Adventures and Experiments Adding Namecoin to Tor Browser

Jeremy Rand
Lead Application Engineer, The Namecoin Project
https://www.namecoin.org/

OpenPGP: 5174 0B7C 732D 572A 3140 4010 6605 55E1 F8F7 BF85

Presented at 36C3 Monero Assembly / Critical Decentralization Cluster

This is not a talk by The Tor Project

- Everything in this talk only represents me.
- I’m coming at this from the perspective of a Namecoin developer.
- The perspective of the Tor developers may differ from mine. That’s okay!

Onion services are cool

- Built-in encryption, authentication, and censorship resistance.
  - Security doesn’t depend on a trusted third party (e.g. certificate authorities).
- But… the UX has a problem.

Onion services are cool… Except for this

- http://7fa6xlti5joarlmkuhjaifa47ukgcwz6tfndgax45ocyn4rixm632jid.onion/
- Impossible for humans to remember.
- Impossible for humans to recognize.
  - Humans will often not check the entire address.
  - Result: phishing attacks.

A brief introduction to Namecoin

- Like the DNS, but secured by a blockchain.
- Uses the “.bit” top-level domain.
- Names are represented by special coins.
- First project forked from Bitcoin (in 2011; Bitcoin was created in 2009).

Namecoin as a naming layer for onion services

- Recognized by Namecoin developers as a potential use case for Namecoin (very early in Namecoin history).
- Provides global, decentralized names (like .onion) but also human-meaningful (e.g. federalistpapers.bit).
- Mapping .bit (Namecoin) domains to .onion domains would solve the onion services UX issue.

Namecoin as a naming layer for onion services

- Experimental implementations date back to July 2011 (NmcSocks by itsnotlupus).
- Experiments continued for years.
- Some discussions happened with Tor developers, not much adoption materialized.
- See my 34C3 talk for more background.

Signs of renewed Tor interest

- By chance I ran into a Tor Browser developer (Arthur Edelstein) on Twitter in October 2018.
  - Because I subscribe to an RSS feed of Twitter search results for “Namecoin”.

Arthur Edelstein @arthuredelstein · 24 Oct 2018

I'm interested in making onion sites easier to use by making the domain names memorable. Different approaches have different properties which I'm trying to explore. DNS, namecoin, GNS, OnioNS, AltSvc, Location, Onion-Location are all interesting to me as possible approaches.

2

Jeremy Rand #FreeAssange
@biolizard89

Follow

Replying to @arthuredelstein @AlecMuffett and 2 others

Hi Arthur, I'm the Namecoin dev who prototyped Namecoin naming for onion services. I'm definitely still interested in this, would be cool to chat about moving forward.

4:09 AM - 24 Oct 2018

1

2018 discussions with Arthur from Tor

- Previous Tor discussions had focused on helping users experiment with installing Namecoin into Tor Browser themselves.
  - E.g. “Proposal 279” pluggable naming API.
- This time was different.

Arthur Edelstein @arthuredelstein · 23 Oct 2018

I think modularity is good, but I don't think waiting for the "sands of time" to pick a winner is the right approach. Instead @torproject should start implementing and deploying resolvers in tor and Tor Browser. Then we can learn and iterate.

1 1

Likely criteria (and non-criteria) that Arthur identified

- Tor recognized that the non-meaningful names of .onion domains were a massive UX and security problem.
- To the point that they were willing to compromise on the security model for a fix.
- In particular, anonymity for name owners was not a short-term requirement as long as we had a plan to deal with it later.

The big stickler: performance/scalability

- Arthur’s suggested performance goal:
  - User launches a fresh Tor Browser install.
  - User immediately types a .bit domain into address bar.
  - Delay in loading the .bit website shouldn’t be noticeably more than what you would expect from a .onion website (given random variation in speed of building circuits to onion services).

Is near-instant Namecoin resolution possible?

- My mental reaction: “No way is that happening.”
- What I (approximately) said: “That’s going to be really hard, but there are a lot of optimizations we could be doing that we aren’t yet. I’ll do some experiments and see what we can manage.”
- General rule: don’t tell people on the spot that something is impossible; actually look into it first.

Enter Electrum-NMC

- Ahmed Bodiwala and I were being funded by NLnet and Cyphrs to port the lightweight Bitcoin wallet Electrum to Namecoin.
- The port had only recently become a thing when Arthur and I talked.
- Based on talking to Arthur, it became clear that the embryonic Electrum-NMC was our best bet at achieving the performance needs that Arthur outlined.

Diverting funding to Electrum-NMC

- NLnet had allocated funding for us to spend on a different lightweight name lookup client, ConsensusJ-Namecoin.
  - ConsensusJ-Namecoin’s design was not capable of achieving Arthur’s performance goals (~5 minute initial sync).
  - We decided to divert the NLnet ConsensusJ-Namecoin funding to focus on Electrum-NMC.
  - Kudos to NLnet for giving us that flexibility!

The state of Electrum-NMC in October 2018

- Initial syncup downloaded 672 MB.
- Took around 6 minutes on a Talos II workstation (very fast CPU) without Tor.
- This was needed before name lookups could be performed.
- Nowhere near meeting Arthur’s performance goal.

Checkpoints

- Electrum (for Bitcoin) encodes checkpoints as a list of every 2016th block hash.
  - It only downloads the headers postdating the last checkpoint hash on startup.
  - If you need to validate a transaction that predates the last checkpoint hash, Electrum downloads the chunk of 2016 headers between the two checkpoints that surround that transaction.

Checkpoints in Namecoin

- Unexpired names can be anywhere in the last 36 kiloblocks.
  - So if we want to look up names quickly, we need to already have the last 36 kiloblocks' headers.
  - So we can't set a checkpoint more recent than 36 kiloblocks ago.
  - Setting a checkpoint 36 kiloblocks ago dropped the syncup download usage to 66 MB (compared to 672 MB without a checkpoint).

On-demand header download

- What if we *did* set a checkpoint more recent than 36 kiloblocks?
  - Good: Drops the initial syncup download to 4.9 MB.
  - Bad: When looking up a name whose header we hadn’t downloaded yet, we’d need to download 2016 headers.
  - That’s 3.2 MB, downloading **during** a name lookup.
  - Is this improvable?

Merged mining and checkpoints

- Namecoin block headers consist of 2 parts:
  - Bitcoin-style block header (80 bytes).
  - Merged mining header (variable length, often ~10 KB).
- Both parts are needed to verify proof of work.
  - If you have some other method to verify that a header is valid, you don’t need the merged mining header.
  - Guess what! A checkpoint is such a method.

Removing merged mining headers from checkpointed headers

- Since a checkpoint can verify a block header without needing the merged mining header, we can just not download the merged mining header.
- Drops the size of an on-demand chunk (of 2016 headers) from 3.2 MB to 323 KB.
- This is getting closer to the realm of usability for Tor, but can we do better?

Merkle Checkpoints

- The Electrum protocol supports a 2nd checkpoint format.
  - The client checkpoint is just a Merkle root of all headers prior to the checkpoint height.
  - The server provides a Merkle branch proof when a header is requested.
  - You can download a single header at a time and still connect it to the checkpoint.
  - Merkle branch proofs are logarithmic in size – much better scalability.

Merkle Checkpoints for Electrum-NMC

- The only client implementation of Merkle Checkpoints was Electron-Cash (the BCH fork of Electrum).
  - I ended up porting that implementation to Electrum for Bitcoin.
  - And then merged it to Electrum-NMC.
  - Dropped from 323 KB to under 2 KB. Now we’re talking!

Parallelized Blockchain Download

- Electrum typically only downloads headers from one server at a time.
  - I patched it to download from multiple servers in parallel.
  - Improves initial syncup time considerably.

Binary size requirements

- “The Tor Browser download is on the order of 60-80 MB (after compression). So adding a few megabytes is probably acceptable, but 10 megabytes will probably be too much.” – Arthur
  - Namecoin’s Tor Browser integration consisted of: Electrum-NMC, ncdns, ncprop279, TorNS, txtorcon.
  - Uh oh. These components totaled to 39.7 MB.

Optimizing binary size: Stripping unneeded features

- I stripped the GUI, plugins, payment protocol, and wallet code from Electrum-NMC.
- Also stripped a lot of TLS and DNS code from our Go codebases (ncdns and dns-prop279).
- All of these features are now optional at build time.

Optimizing binary size: Avoiding redundant static libraries

- Go binaries are statically linked (usually).
- Go runtime is massive; gets statically linked into every binary.
- Combining ncdns and dns-prop279 tools into a single specialized tool (ncprop279) avoided static library redundancy.

Optimizing binary size: Tor controller library

- We were using the txtorcon library to interact with the Tor control port API.
  - Because that's what meejah's TorNS example code did.
  - txtorcon has lots of dependencies.
- Refactored TorNS to use Stem instead of txtorcon (much smaller).
  - Stem has no additional dependencies, and is generally smaller.
- Kudos to Jesse Victors of OnioNS for some Stem example code.

Optimizing binary size: Final results?

- Reduced Namecoin’s impact on binary size from 39.7 MB to 3.3 MB.
  - Still a significant impact, but no longer a horrifying dealbreaker.

Proposing it to the other Tor devs…

- Once Arthur’s performance goals were met, we scheduled a demo for several other Tor developers, 2019 Apr 26.
  - Including Georg Koppen (lead Tor Browser developer).
  - The response was cautious but optimistic.

Tor’s review culture is excellent

- At one point I was asked: “Assuming you'd have to argue against including Namecoin support in Tor Browser, which arguments would you bring up?”
  - More FLOSS projects should be asking this question to new contributors making proposals.

Limiting the scope

- Everyone agreed that this is an experiment, and therefore the scope should be extremely narrow.
  - No resolution to IP addresses (only onion services).
  - No TLS validation.
  - In Nightly builds only.
  - In GNU/Linux only.
  - Disabled by default.

Why limit the scope?

- This limits the risk to users, and facilitates quicker review.
- If the experiment goes well, we can always expand the scope later.
- No commitment from Tor to keep the code there or to advance the experiment.

Stream isolation

- Anonymity for name owners wasn’t a requirement.
  - But anonymity for people *viewing* websites definitely is.
  - A critical Tor feature is stream isolation – isolates traffic from different activities on different Tor circuits.
  - Makes users anonymous rather than pseudonymous.

Stream isolation in Namecoin

- All of Namecoin’s Tor integration components, and many of their libraries, needed patching to properly handle stream isolation.
  - Side benefit: stream isolation + parallelized blockchain download == downloading headers over multiple Tor circuits (no more speed bottlenecks from a bad Tor circuit).
  - The tor daemon and Firefox also needed patches to ensure stream isolation worked with Namecoin.

What remained...

- Code cleanup.
- In-person meeting with Tor devs in Stockholm, 2019 July 11-15.
- Reproducible builds (and integration in Tor Browser’s build system).
- Handed off to the Tor devs for review 2019 Nov 11.
- After lots of review cycles (and me addressing each piece of review), finally merged on 2019 Dec 18.
- First Tor Browser Nightly to support Namecoin was 2019 Dec 20.

Want to try it out?

- Download a Tor Browser Nightly for GNU/Linux.
- Run it with the environment variable TOR_ENABLE_NAMECOIN=1
- Namecoin will then be used to resolve .bit and .bit.onion addresses.
  - Currently both suffixes are equivalent; in the future, .bit might also resolve to IP addresses with TLS.

Example domains to try

- http://federalistpapers.bit/
- http://onionshare.bit/
- http://riseuptools.bit/
- http://submit.theintercept.bit/
- http://submit.wikileaks.bit/
- (All of these are owned by Namecoin supporters who are happy to donate them to the “rightful” owners.)

So what's next? (1)

- Support Windows and macOS.
  - Will require some refactors, due to GNU/Linux-specific assumptions in the Namecoin Tor Browser integration.
  - Building Python for Windows reproducibly will be an interesting adventure.
- Support Android/Linux.
  - Lots of uncharted waters here.

So what's next? (2)

- More latency / bandwidth / size improvements.
  - Cut name lookup round trips from 2 to 1 via API changes.
  - Cut bandwidth in half via CBOR encoding instead of JSON.
  - Cut binary size by merging Namecoin's Go binary with the pluggable transport binaries (via Go-Busybox from u-root project).
- Electrum-NMC GUI improvements for name owners.
  - User selects "Tor" and enters their .onion domain.
  - No need to construct JSON manually.

So what’s next? (3)

- Anonymity for name owners.
  - Via integration with Monero and Bisq.
  - See my 34C3 talk.
- Support IP addresses (not just onion services) and TLS.
- Support Whonix and Tails (currently broken due to control port filtering).

So what's next? (4)

- Improve blockchain validation.
  - Verify ECDSA signatures for names (not just PoW signatures).
  - Support authenticated nonexistence proofs.
  - Maybe find a way to use actual full nodes with Tor Browser?
    - Not entirely out of the question.

So what's next? (5)

- And of course… listening to feedback from the Tor community.
  - Not clear yet what Tor's criteria would be for advancing from Nightly to Alpha or Stable, or enabling by default.
  - Tor devs may want things not on this list.
  - Tor devs may not want things that are on this list.
  - It's Tor Browser, not Namecoin Browser – the Tor devs get the final say. We'll honor their requests.

Workshop coming up!

- Come try it out for yourself!
  - Bring a GNU/Linux machine (VM or bare metal are both fine).
- Please tell me what you think.
  - It’s an experiment – any good experiment needs feedback.
- Right after this talk, in the Critical Decentralization Cluster Workshop Area.

Huge thanks to our funders

- NLnet Foundation’s Internet Hardening Fund
- Netherlands Ministry of Economic Affairs (via Internet Hardening Fund)
- Cyphrs

Contact Me At...

- https://www.namecoin.org/
- OpenPGP (after Congress ends):
  5174 0B7C 732D 572A 3140 4010 6605 55E1 F8F7 BF85
- jeremyrand@airmail.cc (after Congress ends)
- byronlelah@airmail.cc (during Congress; no OpenPGP)
- Or just find me here at the Congress! (The Namecoin logo on my shirt should help you find me.)