Bitcoin Core version 0.11.1 is now available from:

https://bitcoin.org/bin/bitcoin-core-0.11.1/

This is a new minor version release, bringing security fixes. It is recommended to upgrade to this version as soon as possible.

Please report bugs using the issue tracker at github:

https://github.com/bitcoin/bitcoin/issues

Upgrading and downgrading

How to Upgrade

If you are running an older version, shut it down. Wait until it has completely shut down (which might take a few minutes for older versions), then run the installer (on Windows) or just copy over /Applications/Bitcoin-Qt (on Mac) or bitcoind/bitcoin-qt (on Linux).

Downgrade warning

Because release 0.10.0 and later makes use of headers-first synchronization and parallel block download (see further), the block files and databases are not backwards-compatible with pre-0.10 versions of Bitcoin Core or other software:

- Blocks will be stored on disk out of order (in the order they are received, really), which makes it incompatible with some tools or other programs. Reindexing using earlier versions will also not work anymore as a result of this.
- The block index database will now hold headers for which no block is stored on disk, which earlier versions won't support.

If you want to be able to downgrade smoothly, make a backup of your entire data directory. Without this your node will need start syncing (or importing from bootstrap.dat) anew afterwards. It is possible that the data from a completely synchronised 0.10 node may be usable in older versions as-is, but this is not supported and may break as soon as the older version attempts to reindex.

This does not affect wallet forward or backward compatibility. There are no known problems when downgrading from 0.11.x to 0.10.x.

Notable changes

Fix buffer overflow in bundled upnp

Bundled miniupnpc was updated to 1.9.20151008. This fixes a buffer overflow in the XML parser during initial network discovery.

Details can be found here: http://talosintel.com/reports/TALOS-2015-0035/

This applies to the distributed executables only, not when building from source or using distribution provided packages.

Additionally, upnp has been disabled by default. This may result in a lower number of reachable nodes on IPv4, however this prevents future libupnpc vulnerabilities from being a structural risk to the network (see https://github.com/bitcoin/bitcoin/pull/6795).

Test for LowS signatures before relaying

Make the node require the canonical 'low-s' encoding for ECDSA signatures when relaying or mining. This removes a nuisance malleability vector.

Consensus behavior is unchanged.

If widely deployed this change would eliminate the last remaining known vector for nuisance malleability on SIGHASH_ALL P2PKH transactions. On the downside it will block most transactions made by sufficiently out of date software.

Unlike the other avenues to change txids on transactions this one was randomly violated by all deployed bitcoin software prior to its discovery. So, while other malleability vectors where made non-standard as soon as they were discovered, this one has remained permitted. Even BIP62 did not propose applying this rule to old version transactions, but conforming implementations have become much more common since BIP62 was initially written.

Bitcoin Core has produced compatible signatures since a28fb70e in September 2013, but this didn't make it into a release until 0.9 in March 2014; Bitcoinj has done so for a similar span of time. Bitcoinjs and electrum have been more recently updated.

This does not replace the need for BIP62 or similar, as miners can still cooperate to break transactions. Nor does it replace the need for wallet software to handle malleability sanely[1]. This only eliminates the cheap and irritating DOS attack.

[1] On the Malleability of Bitcoin Transactions Marcin Andrychowicz, Stefan Dziembowski, Daniel Malinowski, Łukasz Mazurek http://fc15.ifca.ai/preproceedings/bitcoin/paper_9.pdf

Minimum relay fee default increase

The default for the `-minrelaytxfee` setting has been increased from `0.00001` to `0.00005`.

This is necessitated by the current transaction flooding, causing outrageous memory usage on nodes due to the mempool ballooning. This is a temporary measure, bridging the time until a dynamic method for determining this fee is merged (which will be in 0.12).

(see https://github.com/bitcoin/bitcoin/pull/6793, as well as the 0.11 release notes, in which this value was suggested)

0.11.1 Change log

Detailed release notes follow. This overview includes changes that affect behavior, not code moves, refactors and string updates. For convenience in locating the code changes and accompanying discussion, both the pull request and git merge commit are mentioned.

- #6438 `2531438` openssl: avoid config file load/race
- #6439 `980f820` Updated URL location of netinstall for Debian
- #6384 `8e5a969` qt: Force TLS1.0+ for SSL connections
- #6471 `92401c2` Depends: bump to qt 5.5
- #6224 `93b606a` Be even stricter in processing unrequested blocks
- #6571 `100ac4e` libbitcoinconsensus: avoid a crash in multi-threaded environments
- #6545 `649f5d9` Do not store more than 200 timedata samples.
- #6694 `834e299` [QT] fix thin space word wrap line break issue
- #6703 `1cd7952` Backport bugfixes to 0.11
- #6750 `5ed8d0b` Recent rejects backport to v0.11
- #6769 `71cc9d9` Test LowS in standardness, removes nuisance malleability vector.
- #6789 `b4ad73f` Update miniupnpc to 1.9.20151008
- #6785 `b4dc33e` Backport to v0.11: In (strCommand == "tx"), return if AlreadyHave()
- #6412 `0095b9a` Test whether created sockets are select()able
- #6795 `4dbcec0` net: Disable upnp by default
- #6793 `e7bcc4a` Bump minrelaytxfee default

Credits

Thanks to everyone who directly contributed to this release:

- Adam Weiss
- Alex Morcos
- Casey Rodarmor
- Cory Fields
- fanquake
- Gregory Maxwell
- Jonas Schnelli
- J Ross Nicoll
- Luke Dashjr
- Pavel Janík
- Pavel Vasin

- Peter Todd
- Pieter Wuille
- randy-waterhouse
- Ross Nicoll
- Suhas Daftuar
- tailsjoin
- 𝔹tcDrak
- Tom Harding
- Veres Lajos
- Wladimir J. van der Laan

And those who contributed additional code review and/or security research:

- timothy on IRC for reporting the issue
- Vulnerability in miniupnp discovered by Aleksandar Nikolic of Cisco Talos

As well as everyone that helped translating on Transifex.