Wallet Building Guide V1.1 (October 19, 2018)

What Does this Document Cover?

This document covers building of the Windows, Mac OS X, and Ubuntu wallets. It gives detailed setup instructions to guide you and assist you in any issues that you may encounter. This document will be revised periodically to contain updated information over the build environment, and to have the most updated documentation over the Spectre Security Coin software itself.

1. Installation
2. Configuration
3. Joining the main network
4. Joining the test network
5. Building the software
6. Masternode Setup
7. Additional resources

Installing the Spectre Security Coin Build Environment

Building Applications for the blockchain will always be a great importance to the development of the community. We wanted to make sure that we had a strong process. We want people to be able to follow our progress and play with the latest builds, or build applications that uses the blockchain in some way. This will help include the community in the development process and increase the feeling that can connect to the developers. This promotes a more harmonious culture and promotes communities working together, by creating new applications that uses Spectre Security Coin as a form of payment.

To get started, you can use the Spectre Security Coin installation script to do most tasks or manually build the software yourself. To create the correct environment to build Spectre Security you will need to have the following items.

Requirements:

1. Ubuntu 16.0.4.5 AMD 64 Desktop
2. http://releases.ubuntu.com/16.04/ubuntu-16.04.5-desktop-amd64.iso
3. 8 Gigs of Ram
4. 20 Gigs of disk space

When installing Ubuntu Linux 16.04 AMD64 Desktop you need to create the first user as “spectre”. This should be all lowercase. This is the path in the source files that we will be using to build the software. Do not update to Ubuntu Linux 17 or 18 as this will break software dependencies.
After the system is installed, you need to run the following commands from the desktop of the “spectre” user account.

1. sudo apt-get update -qq
2. sudo apt-get upgrade -qq
3. sudo apt-get dist-upgrade -qq

This will ensure that the operating system is up to date and ready to begin. This script is located on our github account for anyone to use freely. Once this has been done, copy and paste the next line into your terminal.

wget -O -
https://raw.githubusercontent.com/SpectreSecurity/SpectreSecurityCoin/master/doc/spectre_install.sh | bash

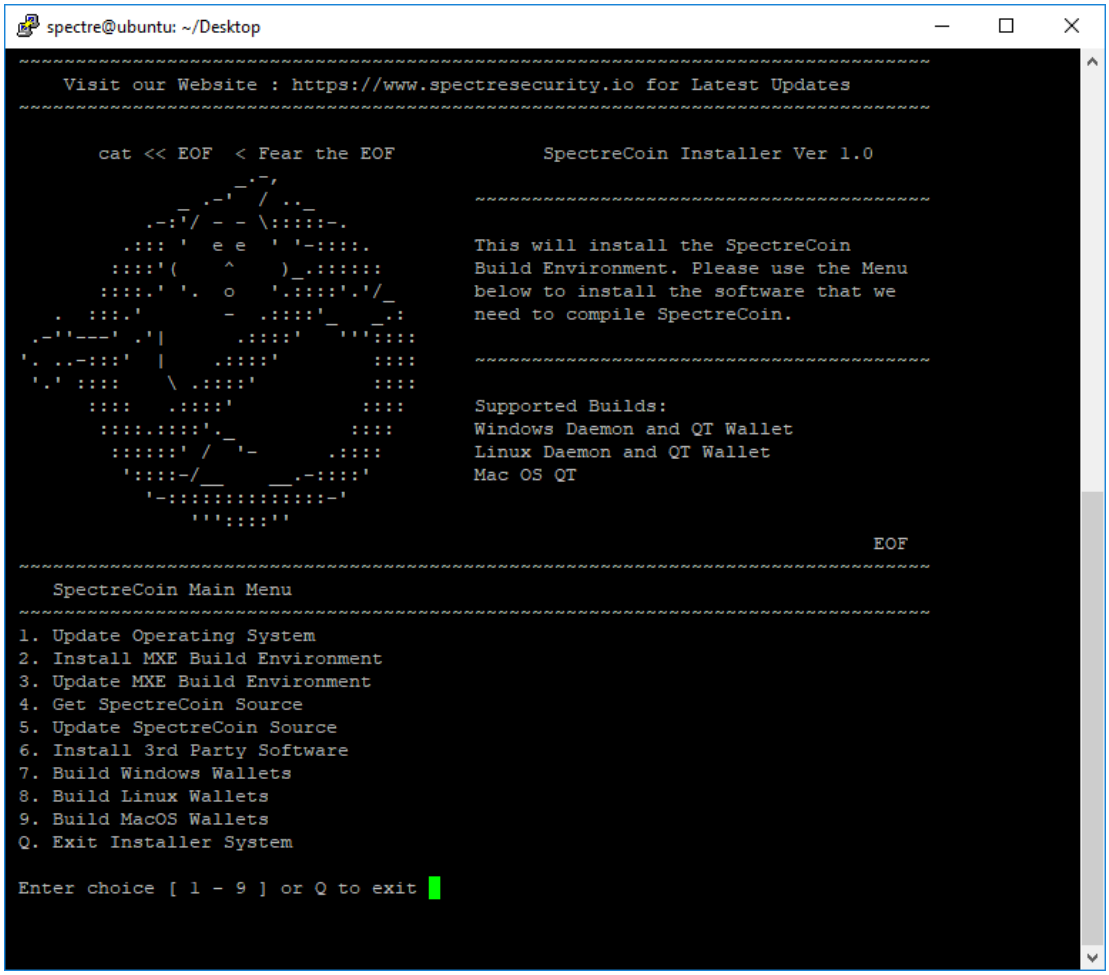
This should download and run the spectre_install.sh script and begin the process of creating the build environment. Some parts of the process will take a long time and other parts will move quicker. You will need to perform the correct steps in the correct order for the platform that you want to build for.

Spectre Security Coin Installer

The first time you enter this menu you will need to perform the installation and then choose which platform to build for. If you had built it before and need to update components, then please, use the updated items or manually update the components, yourself. You need to execute the following options in the correct order.

1. Select 1 to update the operating system.
2. Select 2 to install MXE build Environment
3. Select 4 to get Spectre Security Source Code
4. Select 6 to install 3rd party packages

While performing these steps please make sure to pay attention for any errors. Any error may not let you go to the next step. It is frequent that you may see warnings. These warnings are normal and should not be of concern. It should not affect the software building.



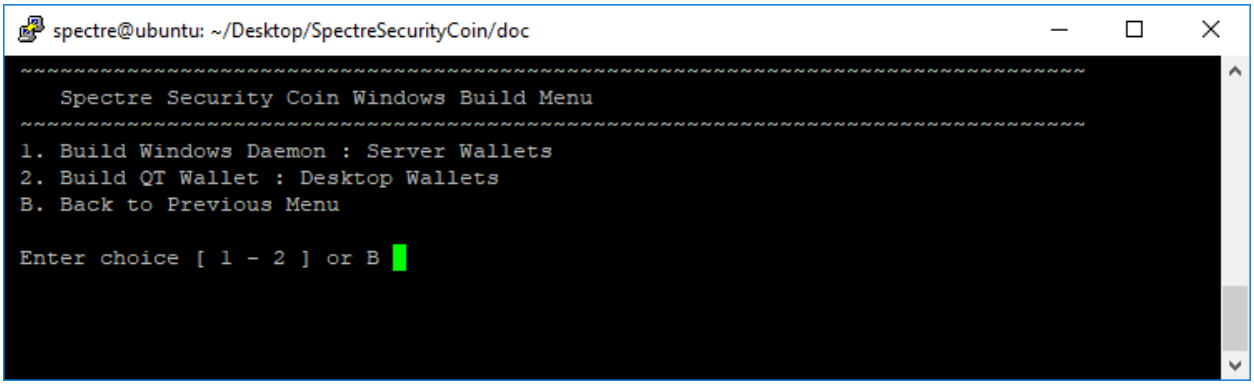
From here you have few choices, when you are done use the Q Letter to exit the installer. As long as the above steps completed successfully, you can move on the next step in this guide. We will outline the windows wallet first.

Windows QT Wallet

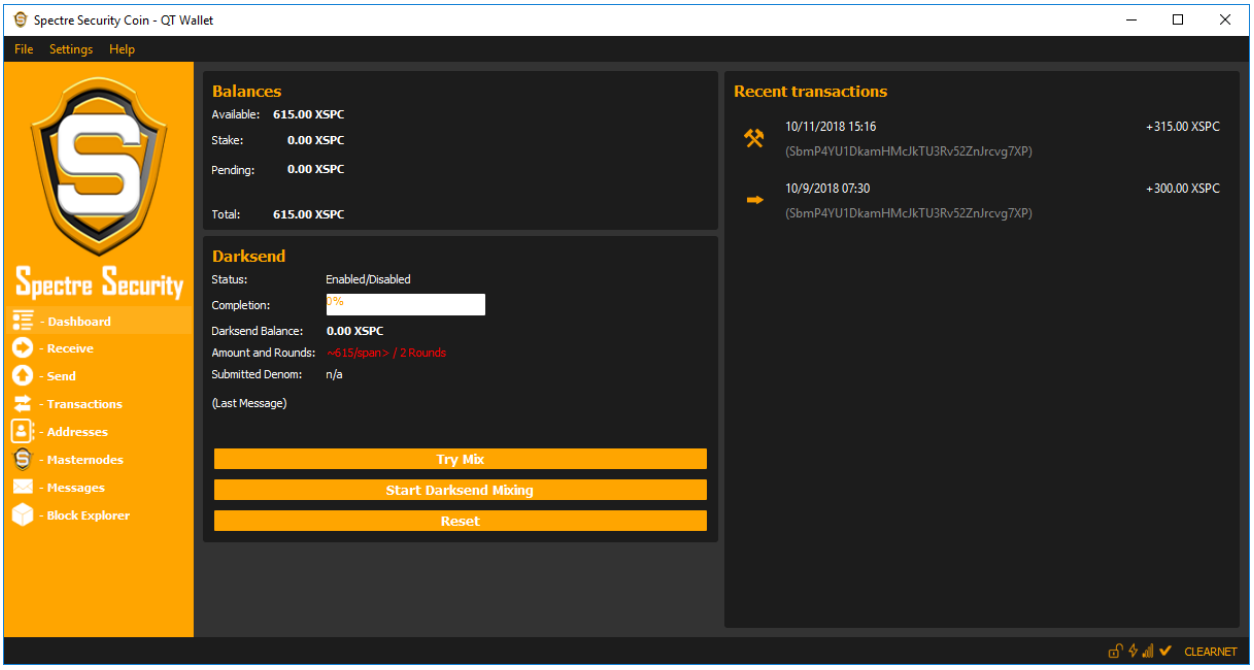
One of the biggest ways for people to get involved is to have them join the network. For people who want to buy and sell, or even PoS, they can use this wallet. By default, this wallet is not an internet peer. You will not be required to open any internet ports for the wallet to connect. For any reason if you cannot connect and get peers, you can simply add peers in the debug menu. Peers can also be added in the spectresecuritycoin.conf file, as well as many other options. Please see the ending of this paper for more advanced uses.

You can build the wallet from the following menu by selecting these following options. Once the binary is built it will be in place is the ~/Desktop/SpecreSecurityCoin/release folder.

1. Select 7 to get to the Windows build wallet menu
2. Select 2 to build the Windows Desktop (QT) Wallet



A file called SpectreSecurityCoin-qt.exe will be now in the release folder. After the file is built you can use B to return to the main menu.



Manually Building the Windows Wallet

If you want to manually build the wallet, you can follow the commands below. This needs to be performed in the correct steps.

1. sudo apt-get update -qq
2. sudo apt-get upgrade -qq
3. sudo apt-get dist-upgrade -qq
4. sudo apt-get install g++-multilib libc6-dev-i386 p7zip-full autoconf automake autopoint bash bison bzip2 cmake flex gettext git g++ gperf intltool libffi-dev libtool libltdl-dev libssl-dev libxml-parser-perl make openssl patch perl pkg-config python ruby scons sed unzip wget xz-utils libtool-bin ruby scons libtool libgdk-pixbuf2.0-dev dos2unix ntp -qq
5. cd /mnt
6. sudo git clone https://github.com/mxe/mxe.git
7. sudo make MXE_TARGETS="i686-w64-mingw32.static" qt
8. sudo make MXE_TARGETS="i686-w64-mingw32.static" qttools
9. sudo make MXE_TARGETS="i686-w64-mingw32.static" boost
10. cd /mnt/mxe/src
11. sudo rm openssl.mk
12. sudo wget https://media.spectresecurity.io/scripts/mxe/openssl.mk
13. cd ..
14. sudo make qt5 miniupnpc db boost -j4 MXE_PLUGIN_DIRS=$PWD/plugins/examples/qt5-freeze
15. sudo make build-only-openssl_i686-w64-mingw32.static
16. cd /home/spectre/Desktop
17. git clone https://github.com/SpectreSecurityCoin/SpectreCoin.git
18. cd /home/spectre/Desktop/
19. wget https://github.com/SpectreSecurityCoin/Scripts/raw/master/openssl-1.0.2b.tar.gz
20. tar -xzvf openssl-1.0.2b.tar.gz
21. cp -R openssl-1.0.2b openssl-win32-build
22. cd openssl-win32-build
23. make clean
24. chmod -Rv 755 *
25. export PATH=/mnt/mxe/usr/bin:$PATH
26. export PATH=$MXEPATH/bin:$PATH
27. CROSS_COMPILE="i686-w64-mingw32.static-" ./Configure mingw no-asm no-shared --prefix=/mnt/mxe/usr/i686-w64-mingw32.static
28. make -j4
29. cd /home/spectre/Desktop/
30. wget https://github.com/SpectreSecurityCoin/Scripts/raw/master/db-4.8.30.tar.gz
31. tar -xzvf db-4.8.30.tar.gz
32. cd /home/spectre/Desktop/db-4.8.30
33. make clean
34. wget https://raw.githubusercontent.com/SpectreSecurityCoin/Scripts/master/berekeydb.sh
35. dos2unix berekeydb.sh
36. chown -Rv 755 *

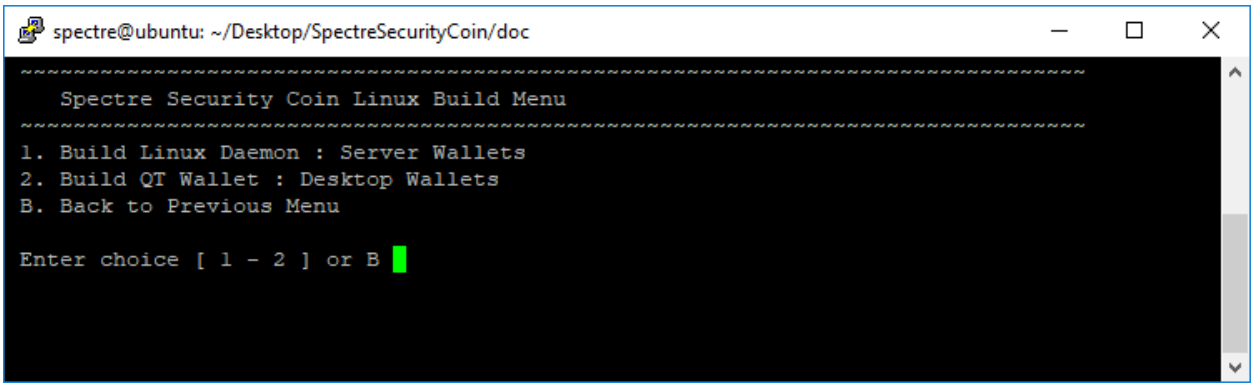
37. export PATH=/mnt/mxe/usr/bin:$PATH
38. export PATH=$MXEPATH/bin:$PATH
39. sudo ./berekeydb.sh
40. cd /home/spectre/Desktop/SpectreSecurityCoin
41. make clean
42. cd src/leveldb
43. make clean
44. export PATH=/mnt/mxe/usr/bin:$PATH
45. export PATH=$MXEPATH/bin:$PATH
46. TARGET_OS=NATIVE_WINDOWS make -j4 CC=i686-w64-mingw32.static-gcc CXX=i686-w64-mingw32.static-g++ libleveldb.a libmemenv.a
47. cd ../..
48. chmod 775 * -R
49. cd src/secp256k1
50. make clean
51. export PATH=/mnt/mxe/usr/bin:$PATH
52. export PATH=$MXEPATH/bin:$PATH
53. chmod 775 * -R
54. ./autogen.sh
55. ./configure --host=i686-w64-mingw32.static --enable-static --disable-shared --enable-module-recovery
56. make -j4
57. cd ../..
58. export PATH=/mnt/mxe/usr/bin:$PATH
59. export PATH=$MXEPATH/bin:$PATH
60. /mnt/mxe/usr/i686-w64-mingw32.static/qt5/bin/qmake SpectreSecurityCoin.pro
61. make -j4

You should now have the SpectreSecurityCoin-qt.exe in the release folder.

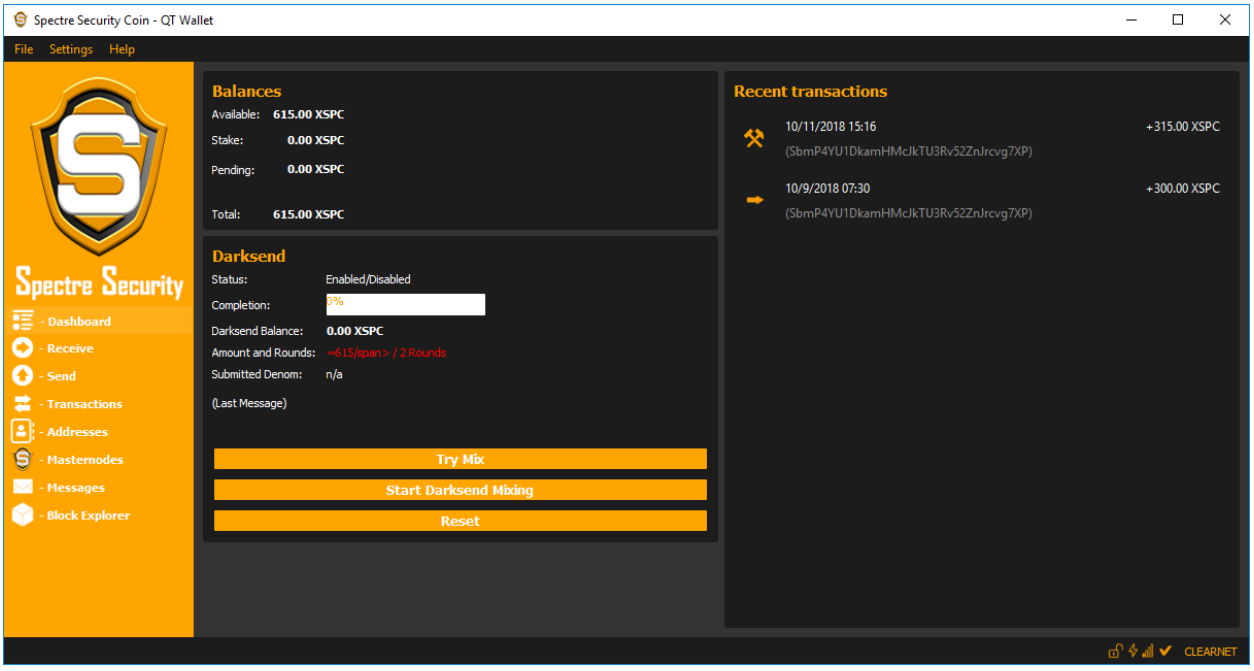
Building the Ubuntu Desktop QT Wallet

You can build the wallet from the following menu by selecting these following options. Once the binary is built, it will be placed in the ~/Desktop/SpecreSecurityCoin/release folder.

1. Select 8 to get the Windows wallet menu
2. Select 2 to build the Windows Desktop (QT) Wallet



You should now have a SpecreSecurityCoin-qt file generated. This is the Ubuntu Linux Desktop (QT) Wallet. After the file is built, you can use B to return to the main menu.



If you want to manually build the linux qt wallet, you can do by following these steps.

1. sudo apt-get update -qq
2. sudo apt-get upgrade -qq
3. sudo apt-get dist-upgrade -qq
4. sudo apt-get install g++-multilib libc6-dev-i386 p7zip-full autoconf automake autopoint bash bison bzip2 cmake flex gettext git g++ gperf intltool libffi-dev libtool libltdl-dev libssl-dev libxml-parser-perl make openssl patch perl pkg-config python ruby scons sed unzip wget xz-utils libtool-bin ruby scons libtool libgdk-pixbuf2.0-dev dos2unix ntp -qq
5. cd /mnt
6. sudo git clone https://github.com/mxe/mxe.git && cd /mnt/mxe/
7. sudo make MXE_TARGETS="i686-w64-mingw32.static" qt
8. sudo make MXE_TARGETS="i686-w64-mingw32.static" qttools
9. sudo make MXE_TARGETS="i686-w64-mingw32.static" boost
10. cd /mnt/mxe/src
11. sudo rm openssl.mk
12. sudo wget https://raw.githubusercontent.com/SpectreSecurityCoin/Scripts/master/openssl.mk
13. cd ..
14. sudo make qt5 miniupnpc db boost -j4 MXE_PLUGIN_DIRS=$PWD/plugins/examples/qt5-freeze
15. sudo make build-only-openssl_i686-w64-mingw32.static
16. cd /home/spectre/Desktop
17. git clone https://github.com/SpectreSecurityCoin/SpectreCoin.git
18. cd /home/spectre/Desktop/
19. wget https://github.com/SpectreSecurityCoin/Scripts/raw/master/openssl-1.0.2b.tar.gz
20. tar -xzvf openssl-1.0.2b.tar.gz
21. cp -R openssl-1.0.2b openssl-win32-build
22. make clean
23. cd openssl-win32-build
24. export PATH=/mnt/mxe/usr/bin:$PATH
25. export PATH=$MXEPATH/bin:$PATH
26. CROSS_COMPILE="i686-w64-mingw32.static-" ./Configure mingw no-asm no-shared --prefix=/mnt/mxe/usr/i686-w64-mingw32.static
27. sudo make depend
28. make -j4
29. cd /home/spectre/Desktop/
30. wget https://github.com/SpectreSecurityCoin/Scripts/raw/master/db-4.8.30.tar.gz
31. tar zxvf db-4.8.30.tar.gz
32. cd /home/spectre/Desktop/db-4.8.30
33. make clean
34. wget https://raw.githubusercontent.com/SpectreSecurityCoin/Scripts/master/berekeydb.sh
35. export PATH=/mnt/mxe/usr/bin:$PATH
36. export PATH=$MXEPATH/bin:$PATH
37. dos2unix berekeydb.sh
38. chmod ugo+x berekeydb.sh
39. sudo ./berekeydb.sh

40. cd /home/spectre/Desktop/SpectreSecurityCoin
41. make clean
42. cd src/leveldb
43. dos2unix
44. export PATH=/mnt/mxe/usr/bin:$PATH
45. export PATH=$MXEPATH/bin:$PATH
46. TARGET_OS=NATIVE_WINDOWS make -j4 CC=i686-w64-mingw32.static-gcc CXX=i686-w64-mingw32.static-g++ libleveldb.a libmemenv.a
47. cd ../..
48. chmod 775 * -R
49. cd src/secp256k1
50. make clean
51. export PATH=/mnt/mxe/usr/bin:$PATH
52. export PATH=$MXEPATH/bin:$PATH
53. chmod 775 * -R
54. ./autogen.sh
55. ./configure --host=i686-w64-mingw32.static --enable-static --disable-shared --enable-module-recovery
56. make -j4
57. cd ../..
58. sudo apt-get install build-essential libtool autotools-dev automake pkg-config libssl-dev libevent-dev bsdmainutils libgmp-dev libboost-system-dev libboost-filesystem-dev libboost-chrono-dev libboost-program-options-dev libboost-test-dev libboost-thread-dev libboost-all-dev -qq
59. sudo apt-get install software-properties-common
60. sudo add-apt-repository ppa:bitcoin/bitcoin
61. sudo apt-get update -qq
62. sudo apt-get install libdb4.8-dev libdb4.8++-dev libminiupnpc-dev libzmq3-dev -qq
63. sudo apt-get install libqt5gui5 libqt5core5a libqt5dbus5 qttools5-dev qttools5-dev-tools -qq
64. make clean
65.
66. export PATH=/mnt/mxe/usr/bin:$PATH
67. export PATH=$MXEPATH/bin:$PATH
68. qmake linux.pro
69. make -j4

Building the Linux Daemon

To build the Linux daemon for application use, please follow the steps below. You will need a VPS with the following requirements.

1. Ubuntu 16.04.5 Server (ubuntu-16.04.5-server-amd64.iso)
2. 1 Gig ram
3. 1 core processor

When you have installed the main Ubuntu server OS, please use these instructions to build the daemon software.

1. apt-get install build-essential libtool autotools-dev automake pkg-config libssl-dev libevent-dev bsdmainutils libgmp-dev -qq
2. apt-get install libboost-system-dev libboost-filesystem-dev libboost-chrono-dev libboost-program-options-dev libboost-test-dev libboost-thread-dev libboost-all-dev -qq
3. apt-get install software-properties-common
4. add-apt-repository ppa:bitcoin/bitcoin
5. apt-get update -qq
6. apt-get install libdb4.8-dev libdb4.8++-dev libminiupnpc-dev libzmq3-dev -qq
7. git clone https://github.com/SpectreSecurityCoin/SpectreSecurityCoin.git
8. cd SpectreSecurityCoin/src
9. make -j4 -f makefile.unix

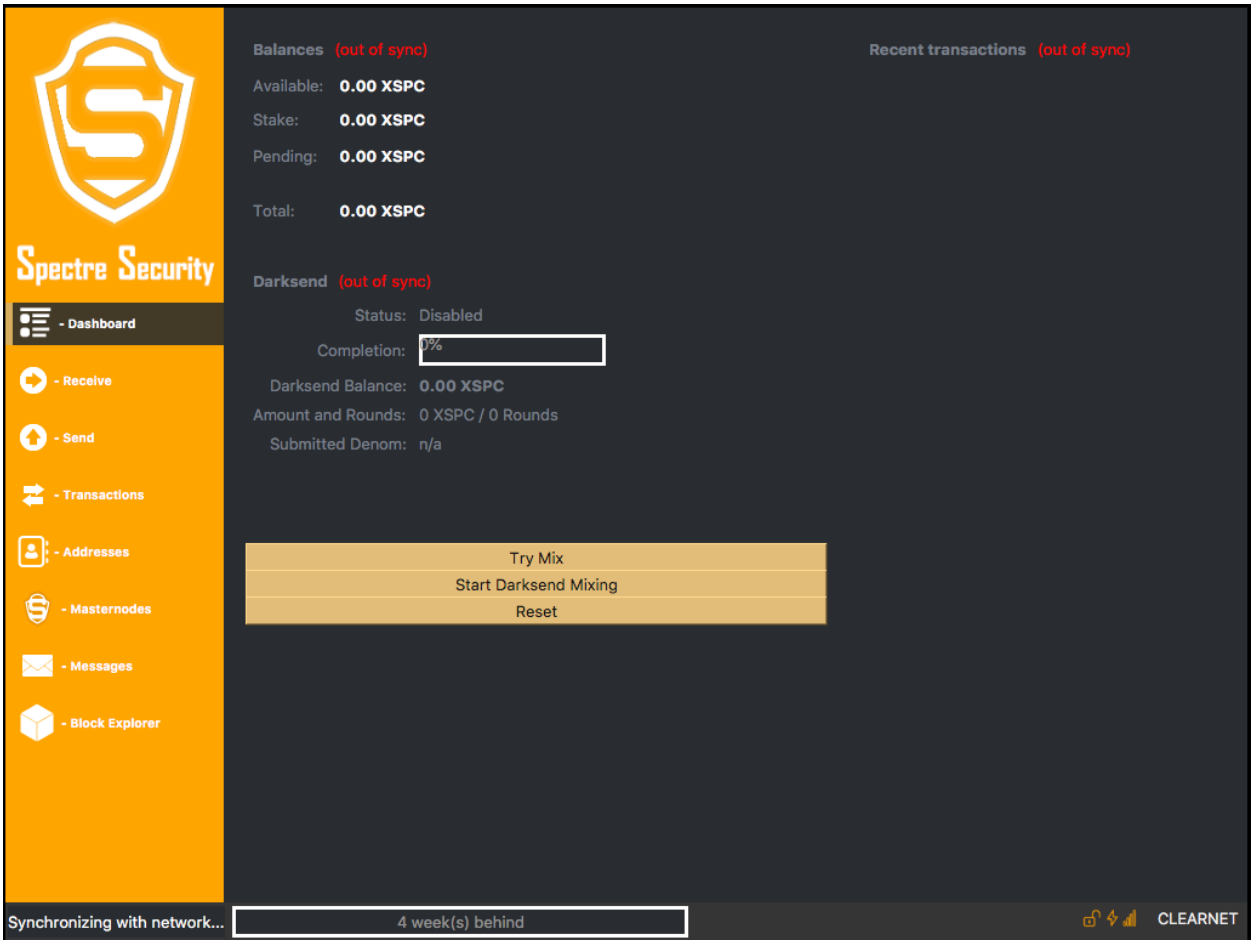
MAC OS X: High Serra 10.13.3 DMG QT Build

Please make sure that your mac is fully patched and can be connected to iTunes. We will need to install software such as XCode and qt creator**.** Sometimes, if the application will not build, it will be necessary to run make clean first.

Requirements:

1. MAC OS X High Serra
    a. Version 10.13.3
2. Install XCODE 8.1 (or higher 9.2 tested)
    a. https://itunes.apple.com/us/app/xcode/id497799835?mt=12
3. Install Qt Creator 5.10.1
4. Install GIT
    a. http://mac.github.com/
5. Install Brew
    a. /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"

Mac OS X QT Wallet Build Instructions:



The following will install the dependencies that are needed to build the wallet. Please run them in order and make sure that each step is completed before moving on to the next.

1. sudo xcode-select –install
2. brew install automake berkeley-db4 libtool boost miniupnpc openssl pkg-config protobuf python qt libevent qrencode librsvg cmake autoconf libtool gcc automake
3. git clone https://github.com/SpectreSecurityCoin/SpectreSecurityCoin.git
4. cd SpectreSecurityCoin

Your environment should now be set up and ready to compile the SpectreSecurityCoin QT Mac Wallet.

5. make clean
6. export CPPFLAGS="-I/usr/local/opt/openssl/include"
7. export LDFLAGS="-L/usr/local/opt/openssl/lib"
8. qmake -spec macx-g++ macos.pro
9. make -j4

Debug Log File:

The debug file is created when the application is running. Please use the debug to log any errors and please pay attention to it as it contains a lot of useful information. It will be located in the following folder C:\Users\SpectreSecurity\AppData\Roaming\ SpectreSecurityCoin\ in this build example.

As the application runs this file will grow in size. Periodically removing the file will make troubleshooting easier. If deleted this file will regenerate on the next application start.

Spectre Security Coin Wallet Start Up Parameters:

You can start the Spectre Security Coin software with extra options. These are considered advanced features. This example will be over the windows wallet but should still work for mac so and Linux to a good point. But your experience may vary.

In this example we will call the wallet exe file, say where the blockchain is stored, and what spectresecuritycoin.conf file to read.

M:\Coins\c11\SpectreSecurityCoin\SpectreSecurityCoin-qt.exe
-conf=M:\Coins\c11\SpectreSecurityCoin\blockchain\spectresecuritycoin.conf
-datadir=M:\Coins\c11\SpectreSecurityCoin\blockchain

If we use the options -datadir and -conf we can over-ride the normal storage location, which is %appdata% by default. It also says which spectresecuritycoin.conf file to load.

If we wanted to run a wallet as a server, daemon, and to listen for peers we would use this example:

M:\Coins\c11\SpectreSecurityCoin\SpectreSecurityCoin-qt.exe
-conf=M:\Coins\c11\SpectreSecurityCoin\blockchain\spectresecuritycoin.conf
-datadir=M:\Coins\c11\SpectreSecurityCoin\blockchain -server -daemon -listen

Sometimes we will set application flags when we first call the wallet binary, other times we will set these flags in the spectresecuritycoin.conf.

In addition, it also means that we can run multiple copies of **Spectre Security Coin** on a machine. We need to specify another directory for blockchain and also have a copy of spectresecuritycoin.conf in that folder. When running multiple wallets on the same server, you will also have to increment the ports. Each wallet running has two (2) ports in each config.

1. Wallet 1
    a. rpcport=13337
    b. port=13338
2. Wallet 2
    a. rpcport=13339
    b. port=13340

Options:
-? This help message
-conf=<file> Specify configuration file (default: SpectreSecurityCoin.conf)
-pid=<file> Specify pid file (default: SpectreSecurityCoind.pid)
-datadir=<dir> Specify data directory
-wallet=<dir> Specify wallet file (within data directory)
-dbcache=<n> Set database cache size in megabytes (default: 100)
-dblogsize=<n> Set database disk log size in megabytes (default: 100)
-timeout=<n> Specify connection timeout in milliseconds (default: 5000)
-proxy=<ip:port> Connect through SOCKS5 proxy
-tor=<ip:port> Use proxy to reach tor hidden services (default: same as -proxy)
-dns Allow DNS lookups for -addnode, -seednode and -connect
-port=<port> Listen for connections on <port> (default: 13338)
-maxconnections=<n> Maintain at most <n> connections to peers (default: 125)
-addnode=<ip> Add a node to connect to and attempt to keep the connection open
-connect=<ip> Connect only to the specified node(s)
-seednode=<ip> Connect to a node to retrieve peer addresses, and disconnect
-externalip=<ip> Specify your own public address
-onlynet=<net> Only connect to nodes in network <net> (IPv4, IPv6 or Tor)
-discover Discover own IP address (default: 1 when listening and no -externalip)
-listen Accept connections from outside (default: 1 if no -proxy or -connect)
-bind=<addr> Bind to given address. Use [host]:port notation for IPv6
-dnsseed Query for peer addresses via DNS lookup, if low on addresses (default: 1 unless -connect)
-forcednsseed Always query for peer addresses via DNS lookup (default: 0)
-synctime Sync time with other nodes. Disable if time on your system is precise e.g. syncing with NTP (default: 1)
-banscore=<n> Threshold for disconnecting misbehaving peers (default: 100)
-bantime=<n> Number of seconds to keep misbehaving peers from reconnecting (default: 86400)
-maxreceivebuffer=<n> Maximum per-connection receive buffer, <n>*1000 bytes (default: 5000)
-maxsendbuffer=<n> Maximum per-connection send buffer, <n>*1000 bytes (default: 1000)
-upnp Use UPnP to map the listening port (default: 1 when listening)
-paytxfee=<amt> Fee per KB to add to transactions you send
-mininput=<amt> When creating transactions, ignore inputs with value less than this (default: 0.01)
-server Accept command line and JSON-RPC commands
-testnet Use the test network
-debug=<category> Output debugging information (default: 0, supplying <category> is optional)
If <category> is not supplied, output all debugging information.
<category> can be: addrman, alert, db, lock, rand, rpc, selectcoins, mempool, net, coinage, coinstake, creation, stakemodifier, qt.
-logtimestamps Prepend debug output with timestamp
-shrinkdebugfile Shrink debug.log file on client startup (default: 1 when no -debug)
-printtoconsole Send trace/debug info to console instead of debug.log file
-regtest Enter regression test mode, which uses a special chain in which blocks can be solved instantly. This is intended for regression testing tools and app development.
-rpcuser=<user> Username for JSON-RPC connections
-rpcpassword=<pw> Password for JSON-RPC connections
-rpcport=<port> Listen for JSON-RPC connections on <port> (default: 13338)
-rpcallowip=<ip> Allow JSON-RPC connections from specified IP address
-rpcthreads=<n> Set the number of threads to service RPC calls (default: 4)
-blocknotify=<cmd> Execute command when the best block changes (%s in cmd is replaced by block hash)
-walletnotify=<cmd> Execute command when a wallet transaction changes (%s in cmd is replaced by TxID)
-confchange Require a confirmations for change (default: 0)
-alertnotify=<cmd> Execute command when a relevant alert is received (%s in cmd is replaced by message)
-upgradewallet Upgrade wallet to latest format
-createwalletbackups=<n> Number of automatic wallet backups (default: 10)
-keypool=<n> Set key pool size to <n> (default: 100) (litemode: 10)
-rescan Rescan the block chain for missing wallet transactions
-salvagewallet Attempt to recover private keys from a corrupt wallet.dat
-checkblocks=<n> How many blocks to check at startup (default: 500, 0 = all)

-checklevel=<n> How thorough the block verification is (0-6, default: 1)
-loadblock=<file> Imports blocks from external blk000?.dat file
-maxorphanblocks=<n> Keep at most <n> unconnectable blocks in memory (default: 10000)

Block creation options:
-blockminsize=<n> Set minimum block size in bytes (default: 0)
-blockmaxsize=<n> Set maximum block size in bytes (default: 250000)
-blockprioritysize=<n> Set maximum size of high-priority/low-fee transactions in bytes (default: 27000)

SSL options: (see the Bitcoin Wiki for SSL setup instructions)
-rpcssl Use OpenSSL (https) for JSON-RPC connections
-rpcsslcertificatechainfile=<file.cert> Server certificate file (default: server.cert)
-rpcsslprivatekeyfile=<file.pem> Server private key (default: server.pem)
-rpcsslciphers=<ciphers> Acceptable ciphers (default:
TLSv1.2+HIGH:TLSv1+HIGH:!SSLv3:!SSLv2:!aNULL:!eNULL:!3DES:@STRENGTH)
-litemode=<n> Disable all Darksend and Stealth Messaging related functionality (0-1, default: 0)

Masternode options:
-masternode=<n> Enable the client to act as a masternode (0-1, default: 0)
-mnconf=<file> Specify masternode configuration file (default: masternode.conf)
-mnconflock=<n> Lock masternodes from masternode configuration file (default: 1)
-masternodesoftlock=<n> Prevent masternode collateral from being used (0-1, default: 0)
-masternodeprivkey=<n> Set the masternode private key
-masternodeaddr=<n> Set external address:port to get to this masternode (example: address:port)
-masternodeminprotocol=<n> Ignore masternodes less than version (example: 61401; default : 0)

Darksend options:
-enabledarksend=<n> Enable use of automated darksend for funds stored in this wallet (0-1, default: 0)
-darksendrounds=<n> Use N separate masternodes to anonymize funds (2-8, default: 2)
-anonymizeSpectreSecurityCoinamount=<n> Keep N SpectreSecurityCoin anonymized (default: 0)
-liquidityprovider=<n> Provide liquidity to Darksend by infrequently mixing coins on a continual basis (0-100, default: 0, 1=very frequent, high fees, 100=very infrequent, low fees)

InstantX options:
-enableinstantx=<n> Enable instantx, show confirmations for locked transactions (bool, default: true)
-instantxdepth=<n> Show N confirmations for a successfully locked transaction (0-9999, default: 10)
Secure messaging options:
-nosmsg Disable secure messaging.
-debugsmsg Log extra debug messages.
-smsgscanchain Scan the block chain for public key addresses on startup.
-stakethreshold=<n> This will set the output size of your stakes to never be below this number (default: 100)

UI options:
-lang=<lang> Set language, for example "de_DE" (default: system locale)
-min Start minimized
-splash Show splash screen on startup (default: 1)

Wallet RPC Calls

addmultisigaddress nrequired ["key",...] ("account")
addnode <node> <add|remove|onetry>
addredeemscript <redeemScript> [account]
backupwallet "destination"
checkkernel [{"txid":txid,"vout":n},...] [createblocktemplate=false]
checkwallet
clearbanned
createmultisig nrequired ["key",...]
createrawtransaction [{"txid":txid,"vout":n},...] {address:amount,...}
darksend <SpectreSecurityCoinaddress> <amount>
decoderawtransaction <hex string>
decodescript <hex string>
dumpprivkey <SpectreSecurityCoinaddress>
dumpwallet <filename>
encryptwallet "passphrase"
getaccount "SpectreSecurityCoinaddress"
getaccountaddress "account"
getaddednodeinfo <dns> [node]
getaddressesbyaccount "account"
getbalance ("account" minconf includeWatchonly)
getbestblockhash
getblock <hash> [txinfo]
getblockbynumber <number> [txinfo]
getblockcount
getblockhash <index>
getblocktemplate [params]
getcheckpoint
getconnectioncount
getdifficulty
getinfo
getmininginfo
getnettotals
getnetworkhashps
getnewaddress ("account")
getnewpubkey [account]
getnewstealthaddress [label]
getpeerinfo
getrawmempool
getrawtransaction <txid> [verbose=0]
getreceivedbyaccount "account" (minconf)
getreceivedbyaddress "SpectreSecurityCoinaddress" (minconf)
getstakesubsidy <hex string>
getstakinginfo
getsubsidy [nTarget]
gettransaction "txid" (includeWatchonly)
gettxout "txid" n (includemempool)
getwork [data]
getworkex [data, coinbase]
help [command]
importaddress "address" ("label" rescan)
importprivkey <SpectreSecurityCoinprivkey> [label] [rescan=true]
importstealthaddress <scan_secret> <spend_secret> [label]

importwallet <filename>
keypoolrefill (newsize)
listaccounts (minconf includeWatchonly)
listaddressgroupings
listbanned
listreceivedbyaccount (minconf includeempty includeWatchonly)
listreceivedbyaddress (minconf includeempty includeWatchonly)
listsinceblock ("blockhash" target-confirmations includeWatchonly)
liststealthaddresses [show_secrets=0]
listtransactions ("account" count from includeWatchonly)
listunspent [minconf=1] [maxconf=9999999] ["address",...]
makekeypair [prefix]
masternode "command"... ("passphrase")
masternodelist ("mode" "filter")
moneysupply
move "fromaccount" "toaccount" amount (minconf "comment")
ping
repairwallet
resendtx
reservebalance [<reserve> [amount]]
scanforalltxns [fromHeight]
scanforstealthtxns [fromHeight]
searchrawtransactions <address> [verbose=1] [skip=0] [count=100]
sendalert <message> <privatekey> <minver> <maxver> <priority> <id> [cancelupto]
sendfrom "fromaccount" "toSpectreSecurityCoinaddress" amount (minconf "comment" "comment-to")
sendmany "fromaccount" {"address":amount,...} (minconf "comment")
sendrawtransaction <hex string>
sendtoaddress "SpectreSecurityCoinaddress" amount ("comment" "comment-to")
sendtostealthaddress <stealth_address> <amount> [comment] [comment-to] [narration]
setaccount "SpectreSecurityCoinaddress" "account"
setban "ip(/netmask)" "add|remove" (bantime) (absolute)
settxfee amount
signmessage "SpectreSecurityCoinaddress" "message"
signrawtransaction <hex string> [{"txid":txid,"vout":n,"scriptPubKey":hex,"redeemScript":hex},...] [<privatekey1>,...] [sighashty
pe="ALL"]
smsgaddkey <address> <pubkey>
smsgbuckets [stats|dump]
smsgdisable
smsgenable
smsggetpubkey <address>
smsginbox [all|unread|clear]
smsglocalkeys [whitelist|all|wallet|recv <+/-> <address>|anon <+/-> <address>]
smsgoptions [list|set <optname> <value>]
smsgoutbox [all|clear]
smsgscanbuckets
smsgscanchain
smsgsend <addrFrom> <addrTo> <message>
smsgsendanon <addrTo> <message>
spork <name> [<value>]
stop
submitblock <hex data> [optional-params-obj]
validateaddress <SpectreSecurityCoinaddress>
validatepubkey <SpectreSecurityCoinpubkey>
verifymessage <SpectreSecurityCoinaddress> <signature> <message>