
i2pd Documentation

Выпуск 2.2.0

PurpleI2P team

окт. 07, 2020

1	Установка	3
2	Использование	5
3	Содержание:	7
3.1	Сборка на Unix системах	7
3.2	Сборка для Android	9
3.3	Примеры использования	11

i2pd (I2P Демон) это полнофункциональная реализация I2P клиента на языке C++.

- [Сайт](#)
- [GitHub](#)
- [Вики](#)
- [Багтрекер](#)
- [Twitter](#)

Самый простой способ установить i2pd это использование [бинарных файлов](#). Инструкции по сборке из исходников для вашей ОС смотрите в документации.

Использование

Смотрите в документацию и документированный конфигурационный файл.

3.1 Сборка на Unix системах

Этот документ описывает сборку на системах:

- *Debian/Ubuntu* (содержит инструкцию для сборки пакетов)
- *Fedora/Centos*
- *MAC OS X*
- *FreeBSD*

Убедитесь, что все зависимости в вашей системе удовлетворены.

Если это так, то приступаем к сборке i2pd. Клонировем репозиторий и собираем:

```
git clone https://github.com/PurpleI2P/i2pd.git
cd i2pd/build
cmake -DCMAKE_BUILD_TYPE=Release # есть больше опций, смотрите раздел "Опции CMake"
make                             # можно добавить опцию VERBOSE=1 для отладки
```

После сборки i2pd можно установить в систему следующей командой:

```
make install
```

Вы так же можете использовать упрощенный вариант сборки:

```
git clone https://github.com/PurpleI2P/i2pd.git
cd i2pd
make
```

3.1.1 Debian/Ubuntu

Устанавливаем компилятор и прочие программы для сборки:

```
sudo apt-get install build-essential
```

Устанавливаем библиотеки разработчиков для сборки:

```
sudo apt-get install \  
    libboost-chrono-dev \  
    libboost-date-time-dev \  
    libboost-filesystem-dev \  
    libboost-program-options-dev \  
    libboost-system-dev \  
    libboost-thread-dev \  
    libssl-dev
```

Если нужна поддержка UPnP (не забудьте потом запустить CMake с параметром WITH_UPNP=ON):

```
sudo apt-get install libminiupnpc-dev
```

Вы можете собрать пакет .deb следующим образом:

```
sudo apt-get install fakeroot devscripts  
cd i2pd  
debuild --no-tgz-check -b
```

3.1.2 Fedora/Centos

Устанавливаем компилятор и прочие программы для сборки:

```
sudo yum install make cmake gcc gcc-c++
```

В новых версиях Fedora используется [DNF](#) вместо YUM by по-умолчанию, можете использовать его

В Centos 7 есть только CMake 2.8.11 в официальных репозиториях, это слишком старая версия для i2pd. Нужен CMake версии 2.8.12 или новее.

Вы можете собрать CMake для Centos вручную (ВНИМАНИЕ, там много зависимостей для сборки):

```
wget https://kojipkgs.fedoraproject.org/packages/cmake/2.8.12/3.fc21/src/cmake-2.8.12-3.  
↪fc21.src.rpm  
yum-builddep cmake-2.8.12-3.fc21.src.rpm  
rpmbuild --rebuild cmake-2.8.12-3.fc21.src.rpm  
yum install ~/rpmbuild/RPMS/x86_64/cmake-2.8.12-3.el7.centos.x86_64.rpm
```

Устанавливаем библиотеки разработчиков для сборки:

```
sudo yum install boost-devel openssl-devel
```

Если нужна поддержка UPnP (не забудьте потом запустить CMake с параметром WITH_UPNP=ON):

```
sudo yum install miniupnpc-devel
```

3.1.3 MAC OS X

Необходим [homebrew](#).

Устанавливаем библиотеки разработчиков для сборки:

```
brew install libressl boost
```

Собираем:

```
make HOMEBREW=1
```

3.1.4 FreeBSD

Для 10.X используйте clang. Вам так же понадобятся порты boost и openssl. Запустите gmake, он прочитает Makefile.bsd и сделает необходимые изменения.

Ветка 9.X использует gcc v4.2 который не поддерживает необходимый стандарт c++11.

Необходимые порты:

- devel/cmake
- devel/boost-libs
- lang/gcc47(или более новые версии)

Чтобы использовать более новый компилятор, установите следующие переменные (замените «47» своей версией gcc):

```
export CC=/usr/local/bin/gcc47
export CXX=/usr/local/bin/g++47
```

3.1.5 Опции CMake

Доступные опции CMake (подробности смотрите в `man 1 cmake`):

- CMAKE_BUILD_TYPE профиль сборки, отладочный или релиз (Debug/Release)
- WITH_BINARY сборка самого i2pd
- WITH_LIBRARY сборка библиотеки libi2pd
- WITH_STATIC сборка статических версий библиотеки и самого i2pd
- WITH_UPNP сборка с поддержкой UPnP (нужна библиотека libupnp)
- WITH_AESNI сборка с поддержкой AES-NI (ON/OFF)
- WITH_HARDENING включить Hardending (ON/OFF) (только с gcc)
- WITH_PCH использовать pre-compiled header (экспериментально, ускоряет процесс сборки)

Так же у CMake есть -L флаг, который показывает список текущих установленных опций:

```
cmake -L
```

3.2 Сборка для Android

Существуют 2 версии: с использованием QT и без него

3.2.1 Необходимые пакеты

Требуются Android SDK, NDK и QT с поддержкой андроида (для QT).

- [SDK](#) (выбрать command line tools only)
- [NDK](#)
- [QT](#)(только для QT). Выбрать версию для андроида для вашей платформы. Например QT 5.6 под линукс будет [этот файл](#)

Также потребуется Java JDK и Ant.

3.2.2 QT-Creator (только для QT)

Запустить QT-creator, который должен быть установлен вместе с QT. Идти в Settings/Android и указать пути к SDK and NDK. Если все сделано правильно то должно появиться: Android for armeabi-v7a (gcc, qt) и Android for x86 (gcc, qt).

3.2.3 Библиотеки

Следующие собранные библиотеки следует взять из репозитория PurpleI2P.

```
git clone https://github.com/PurpleI2P/Boost-for-Android-Prebuilt.git
git clone https://github.com/PurpleI2P/OpenSSL-for-Android-Prebuilt.git
git clone https://github.com/PurpleI2P/MiniUPnP-for-Android-Prebuilt.git
git clone https://github.com/PurpleI2P/android-ifaddrs.git
```

3.2.4 Сборка с QT

- Открыть qt/i2pd_qt/i2pd_qt.pro в QT-creator
- Поменять строчку MAIN_PATH = /path/to/libraries на директорию где лежат зависимости из предыдущего пункта
- Выбрать проект для целевой платформы (как правило armeabi-v7a) и собрать
- .apk файл появится в android-build/bin

3.2.5 Сборка без QT

- Поменять строку I2PD_LIBS_PATH в android/jni/Application.mk на директорию с зависимостями
- Запустить команду „ndk-build -j4“ из директории android
- Поправить файл „local.properties“. Строчки „sdk.dir=<путь к SDK>“ и „ndk.dir=<путь к NDK>“
- Run „ant clean debug“

3.2.6 Релизный .apk

Релизный .apk следует подписать своей подписью для чего требуется Java keystore file(.jks). Либо взять уже существующий, либо сделать новый с помощью keytool либо сконвертировать какой нибудь из своих сертификатов. Например i2pd подписан этим [сертификатом](#). Создать файл „ant.properties“ key.store=“путь к keystore file“ key.alias=“alias name“ key.store.password=“keystore password“ key.alias.password=“alias password“ Запустить „ant clean release“

3.3 Примеры использования

i2pd может быть использован для:

- *анонимных веб-сайтов*
- *анонимных чатов*
- *анонимного файлообмена*

и для много чего еще.

3.3.1 Посещение и создание веб-сайтов

Посещение анонимных веб-сайтов

Чтобы посещать анонимные веб-сайты внутри Невидимого Интернета, настройте свой веб-браузер на использование HTTP прокси 127.0.0.1:4444 (включено по-умолчанию в i2pd).

В Firefox: Настройки -> Дополнительные -> Сеть -> Соединение -> Настроить -> выберите Ручная настройка сервиса прокси, введите HTTP прокси 127.0.0.1, Порт 4444.

В Chromium: запустите chromium с ключом

```
chromium --proxy-server="http://127.0.0.1:4444"
```

Если вы хотите остаться анонимным при посещении веб-сайтов, вам необходимо так же настроить свой веб-браузер для защиты от деанонимизации. Как именно это делается можно найти в интернете, например [здесь](#).

Большой список сайтов Невидимого Интернета на сайте [identiguu.i2p](#).

Создание анонимных веб-сайтов

Если вы хотите запустить свой сайт в Невидимом Интернете, следуйте данной инструкции:

1. Запустите веб-сервер и узнайте, какой хост:порт он использует (например, 127.0.0.1:8080).
2. Настройте i2pd для создания серверного HTTP туннеля. В файл ~/.i2pd/tunnels.conf добавьте:

```
[anon-website]
type = http
host = 127.0.0.1
port = 8080
keys = anon-website.dat
```

3. Перезапустите i2pd.

4. Найдите b32 destination вашего вебсайта.

Зайдите в веб-консоль -> [страница I2P tunnels](#). Ищите Sever tunnels и увидите адрес <длинная строка>.b32.i2p рядом с anon-website.

Теперь веб-сайт доступен в Невидимом Интернете по этому адресу.

5. (Опционально) Зарегистрируйте короткий и запоминающийся .i2p домен на [inr.i2p](#).

3.3.2 Использование и создание чат-серверов

Запуск анонимного IRC сервера

1. Запустите свой IRC сервер и узнайте, какой хост:порт он использует (например, 127.0.0.1:5555).

Для малых частных IRC серверов вы можете использовать [miniircd](#), для больших публичных [UnrealIRCd](#).

2. Настройте i2pd для создания серверного IRC туннеля.

В самом простом случае, если ваш сервер не поддерживает WebIRC, добавьте следующее в `~/i2pd/tunnels.conf`:

```
[anon-chatserver]
type = irc
host = 127.0.0.1
port = 5555
keys = chatserver-key.dat
```

И этого достаточно.

В случае, если ваш IRC сервер поддерживает WebIRC, например, это UnrealIRCd, добавьте следующее в конфиг UnrealIRCd:

```
webirc {
    mask 127.0.0.1;
    password your_password;
};
```

Также измените строку:

```
modes-on-connect "+ixw";
```

на

```
modes-on-connect "+iw";
```

И добавьте это в `~/i2pd/tunnels.conf`:

```
[anon-chatserver]
type = irc
host = 127.0.0.1
port = 5555
keys = chatserver-key.dat
webircpassword = your_password
```

3. Перезапустите i2pd.

4. Найдите b32 destination вашего анонимного IRC сервера.

Зайдите в веб-консоль -> [страница I2P tunnels](#). Ищите Sever tunnels и увидите адрес <длинная строка>.b32.i2p рядом с anon-chatserver.

Пользователи будут использовать этот адрес для анонимного подключения к вашему серверу.

Подключение к анонимному IRC серверу

Чтобы подключиться к IRC серверу по адресу *walker.i2p*, добавьте следующее в `~/i2pd/tunnels.conf`:

```
[IRC2]
type = client
address = 127.0.0.1
port = 6669
destination = walker.i2p
#keys = walker-keys.dat
```

Перезапустите i2pd, после чего подключайтесь к `irc://127.0.0.1:6669` в вашем IRC клиенте.

3.3.3 Файлообмен

Вы можете качать и раздавать торренты клиентом [Transmission-I2P](#).

Другие торрент-клиенты [Robert](#) и [Vuze](#).

Robert использует протокол BOB, i2pd должен быть запущен с настройкой `-bob.enabled=true`.

Vuze использует протокол I2CP, i2pd должен быть запущен с настройкой `-i2cp.enabled=true`.

Используйте [трекер postman](#).