudo service tor@default restart
```

Если интернет на WS работает, значит все OK. Если нет, значит вы где-то напортачили или ваша прокся дохлая.

HTTP прокси

Если у вас прокси HTTP, то в конфиг. файл нужно добавить

```
HTTPProxy host[:port]
```

Где host – ip или hostname сервера

port – порт, на котором слушает прокси http

Пример

```
HTTPProxy 185.100.100.100:12759
```

Если у вас HTTP прокси с аутентификацией, то нужно добавить следующую строку

```
HTTPProxyAuthenticator username:password
```

username – ваш username от HTTP проксей

password – ваш пароль от HTTP проксей

Пример

```
HTTPProxy 185.100.100.100:12759
HTTPProxyAuthenticator pprunion:superHTTPproxyPP111
```

Таким образом, конфиг файл пришел к виду

```
# This file is part of Whonix
# Copyright (C) 2012 - 2013 adrelanos
# See the file COPYING for copying conditions.

# Use this file for your user customizations.
# Please see /etc/tor/torrc.examples for help, options, comments etc.

# Anything here will override Whonix's own Tor config customizations in
# /usr/share/tor/tor-service-defaults-torrc

# Enable Tor through whonixsetup or manually uncomment "DisableNetwork 0" by
# removing the # in front of it.
DisableNetwork 0
HTTPProxy 185.187.113.110:12759
HTTPProxyAuthenticator pprunion:superHTTPproxyPP111
```

Сохраняем конфиг файл -> перезапускаем Tor

```
sudo service tor@default restart
```

Если интернет на WS работает, значит все OK. Если нет, значит вы где-то напортачили или ваша прокся дохлая.

HTTPS прокси

Если у вас прокси HTTPS, то в конфиг файл нужно добавить

```
HTTPSProxy host[:port]
```

host – ip или hostname сервера

port – порт, на котором слушает прокси http

Пример

```
HTTPSProxy 185.187.113.110:12759
```

Если у вас HTTPS прокси с аутентификацией, то нужно добавить

следующую строку

HTTPSProxyAuthenticator username:password

username – ваш username от HTTPS проксей

password – ваш пароль от HTTPS проксей

Пример

HTTPSProxy 185.187.113.110:12759
HTTPSProxyAuthenticator pprunion:superHTTPproxyPP111

username – ваш username от HTTPS проксей

password – ваш пароль от HTTPS проксей

Пример

HTTPSProxy 185.187.113.110:12759
HTTPSProxyAuthenticator pprunion:superHTTPproxyPP111

Таким образом, конфиг файл пришел к виду

```
# This file is part of Whonix
# Copyright (C) 2012 - 2013 adrelanos
# See the file COPYING for copying conditions.

# Use this file for your user customizations.
# Please see /etc/tor/torrc.examples for help, options, comments etc.

# Anything here will override Whonix's own Tor config customizations in
# /usr/share/tor/tor-service-defaults-torrc

# Enable Tor through whonixsetup or manually uncomment "DisableNetwork 0" by
# removing the # in front of it.
DisableNetwork 0
HTTPProxy 185.187.113.110:12759
HTTPProxyAuthenticator pprunion:superHTTPproxyPP111
```

Сохраняем конфиг файл -> перезаписукаем Tor

sudo service tor@default restart

Если интернет на WS работает, значит все OK. Если нет, значит вы где-то напортачили или ваша прокся дохлая.

В итоге, после того как мы пропиали нужную вам прокси и перезапустили Tor, теперь наш траффик идет сначала через указанный прокси, а затем через Tor.

Если вы где-то что-то напортачили, сделали не так, или сделали так, что все не работает, то восстанавливаем оригинальный torrc из созданного ранее бекапа

sudo cp ~/torrc.orig /etc/tor/torrc

Оригинальный файл восстановлен.

VPN (1) -> Proxy (HTTP/SOCKS) -> Tor -> Интернет

Комбинируем следующие цепочки Proxy (HTTP/SOCKS) -> Tor -> Интернет» и «VPN -> Tor -> Интернет»

Получается так:

1) Вы подключаете к VPN на своей хост машине

2) Вы прописываете нужный вам прокси в конфигурационный файл /etc/tor/torrc HA GATEWAY (!!!)

Цепочки прокси (использование Proxychains)

Proxychains – еще одна полезная программа для проксифицирования отдельного приложения.

Последнее обновление оригинального Proxychains вышло в 2006 году (proxychains v3.1). Но работает до сих пор.

На Github существует два форка proxychains, которые периодически обновляются:

1) github.com

2) github.com

По-умолчанию в Debian устанавливается оригинальный proxychains (в Арч, например, ставится proxychains-ng).

Но я рекомендую ставить именно proxychains-ng. Он лучше. Исправлены некоторые ошибки, по сравнению с оригиналом и все же разработка продолжается и поддерживается, проект не заброшен, как в случае с оригинальным proxychains. Именно на примере данного форка я буду разбирать работу.

Теперь перейдем к использованию. Последняя версия Proxychains-ng на момент написания статьи – v4.11. Чекайте обновления на Github разработчиков. Скачиваем Proxychains

```
wget -O ~/proxychains-ng.tar.bz2 https://github.com/rofl0r/proxychains-ng/releases/download/v4.11/proxychains-ng-4.11.tar.bz2
```

После скачивания, разархивируем

```
cd ~/; tar -xf ~/proxychains-ng.tar.bz2
```

Теперь собираем

```
cd ~/proxychains-ng/
sudo ./configure --prefix=/usr --sysconfdir=/etc
make
sudo make install
sudo make install-config
```

Конфигурационный файл довольно простой и находится по адресу /etc/proxychains.conf
Оригинал конфиг файла proxychains

```
# proxychains.conf  VER 4.x
#
#        HTTP, SOCKS4a, SOCKS5 tunneling proxifier with DNS.


# The option below identifies how the ProxyList is treated.
# only one option should be uncommented at time,
```

```
# otherwise the last appearing option will be accepted
#
dynamic_chain
#
# Dynamic - Each connection will be done via chained proxies
# all proxies chained in the order as they appear in the list
# at least one proxy must be online to play in chain
# (dead proxies are skipped)
# otherwise EINTR is returned to the app
#
#strict_chain
#
# Strict - Each connection will be done via chained proxies
# all proxies chained in the order as they appear in the list
# all proxies must be online to play in chain
# otherwise EINTR is returned to the app
#
#round_robin_chain
#
# Round Robin - Each connection will be done via chained
proxies
# of chain_len length
# all proxies chained in the order as they appear in the list
# at least one proxy must be online to play in chain
# (dead proxies are skipped).
# the start of the current proxy chain is the proxy after the
last
# proxy in the previously invoked proxy chain.
# if the end of the proxy chain is reached while looking for
proxies
# start at the beginning again.
# otherwise EINTR is returned to the app
# These semantics are not guaranteed in a multithreaded
environment.
#
#random_chain
#
# Random - Each connection will be done via random proxy
# (or proxy chain, see  chain_len) from the list.
# this option is good to test your IDS :)
```

```
# Make sense only if random_chain or round_robin_chain
#chain_len = 2

# Quiet mode (no output from library)
#quiet_mode

# Proxy DNS requests - no leak for DNS data
proxy_dns

# set the class A subnet number to use for the internal remote
DNS mapping
# we use the reserved 224.x.x.x range by default,
# if the proxified app does a DNS request, we will return an
IP from that range.
# on further accesses to this ip we will send the saved DNS
name to the proxy.
# in case some control-freak app checks the returned ip, and
denies to
# connect, you can use another subnet, e.g. 10.x.x.x or
127.x.x.x.
# of course you should make sure that the proxified app does
not need
# *real* access to this subnet.
# i.e. dont use the same subnet then in the localnet section
#remote_dns_subnet 127
#remote_dns_subnet 10
remote_dns_subnet 224

# Some timeouts in milliseconds
tcp_read_time_out 15000
tcp_connect_time_out 8000

### Examples for localnet exclusion
## localnet ranges will *not* use a proxy to connect.
## Exclude connections to 192.168.1.0/24 with port 80
# localnet 192.168.1.0:80/255.255.255.0

## Exclude connections to 192.168.100.0/24
# localnet 192.168.100.0/255.255.255.0

## Exclude connections to ANYwhere with port 80
```

```
# localnet 0.0.0.0:80/0.0.0.0

## RFC5735 Loopback address range
## if you enable this, you have to make sure remote_dns_subnet
is not 127
## you'll need to enable it if you want to use an application
that
## connects to localhost.
# localnet 127.0.0.0/255.0.0.0

## RFC1918 Private Address Ranges
# localnet 10.0.0.0/255.0.0.0
# localnet 172.16.0.0/255.240.0.0
# localnet 192.168.0.0/255.255.0.0

# ProxyList format
#       type  ip  port [user pass]
#       (values separated by 'tab' or 'blank')
#
#       only numeric ipv4 addresses are valid
#
#
#        Examples:
#
#               socks5        192.168.67.78        1080
lamer        secret
#                http         192.168.89.3         8080
justu        hidden
#              socks4        192.168.1.49        1080
#               http        192.168.39.93        8080
#
#
#       proxy types: http, socks4, socks5
#         ( auth types supported: "basic"-http  "user/pass"-
socks )
#
[ProxyList]
# add proxy here ...
# meanwile
# defaults set to "tor"
socks4  127.0.0.1 9050
```

dynamic_chain – будет перебирать прокси в том порядке, как вы их задали, при этом пропуская мертвые.

strict_chain – будет делать цепочку прокси в том порядке, как вы их задали, при этом все прокси должны быть живые. Если будет одна мертвая прокся, то программа не сможет выполниться.

proxy_dns – проксифицирует dns.

chain_len – указываете длину цепочки.

Теперь про добавление прокси. Дальше, после надписи «[ProxyList]» идут прокси. Одна прокся – одна строка. Чуть выше в конфиг файле есть примеры прописывания разных проксей

```
Examples:
socks5  192.168.67.78        1080        lamer        secret
//socks5 с аутентификацией
socks5  192.168.67.78   1080                    // просто
socks5
http    192.168.89.3        8080        justu        hidden
//httpc аутентификацией
socks4  192.168.1.49        1080                        //
socks4
http    192.168.39.93        8080                        //
просто http
```

По-умолчанию проксичейнс сконфигурирован на работу через Tor, но учитывая специфику данного дистрибутива, Вам нужно обязательно удалить или закомментировать (символ «#» в начале строки) следующую строчку

```
socks4 127.0.0.1 9050
```

Я добавил свои socks5 с аутентификацией в конфиг файл (последняя строка), и теперь он выглядет так

```
# proxychains.conf  VER 4.x
#
#        HTTP, SOCKS4a, SOCKS5 tunneling proxifier with DNS.
```

```
# The option below identifies how the ProxyList is treated.
# only one option should be uncommented at time,
# otherwise the last appearing option will be accepted
#
dynamic_chain
#
# Dynamic - Each connection will be done via chained proxies
# all proxies chained in the order as they appear in the list
# at least one proxy must be online to play in chain
# (dead proxies are skipped)
# otherwise EINTR is returned to the app
#
#strict_chain
#
# Strict - Each connection will be done via chained proxies
# all proxies chained in the order as they appear in the list
# all proxies must be online to play in chain
# otherwise EINTR is returned to the app
#
#round_robin_chain
#
# Round Robin - Each connection will be done via chained
proxies
# of chain_len length
# all proxies chained in the order as they appear in the list
# at least one proxy must be online to play in chain
# (dead proxies are skipped).
# the start of the current proxy chain is the proxy after the
last
# proxy in the previously invoked proxy chain.
# if the end of the proxy chain is reached while looking for
proxies
# start at the beginning again.
# otherwise EINTR is returned to the app
# These semantics are not guaranteed in a multithreaded
environment.
#
#random_chain
#
# Random - Each connection will be done via random proxy
# (or proxy chain, see  chain_len) from the list.
```

```
# this option is good to test your IDS :)

# Make sense only if random_chain or round_robin_chain
#chain_len = 2

# Quiet mode (no output from library)
#quiet_mode

# Proxy DNS requests - no leak for DNS data
proxy_dns

# set the class A subnet number to use for the internal remote DNS mapping
# we use the reserved 224.x.x.x range by default,
# if the proxified app does a DNS request, we will return an IP from that range.
# on further accesses to this ip we will send the saved DNS name to the proxy.
# in case some control-freak app checks the returned ip, and denies to
# connect, you can use another subnet, e.g. 10.x.x.x or 127.x.x.x.
# of course you should make sure that the proxified app does not need
# *real* access to this subnet.
# i.e. dont use the same subnet then in the localnet section
#remote_dns_subnet 127
#remote_dns_subnet 10
remote_dns_subnet 224

# Some timeouts in milliseconds
tcp_read_time_out 15000
tcp_connect_time_out 8000

### Examples for localnet exclusion
## localnet ranges will *not* use a proxy to connect.
## Exclude connections to 192.168.1.0/24 with port 80
# localnet 192.168.1.0:80/255.255.255.0

## Exclude connections to 192.168.100.0/24
# localnet 192.168.100.0/255.255.255.0
```

```
## Exclude connections to ANYwhere with port 80
# localnet 0.0.0.0:80/0.0.0.0

## RFC5735 Loopback address range
## if you enable this, you have to make sure remote_dns_subnet
is not 127
## you'll need to enable it if you want to use an application
that
## connects to localhost.
# localnet 127.0.0.0/255.0.0.0

## RFC1918 Private Address Ranges
# localnet 10.0.0.0/255.0.0.0
# localnet 172.16.0.0/255.240.0.0
# localnet 192.168.0.0/255.255.0.0

# ProxyList format
#       type  ip  port [user pass]
#       (values separated by 'tab' or 'blank')
#
#       only numeric ipv4 addresses are valid
#
#
#        Examples:
#
#               socks5        192.168.67.78        1080
lamer        secret
#                http         192.168.89.3         8080
justu        hidden
#               socks4        192.168.1.49        1080
#              http        192.168.39.93        8080
#
#
#       proxy types: http, socks4, socks5
#         ( auth types supported: "basic"-http  "user/pass"-
socks )
#
[ProxyList]
# add proxy here ...
# meanwile
# defaults set to "tor"
```

```
#socks4 127.0.0.1 9050
socks5 185.187.113.110:12759 pprunion superPPrunion111
```

Теперь о том, как работает proxychains. Если вы ставили proxychains-ng, то команда proxychains4, если ставили оригинальный proxychains, то команда proxychains.

Работает proxychains следующим образом

proxychains4 <приложение> <аргументы>

Запускаем firefox через proxychains

```
proxychains4 firefox
```

Заходим на whoer.net, смотрим IP. Запускаем curl через proxychains, грепаем IPшник с сайта torproject

```
proxychains4 curl.anondist-orig https://check.torproject.org/ | grep "IP"
```

Делаем работу более комфортабельной

Существует некоторая проблема при запуске графических приложений из под root (или через sudo), из терминала. Будет ошибка, что-то вроде Cannot open DISPLAY :0 или подобная. (GTK WARNING) Почему это нас касается? Потому что такие графические текстовые редакторы, как kwrite или mousepad не будут открываться через sudo, соответственно вы не сможете отредактировать файл, к которому у вас не прав на запись. Я нашел два следующих решения проблемы.

1) С использованием gksu/gksudo.

Здесь все просто, мы будем использовать специальну утилиту gksu — это графический фронтенд для su и sudo. Ставим пакет gksu

```
sudo apt-get install gksu
```

Для того чтобы открыть какой-либо приложение от пользователя

root из терминала, нужно из под обычного пользователя user в терминале ввести

gksudo <название приложения> <аргументы приложения>

И ввести пароль от пользователя user

gksu <название приложения> <аргументы приложения>

И ввести пароль от пользователя root

Вводим пароль от user. Откроет файл в редакторе kwrite.

gksudo kwrite /boot/grub/grub.cfg

Вводим пароль от user. Откроет файл в редакторе mousepad (если стоит).

gksudo mousepad /boot/grub/grub.cfg

Вводим пароль от root. Откроет файл в редакторе kwrite.

gksu kwrite /boot/grub/grub.cfg

Вводим пароль от root. Откроет файл в редакторе mousepad (если стоит).

gksu mousepad /boot/grub/grub.cfg

Вводим пароль от user. Откроет файловый менеджер Thunar (если стоит).

gksudo thunar

Вводим пароль от user. Откроет файловый менеджер Dolphin.

gksudo dolphin

Вводим пароль от root. Откроет файловый менеджер Thunar (если стоит).

gksu thunar

Вводим пароль от root. Откроет файловый менеджер Dolphin.

gksu dolphin

При этом, после замены DE на xfce4, некоторые приложения, которые остались от KDE, открываются таким способом некорректно.

2) С копированием .Xauthority в домашнюю папку root (/root). xhost +.

Для того чтобы наши графические приложения запускались из под root, из терминала и при этом не было ошибок, проделываем следующие действия. Входим в работу из под root

sudo -i

При необходимости вводим пароль. Далее, копируем файл ~/.Xauthority в домашнюю директорию пользователя root

cp /home/user/.Xauthority /root/

Выходим из под root, вводим

exit

Работает ровно на текущую сессию. В след. сессии придется проделывать тоже самое. Поэтому учитесь редактировать файлы в консольных редакторах. Все, ошибка больше не будет выскакивать. Но если по какой-либо причине это продолжает происходить, то каждую сессию вводим в терминале (из под пользователя user)

xhost +

Данная команда должна исправить ошибку.

Работа с nano

Лучше работайте с этим редактором.

Вкратце расскажу основные горячие клавиши при работе с консольным редактором nano. Открываем файл в nano.

nano <путь до файла>

Ctrl+O – сохранить. Записать изменения в файл. Жмем Ctrl+O, потом Enter.

Ctrl+X – выйти из nano.

Ctrl+K – вырезать целую строку. Будет скопировано в буфер обмена.

Ctrl+U – вставить вырезанную строку, которая находится в буфере обмена.

Ctrl+W – поиск по файлу. Поиск в редакторе, вводим что нужно -> жмем интер.

Если хотим скопировать часть текста, в самом редакторе выделяем мышкой нужный текст и жмем комбинацию Ctrl+Shift+C. Это скопировать. Для того того чтобы вставить текст, который сейчас находится в буфере обмена, жмем комбинацию Ctrl+Shift+V.

Не забывайте вводить

sudo nano

Если редактируете файл, к которому у вас нет прав на запись.

Ускоряем запуск виртуалок

Покопавшись в некоторых конфигах, можно ускорить запуск виртуалок секунд на 12 или даже больше. Данную операцию можно проделать на обеих виртуалках. Сначала отключим или уменьшим таймаут GRUB. Для редактирования файлов я буду использовать nano, вы можете использовать kwrite. Редактируем файл /etc/default/grub через nano

sudo nano /etc/default/grub

В самом верху ищем строчку GRUB_TIMEOUT и меняем значение «5» на что-то поменьше, например «2» или «1». Мне таймаут не нужен, я его вовсе отключу и поставлю «0». Отредактировать файл также можно через графический текстовый редактор, по

умолчанию в Whonix стоит kwrite. (смотрите выше про запуск граф. приложений и решение проблемы с ошибкой).

sudo kwrite /etc/default/grub

Меняем значение -> сохраняем через Ctrl+S -> закрываем. Переконфигурируем GRUB

sudo grub-mkconfig -o /boot/grub/grub.cfg

Убираем 10 секундную задержку. Данную операцию можно проделать на обеих виртуалках. Еще способ ускорить запуск – убрать или уменьшить 10 секундную задержку при загрузке виртуалок. При запуске Whonix, существует 10 секндная задержка для того, чтобы вы могли успеть нажать Ctrl+C, тем самым дать Хуниксу команду автоматически не запускать Login Manager (по-умолчанию kdm). Нужно отредактировать файл /etc/rads.d/30_default.conf, но сначала сделаем backup

sudo cp /etc/rads.d/30_default.conf /etc/rads.d/30_default.conf.bak

Теперь редактируем

sudo nano /etc/rads.d/30_default.conf

Ищем строчку rads_wait_seconds -> меняем значение на поменьше, я поставлю «2» -> сохраняем -> выходим. Если вовсе хотите отключить эту задержку, то чуть выше есть строчка rads_wait со значением «1», меняем на «0», а строчку rads_wait_seconds закоментируем (в начале строки введите символ ‘#’) -> сохраняем -> выходим

Создаем общие папки (shared folders)

Допустим произошла ситуация, и нам нужно перекинуть файл/архив/и т. д. с хостовой машины, на нашу Workstation, или наоборот, с Workstation на хостовую машину. В данном случае можно воспользоваться файлообменниками, залить файл/архив на файлообменник, поставить пароль и скачать по ссылке на ту машину, на которую нам нужно. А после скачивания удалить файл.

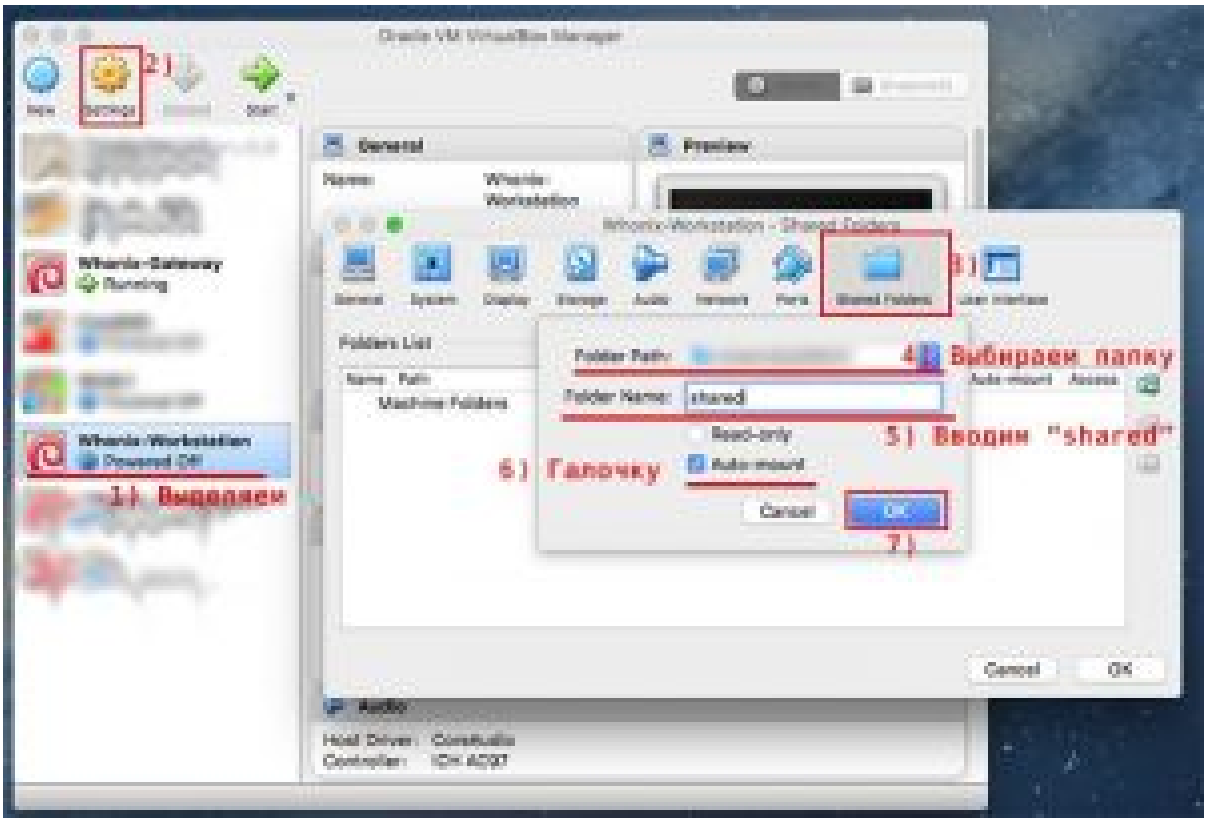
Или если файл с текстовым содержимым, то можно само содержимое залить на privnote.com, поставить пароль на скачивание и по ссылке перейти к содержимому.

Но можно воспользоваться так называемыми общими папками. Кому не понятно, схема простая. Существует папка на хост. машине и на Workstation, и если туда поместить какой-либо файл, то содержимое этих папок будет общим у обеих машин. Общие папки могут быть созданы на обеих виртуалках.

На хост машине создаем любую папку, в любом месте. Далее, если у вас включен Workstation – вырубаем его

```
sudo poweroff
```

Идем в виртуалбокс, открываем настройки Whonix-Workstation. Переходим во вкладку «Shared Foldes» -> жмем на иконку папочки с плюсом -> в первом поле указываем путь до созданной папки на хост. машине -> во второй главе обязательно вводим имя «shared» -> ставим галочку на Auto-Mount (Автомонтирование) -> остальные галки оставляем пустыми -> применяем настройки, жмем Ок -> включаем Whonix-Workstation.



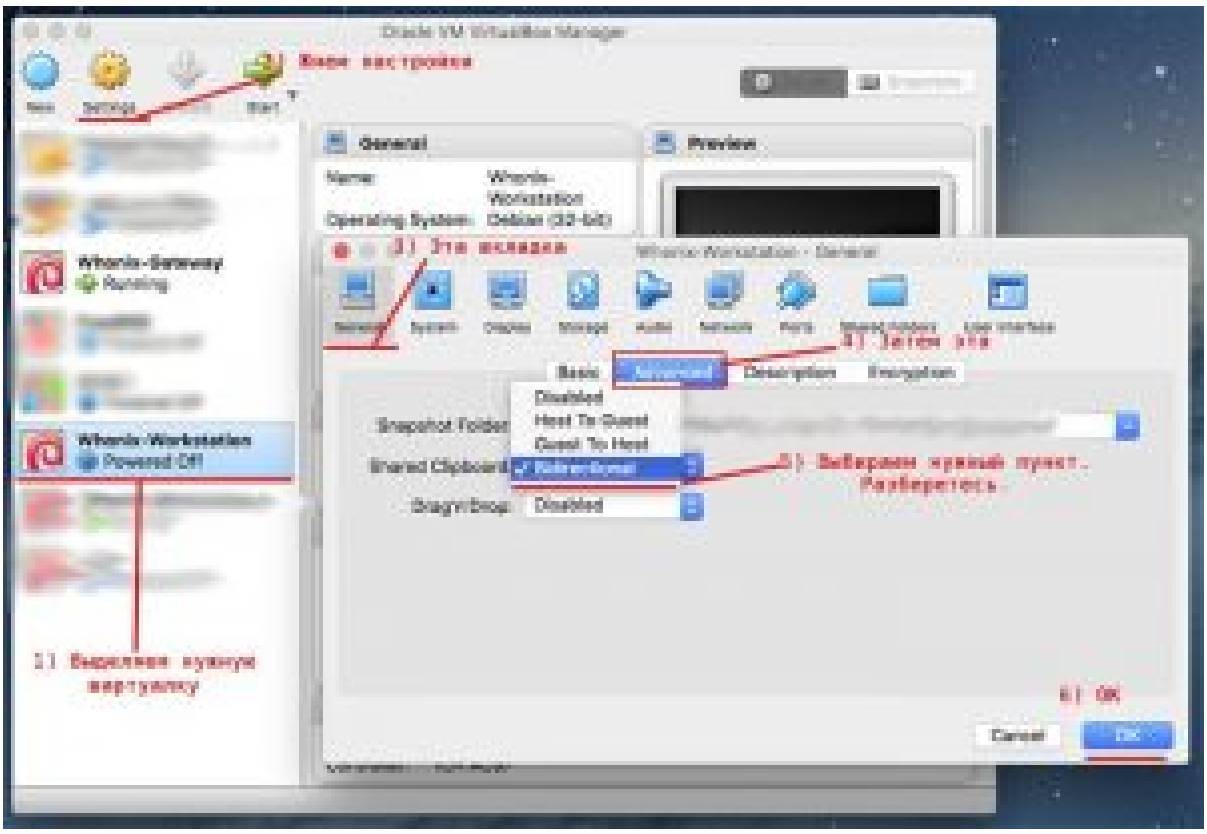
Общая папка на самом Workstation находится по пути /mnt/shared/. И теперь, если мы хотим получить файл с хост. машины, то просто помещаем его в папку, которую создавали и содержимое отобразится на Workstation в папке /mnt/shared/

После того как все перекинули, рекомендую отключить общие папки. В настройках Workstation, во вкладке «Shared Folders» -> выбираете вашу папку -> жмете на иконку папки с минусом -> применяете настройки.

Если создавать общую папку на Whonix-Gateway и выбрать ту же самую папку, которая является общей для хост машины и Workstation, то это будет работать и тогда у вас будет общая папка для трех машин.

Разрешаем копировать/вставлять с хост. машины на Workstation и наоборот

Сразу предупреждаю, что эту фичу лучше не включать из соображений безопасности. Но уж если очень приспичило, то: в VirtualBox открываем настройки Whonix-Workstation, в первой вкладке General (на рус. Общее) -> переходим в подвкладку Advanced -> далее первая кновпка с раскрывающимся списком (англ. «Shared Clipboard») -> выбираем последний пункт (англ. «Bidirectional»). Все, теперь можете копировать на хост. машине и вставлять на WS и наоборот. Повторюсь еще раз, лучше эту фичу не юзать, из соображений безопасности.



Скачиваем и ставим шрифты

По умолчанию в Debian дерьмовое отображение шрифтов. Ставим шрифт Droid Sans

```
sudo apt-get install fonts-droid
```

Открываем System Settings -> Application Appearance -> вкладка Fonts. Меняем «DejaVu Sans» во всех строчках на «Droid Sans». Можно еще «Поиграться со шрифтами», но Droid Sans выглядит вполне прилично. В этой же вкладке в «Use anti-alising» ставим значение на «Enabled» -> Apply.

Еще можем поставить хорошую группу шрифтов Ubuntu Font Family. В репозиториях их нет, поэтому ставим вручную. Создаем временную папочку для шрифтов и переходим туда

```
mkdir ~/fonts; cd ~/fonts/
```

Скачиваем zip архив со шрифтами в текущую папку

```
wget -O ubuntu-font-family.zip http://font.ubuntu.com/download/ubuntu-font-family-0.83.zip
```

Для того чтобы вставить текст в терминале, вместо привычных ctrl+v используем ctrl+shift+v, скопировать из терминала аналогично. Шрифты можно скачать и через браузер, по ссылке: Ubuntu Font Family. Находясь в папке ~/fonts/, выполняем

```
unzip -d ubuntu-fonts ubuntu-font-family.zip
```

Открываем файловый менеджер, в нашем случае это Dolphin, идем в папку, куда сохранили zip архив со шрифтами. Кликаем дважды -> разархивируем в эту же папку -> идем в разархивированную папку. Нас интересуют все файлы с расширением «.ttf». Кликаем дважды на файл -> Открывается Font Manager -> жмем Insall -> Personal -> закрываем -> делаем все тоже самое с остальными файлами «.ttf». Но можно установить и проще. Открываем терминал, идем в папку со скаченными шрифтами

```
cd ~/fonts/ubuntu-fonts
```

Создаем папку для шрифтов ubuntu

```
mkdir -p ~/.fonts/ubuntu
```

Копируем все туда (вы должны находиться в папке со шрифтами, файлы с расширением «.ttf»)

```
cp *.ttf ~/.fonts/ubuntu
```

Выполняем в консоли

```
fc-cache
```

Теперь у нас стоит Ubuntu Font Family. Точно также идем в настройки шрифтов, и можем поменять шрифты на Ubuntu. Перезагружаем

```
sudo reboot
```

Для терминала я использую шрифт Hack, но вы можете использовать Ubuntu Mono. Ставим Hack. Идем в нашу папочку fonts

```
cd ~/fonts/
```

Можно скачать через браузер с сайта Шрифт Hack скачиваем шрифт

```
wget -O ~/fonts/hack-ttf.zip https://github.com/chrissimpkins/Hack/releases/download/v2.020/Hack-v2_020-ttf.zip
```

Разархивируем

```
unzip -d hackfont hack-ttf.zip
```

Ставим шрифт

```
mkdir -p ~/.fonts/hack; cp ~/fonts/hackfont/*.ttf ~/.fonts/hack/
fc-cache
```

Теперь стоит шрифт Hack

Улучшаем рендеринг шрифтов – ставим Infinality

Скачиваем deb-пакет Infinality в домашнюю папку

```
wget -O ~/infinality.deb https://launchpad.net/~no1wantdthisname/+archive/ubuntu/ppa/+files/fontconfig-infinality_20130104-0ubuntu0ppa1_all.deb
```

Устанавливаем

```
sudo dpkg -i ~/infinality.deb
```

Альтернативная установка Infinality, с добавлением репозиториев (если не получилось установить способом выше)

Как только пакет установился, выполняем

```
sudo bash /etc/fonts/infinality/infctl.sh setstyle
```

Выбираем вариант «3». После этого редактируем файл /etc/profile.d/infinality-settings.sh

```
sudo nano /etc/profile.d/infinality-settings.sh
```

Ищем строку «USE_STYLE=». Вместо значения «DEFAULT», можно поставить одно из значений, перечисленных чуть выше в файле. Я ставлю значение «UBUNTU». Получилось USE_STYLE=»UBUNTU» ->

сохраняю -> закрываю. Перезагружаем

sudo reboot

Меняем DE на xfce4 (не обязательно)

Вовсе не обязательно менять одну DE на другую. Я делаю это лично для себя, потому что с xfce4 мне комфортнее работать. Так что можете пропустить этот раздел. Не нравится KDE4? Давайте поменяем на xfce4. Под линукс существует множество DE, такие как KDE4/KDE5, GNOME3, xfce4, Unity, MATE, в том числе и тайловые – i3, bspwm, awesome и так далее. По сравнению с текущим KDE4 – xfce4 более легковесный, более быстрый, жрет меньше RAM. Именно поэтому я буду ставить его. Устанавливаем xfce4

sudo apt-get install xfce4 xfce4-goodies

Пока ставится – читайте. На момент написания статьи, по умолчанию ставится версия xfce 4.10, хотя последняя xfce 4.12. Это происходит из-за того, что используются репозитории Debian Stable. Пытался билдить xfce4 вручную, скачивая с оф. сайта – в итоге что-то ломается. Бросил эту затею. В итоге решил остаться на xfce4.10 Немножко поясню за Testing репозитории. Я менял репозитории с Stable на Testing -> делал apt-get upgrade/dist-upgrade, что-то шло не так, все ломалось кароче. Несколько раз пытался. Не получается. Я сделал вывод, что с Stable на Testing Debian репозитории так просто перейти на Whonix не получится.

Небольшая оговорочка: после установки xfce4 не удаляйте пакеты kde/kdm/plasma. Вообще старайтесь ничего не удаляйть из того, что не ставили сами. Почему? Потому что, во-первых whonixcheck вам потом постоянно орать будет, что «не хватает каких-то пакетов» (хотя этот Warning можно отключить), а во-вторых – хрен знает к чему это может привести. Просто не трогаем ничего из пакетов, которые шли при стандартной установке. Установка некоторых пакетов может также заменить некоторые Хуниксовские, поэтому из-за этого тоже может орать. Делаем все

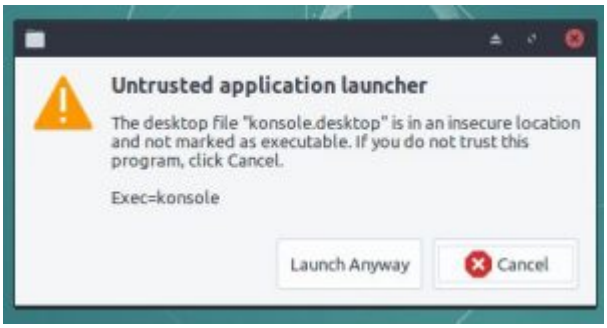
аккуратненько. Вообще, сносить Workstation и ставить по новой – может быть обыкновенным делом. Поставили xfce4? Перезагружаемся

```
sudo reboot
```

xfce4 должен загрузиться автоматически. При первом запуске вы увидите всплывающее окно, жмем s. Для того чтобы настроить панель жмем по ней ПКМ -> Panel Preferences. Тут настраиваем. После того, как вы поставили xfce4, некоторые стандартные приложения поменялись. Вместо текстового редактора kwrite, теперь по-умолчанию mousepad. Вместо файлового менеджера Dolphin, теперь Thunar. При этом старые приложения никуда не делись, вы также можете использовать и их.

Решаем проблему «Untrusted application launcher» на xfce4

После установки xfce4, у вас скорее всего будет проблема с запуском существующих приложений с ярлыков на раб. столе. При нажатии на ярлыке, всплывет окно «Untrusted application launcher», с кнопкой , при нажатию на которую, приложение все же запускается.



Происходит это потому, что существующие ярлыки (которые остались от KDE4) на раб. столе являются симлинками на другие ярлыки, которые находятся в папке /usr/share/applications/, у которых нет права на выполнение. Нужно дать права на запуск. Последовательно выполняем ниже приведенные команды

```
sudo chmod +x $(ls ~/Desktop/ | grep ".desktop" | sed -e 's/^/\/usr\/share\/applications\//' | tr '\n' ' ')
```

konsole и kgpg находятся в другой папке, поэтому вводим

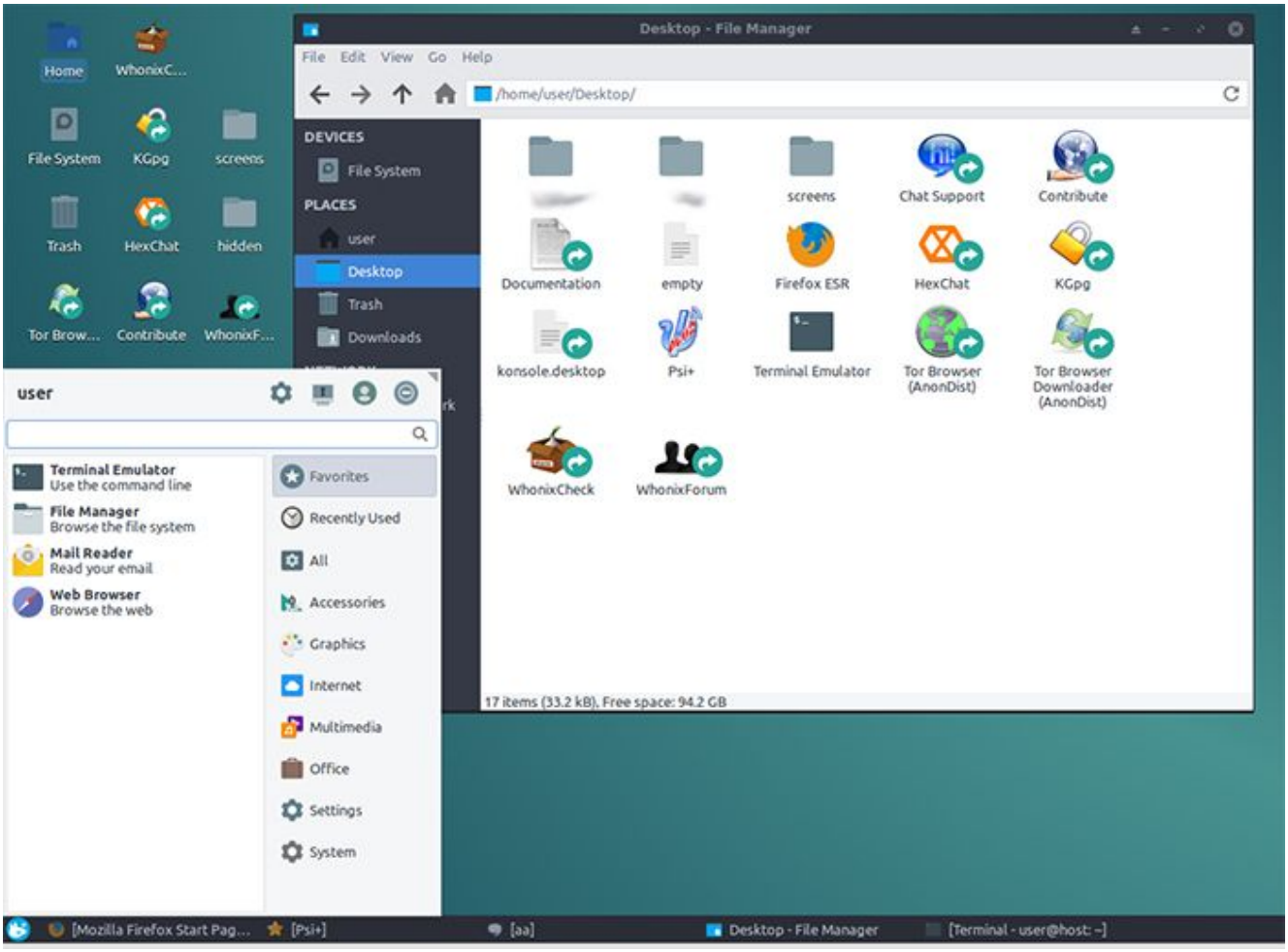
отдельно

```
sudo chmod +x /usr/share/applications/kde4/{kgpg.desktop,konsole.desktop}
```

Когда вы создаете новый ярлык, при первом запуске ярлыка это окно все равно будет появляться, но там будет кнопка, жмем ее и все будет четко.

Кастомизация

Этот раздел писался и работает только если у вас стоит DE xfce4, вместо KDE4 (см. выше установку xfce4). Если у вас KDE4, то забейте на этот раздел. В самом конце все будет выглядеть вот так



Рекомендую поставить «WhiskerMenu» – плагин для панели. И заменить им стандартный «Applications Menu».

```
sudo apt-get install xfce4-whiskermenu-plugin
```

Заменяем: Panel Preferences -> Items -> жмем «+» добавляем Whisker Menu -> двигаем его стрелочками -> убираем Applications Menu по нажатию на «-«. Поставим некоторые пакеты-зависимости

```
sudo apt-get install gtk2-engines-murrine gtk+-2.0 libgtk-3-dev gtk2-engines-xfce
```

Ставим тему Arc-Theme. Arc-Theme представлена в трех вариантах: Arc (светлая), Arc-Darker и Arc-Dark. Лично я использую светлую Arc. Репозиторий разработчиков [темы на github]

Создадим временную папочку для наших пакетов-тем

```
mkdir -p ~/themes
```

Ставим зависимости

```
sudo apt-get install gnome-themes-standard
```

Скачиваем deb-пакет

```
wget -O ~/themes/arc-theme.deb http://download.opensuse.org/repositories/home:/Horst3180/Debian_8.0/all/arc-theme_1480088096.9047b20_all.deb
```

Устанавливаем

```
sudo dpkg -i ~/themes/arc-theme.deb
```

Теперь стоит тема Arc. Применить ее можно в настройках: Setings Manager -> Appearance -> во вкладке Style выбираем Arc/Arc-Dark/Arc-Darker. Далее, Setings Manager -> Window Manager -> во вкладке Style выбираем Arc/Arc-Dark/Arc-Darker.

Ставим иконки paper. Репозиторий разработчиков иконок на github: [Paper Icons]

Скачиваем deb-пакет в папку ~/themes/

```
wget -O ~/themes/paper-src.tar.gz https://github.com/snwh/paper-icon-theme/archive/v1.3.4.tar.gz
```

Разархивируем

```
cd ~/themes/
```

```
mkdir paper-icon && tar -zxf paper-src.tar.gz -C paper-icon/ --strip-components 1
```

Собираем

```
cd paper-icon/
sudo bash autogen.sh
sudo make
sudo make install
```

Иконки paper установлены. Применяем: Setings Manager -> Appearance -> переходим на вкладку Icons -> выбираем Paper.