Hidden Service Naming and TLS Cert Checking with Namecoin

Jeremy Rand
E-mail: biolizard89 AT gmail DOT com
IRC #tor-dev nick: Jeremy_Rand

What project would you like to work on? Use our ideas lists as a starting point or make up your own idea. Your proposal should include high-level descriptions of what you're going to do, with more details about the parts you expect to be tricky. Your proposal should also try to break down the project into tasks of a fairly fine granularity, and convince us you have a plan for finishing it. A timeline for what you will be doing throughout the summer is highly recommended.

Note that we might share which project ideas have strong applications in order to spread applicants out (it's bad for everyone for several strong applicants to be for the exact same project).

I'm interested in enabling Tor hidden services to have human-readable names. I think Namecoin [1] would be an excellent backend for this application. The Namecoin domain name specification [2] includes a feature allowing Namecoin domains (.bit domains) to point to a .onion domain instead of an IPv4/IPv6 address. I've written a proof-of-concept of this for Firefox (using Moxie Marlinspike's Convergence tool as a base; see the code samples section below). Using this, users can visit http://federalistpapers.bit/ to access the Federalist Papers hidden service. I would like to port this functionality to TorBrowser, improve its privacy (including making it support Tor's stream isolation), and make it work with non-HTTP applications (since Tor is fundamentally a generic SOCKS proxy).

Namecoin also has the useful feature of allowing TLS fingerprints to be embedded in the blockchain, which eliminates the need to trust certificate authorities for HTTPS websites. My Convergence-based Firefox extension was also the first implementation of this feature. Since malicious exit nodes interfering with TLS certificates is a known potential problem for the security of Tor users, Namecoin would "kill two birds with one stone," addressing both hidden service naming and TLS MITM attacks with one solution. (This also improves the security of Whonix-hosted .onion domains, which frequently would benefit from HTTPS [3], but don't currently have a way of verifying certificate fingerprints.)

I think the best way of implementing the desired improvements would be to modify my Convergence-based Firefox extension to implement a SOCKS proxy instead of an HTTP/HTTPS proxy as it does currently. The SOCKS proxy would pass-through its SOCKS authentication parameters to the Tor SOCKS proxy, which would enable stream isolation to work as desired. The SOCKS proxy would run on a predictable port, and TorBrowser would be configured to send traffic to that SOCKS port instead of the Tor SOCKS port. (The easiest way to do this without modifying TorBrowser is, I assume, changing the Tor SOCKS port to something new and using the previously existing port for the proxy.)

The two Namecoin support softwares which are required, namecoind and nmcontrol, shouldn't be very hard to make work. namecoind (which manages the blockchain) already supports SOCKS proxies (and so could be tunneled through Tor), and nmcontrol (which parses the blockchain data according to the DNS spec) has a privacy mode which prevents it from making external requests. Making namecoind work in "lite mode" (not requiring the full blockchain) is outside the scope of this project.

I believe that this would be sufficient to allow using named hidden services in TorBrowser.

After this is done, I would try to port the Firefox extension to XULRunner, which would allow named hidden services to be used without needing Firefox or TorBrowser to be running. I am aware that XULRunner typically requires a GUI environment; my understanding is that XPCShell (which is included in XULRunner distributions) would allow running this code entirely from the command line (as might be desired in server environments).

As an additional goal (which is probably lower priority), adding functionality to nmcontrol to handle round-robin load balancing in a non-fingerprintable way would be useful: the cached choice of IP addresses needs to change for different Tor circuits. Also, improving the compatibility of nmcontrol's privacy mode for non-.onion domains would be helpful (right now it simply refuses to load any .bit website whose IP address isn't in the blockchain); this could be improved by substituting in ICANN domains which are specified in the blockchain, or possibly by using DNS over TCP over Tor to instead of DNS over UDP.

I have a strong understanding of how Namecoin's DNS works (I'm a significant contributor to the Namecoin DNS specification, as well as one of the leaders of the Namecoin project), and I've also developed some projects for Namecoin (including the aforementioned Convergence-based Firefox extension, a dynamic DNS client, and a hashrate distribution calculator). So I think I'm a good fit for this project.

Here are some questions I've received from Tor developers about this project, and my answers:

Q: Isn't Namecoin vulnerable to 51% attacks?
A: Namecoin by itself can have its history rewritten by a 51% attack, and at the moment it is plausible that a well-funded adversary such as a government could launch a 51% attack. (Based on the price of mining hardware, my very imprecise calculations are that approximately 1 billion USD could fund a 51% attack right now.) But a checkpoint system (covering block hashes and transaction hashes), combined with reorganization notifications, would make this attack detectable (and provide a way to determine the “correct” blockchain given a trusted third party). While checkpoints introduce a degree of centralization, Tor itself is already somewhat centralized. Namecoin with checkpoints is less centralized than the alternative proposal of having The Tor Project directly generate a database of names. While I'm not particularly familiar (yet) with how checkpointing is performed within Namecoin's block validation system, I do know how to at least verify whether the currently loaded blockchain matches a given checkpoint (which would at least alert users that an attack had taken place).

Q: Isn't domain squatting an issue with Namecoin?
A: Domain squatting is known to be an issue, and there are proposals to adjust the name pricing structure of Namecoin to disincentivise squatting. While these proposals are not implemented at the moment, I think it's likely that they will be implemented in the future.

Q: What about the blockchain download size?
A: “Lite clients” don't yet exist for Namecoin, but they can definitely be built. The UTXO lite client [4] which is being implemented for Bitcoin should be mergeable to Namecoin in the future.

Q: Is lack of revocation a problem?
A: There is a workaround (recently implemented) for a specific use case of revocation: a Namecoin name can import data from a second Namecoin name, in such a way that one name can be held in a safe

location while the other name would be easier to update (but overrideable by the first name). So if the easy-to-update name has its keys compromised, the safely-stored name can recover the situation. This doesn't solve the more general revocation problem. I believe that it will be possible to add more general revocation support to Namecoin in the future, but the Namecoin developers have not yet come to a consensus on the best way to implement this.

Q: Namecoin isn't anonymous; is this a problem?
A: At the moment, I think the easiest way to get mostly-anonymous namecoins is to start with bitcoins, run them through a mixer (I believe Bitcoin Fog [5] is popular, although I haven't used them), and then use the anonymized bitcoins to purchase namecoins on an exchange. (Most exchanges don't ask for identification unless you're depositing or withdrawing government-issued currency.) Some exchanges may block Tor, in which case using a VPN between Tor and the exchange may be necessary. While this solution isn't optimal, it is workable in my opinion. Short-term improvements would involve using CoinJoin [6] to anonymize bitcoins without a risk of theft. The CoinJoin concept is also easily applicable to Namecoin directly if a developer wanted to do that. Long-term, projects like Zerocoin/Zerocash [7] look like a good option, and there is interest among the Namecoin developers in integrating Zerocoin/Zerocash into Namecoin.

Estimated Timeline:

May 10 (day after finals at my university) – End of Community Bonding Period:

* Play around with TorBrowser (embarrassingly I haven't used TorBrowser since approximately a year ago, so I'll benefit from a refresher).
* Review the SOCKS protocol spec (there's already some SOCKS code by Moxie in Convergence, but I suspect that it needs work to function properly with current Firefox releases).
* Play around with XULRunner (I've never used it, so this'll be a good chance to gain some experience with it).

First week:

* Install the existing Namecoin Convergence extension in TorBrowser.
* Hardcode its proxy port, and try to connect to that proxy via some other web browser to verify that the proxy is still functioning in TorBrowser. (If this fails, some time will be spent debugging, but it seems likely given my knowledge of how Convergence works that it will not fail.)
* Begin switching the Convergence proxy code to use the built-in SOCKS proxy instead of HTTP.
* Start fixing the many errors that will almost certainly occur because the SOCKS proxy in Convergence wasn't tested since Firefox versions from 3 years ago.

Next 2 weeks:

* Get the SOCKS proxy to run without errors for standard TCP connections. (This is probably a lot of work, but maybe I'll get lucky and it will take less time.)
* Make the SOCKS proxy handle TLS verification via nmcontrol's RPC interface (as the HTTPS proxy does now).
* Make the SOCKS proxy forward to another SOCKS proxy (e.g. Tor), as the HTTP proxy does now.
* Make the SOCKS proxy do Namecoin DNS resolution (to IP addresses and Tor hidden services) via nmcontrol's RPC interface, as the HTTP proxy does now.

Next week:

* Implement SOCKS authentication in the Convergence proxy, so that it passes those through to the Tor proxy.

Next week:

* Change the Tor SOCKS port, set the Convergence SOCKS port to the old Tor port, and see if TorBrowser can browse sites with the Convergence SOCKS server.
* Test stream isolation (make new tabs in TorBrowser, see if new circuits are generated as they should be).
* Make sure that namecoind and nmcontrol work well with Tor and privacy mode, respectively.
* If any of these tests fail, spend time fixing them. (Murphy's Law predicts that this fixing will take the whole week – hopefully not longer – but if I'm lucky enough to evade Murphy's Law, I'll start on the next week's stuff early.)

Midterm point.

Next 3 weeks:

* Port the Firefox extension to XULRunner. I hope this doesn't take 3 weeks, but since I don't have experience with XULRunner, it's entirely possible that I'll run into major issues that take time to fix.

Next week:

* Implement a basic alert system in nmcontrol, so that if a block doesn't match a given hash, or if a given transaction ID disappears from the blockchain, or if a large blockchain reorganization occurs, nmcontrol's RPC interface will notify the caller that something's wrong. This would give the capacity to detect many 51% attacks.
* Implement a UI to this alert system in the XULRunner application.

Next week:

* Work with TorBrowser developers on packaging/building issues so that this work can be used in the real world.
* (The final choice of whether to use this in the real world is of course up to the TorBrowser devs, but I'd like to make sure that packaging/building isn't a factor in choosing not to use it.)

Next week:

* July 29 – August 4 I will be attending a conference and will be unable to work on GSoC. (See section below.)

Final week:

* Finish any tasks that took longer than expected. (Murphy's Law suggests that there will be something; hopefully not a lot.)
* If free time is available, improve nmcontrol's privacy mode to allow .bit domains to point to ICANN domains instead of IP addresses; this will dramatically improve compatibility with non-.onion .bit

websites while using Tor (and still maintain excellent security since the TLS fingerprint in the blockchain will still prevent DNS hijacking from succeeding).
* If free time is still available, implement round-robin balancing for .bit domains, and make it reset when new identities are generated to prevent fingerprinting. I will need to get feedback on this from the developer community to make sure that I'm not forgetting some kind of attack.
* If there's still free time here, I'll try to implement XPCShell support so that the code can be used without any GUI (including on Linux without an X server). I currently have no experience with XPCShell, so this might be easy or it might be quite difficult.

"Pencils-Down" date.

Remaining week:

* Tie up loose ends, finish any documentation that wasn't already written.

References:

[1] http://www.namecoin.info
[2] https://github.com/namecoin/wiki/wiki/Domain-Name-Specification-2.0
[3] https://www.whonix.org/wiki/Hidden_Services#Notes_about_End-to-end_security_of_Hidden_Services
[4] http://utxo.tumblr.com/
[5] http://fogcore5n3ov3tui.onion/
[6] https://bitcointalk.org/index.php?topic=279249.0
[7] https://www.youtube.com/watch?v=FXU65XsLiFk

Point us to a code sample: something good and clean to demonstrate that you know what you're doing, ideally from an existing project.

* Convergence-based Firefox extension for Namecoin DNS, including TLS verification and Tor/I2P service resolution. Original extension by Moxie Marlinspike; I wrote all of the Namecoin-related code.
https://github.com/JeremyRand/Convergence
* Dynamic DNS client for Namecoin DNS. Entirely my work.
https://github.com/JeremyRand/DyName
* Hashrate distribution calculator for Namecoin, using various public API's. Currently used as the backend for the graph on https://www.namecoin.org/. Entirely my work.
https://github.com/JeremyRand/NamecoinHashrateDistribution
* GreaseMonkey script to index visited websites with the YaCy P2P search engine. Entirely my work.
https://github.com/JeremyRand/YaCyIndexerGreasemonkey
* Custom firmware for the CBC Chumby Botball Robotics Controller. Based on firmware by KIPR.org; modifications by Matthew Thompson and myself.
https://github.com/JeremyRand/cbc
* Custom firmware for the XBC Xport Botball Robotics Controller. Based on firmware by Charmed Labs and KIPR.org; modifications by Farz Hemmati and myself.
https://github.com/JeremyRand/XBC-Firmware

Why do you want to work with The Tor Project in particular?

I think anonymity, privacy, and freedom from censorship are vital prerequisites to free speech and democracy. As a computer science major, I feel that it is my civic duty to contribute my programming skills to projects such as Tor which support free speech and democracy. I have followed Tor at least since I was a high school sophomore, and I ran a Tor bridge at my house for several years (until I had to stop because I no longer had a 24/7 Internet-connected computer). I'm sufficiently enthusiastic about Tor that my college admission and scholarship essays all mentioned that my dream job was to work at an organization such as Tor. When I took a public speaking class in 2013, my final speech was on "why you should run a Tor bridge."

Tell us about your experiences in free software development environments. We especially want to hear examples of how you have collaborated with others rather than just working on a project by yourself.

Since summer 2012, I've been involved with the Namecoin project, becoming one of the lead developers in summer 2013. I've written enhancement proposals for other developers to review, and evaluated proposals by other developers. I thrive in a collaborative environment; I enjoy bouncing ideas off of other people and having them bounce ideas off me. I think this kind of environment produces better code than any single person could produce. At the same time, I enjoy specializing into specific areas of expertise; I think if everyone has identical skills, then people's time is wasted.

My interest in collaborating with other developers goes back to high school, when I participated in the Botball robotics competition. In January of my sophomore year of high school, a new robotics controller was released for Botball, and its quality was clearly rushed. I sought out a collaborator from another high-school team (halfway across the country), and we started reverse-engineering the controller and developing improvements. We worked excellently as a team; our skill sets complemented each other, and we progressed quickly. In June, we submitted the longest ever student paper (14 pages, single-spaced) in Botball history to the Botball-affiliated Global Conference on Educational Robotics, describing what we had improved, and we released the source code. At the conference in July, the original developers of the new controller (several of whom had previously worked at NASA) told us that they were very impressed with our improvements, and that they would be integrating some of our code into the next year's release. This success never would have happened if I hadn't organized a collaboration with the other student.

Will you be working full-time on the project for the summer, or will you have other commitments too (a second job, classes, etc)? If you won't be available full-time, please explain, and list timing if you know them for other major deadlines (e.g. exams). Having other activities isn't a deal-breaker, but we don't want to be surprised.

July 29 – August 4 I will be attending the 2014 Global Conference on Educational Robotics in Los Angeles with my family. My brother's 2012 and 2013 international championship team is competing in a robotics tournament there, and this is his last year competing, so I would feel like a jerk skipping it. I'll also probably be giving a brief talk there, and will be volunteering with the live-streaming of the conference/tournament. I sincerely hope this isn't a dealbreaker.

I believe the rest of my summer is completely free.

Will your project need more work and/or maintenance after the summer ends? What are the chances you will stick around and help out with that and other related projects?

The project will probably require maintenance after the summer ends, as Namecoin's core software receives updates. I am definitely interested in sticking around, assuming that my financial situation permits. (i.e. feeding myself is a higher priority than maintaining this code, but hopefully I'll be able to do both.)

What is your ideal approach to keeping everybody informed of your progress, problems, and questions over the course of the project? Said another way, how much of a "manager" will you need your mentor to be?

Weekly blog posts seem a reasonable way to keep everyone informed on a high level. If I encounter issues which need immediate resolution (e.g. I'm stuck on something), asking on #tor-dev IRC should be suitable, and asking on the tor-dev mailing list would be a good backup if no solution is found on IRC. I think I probably won't need a lot of direct management from my mentor, although any help he/she can provide would certainly be appreciated.

What school are you attending? What year are you, and what's your major/degree/focus? If you're part of a research group, which one?

I'm a 3rd-year student at University of Oklahoma, majoring in computer science.

How can we contact you to ask you further questions? Google doesn't share your contact details with us automatically, so you should include that in your application. In addition, what's your IRC nickname? Interacting with us on IRC will help us get to know you, and help you get to know our community.

See top of this document.

Are you applying to other projects for GSoC and, if so, what would be your preference if you're accepted to both? Having a stated preference helps with the deduplication process and will not impact if we accept your application or not.

I am not applying for any non-Tor projects for GSoC. I might apply for the hidden service search engine project with Tor as well; as of this writing I'm still making up my mind on that. If I'm accepted to both, I would prefer the hidden service naming project (this one) over the hidden service search project.

Is there anything else that we should know that will make us like your project more?

I have considerable experience in working independently on significant projects, which makes a difference in this project since I believe most Tor developers don't know a lot about Namecoin. Starting in my senior year of high school, I enrolled in an independent study class at University of Oklahoma (credited towards my Honors College thesis), where I worked on video game enhancement research (a project which I had come up with on my own). The project involved patching offline GameCube and Wii games to play online. The professor who supervised the research knew nothing about GameCube/Wii hardware or about game enhancement or reverse-engineering in general; all of the direction for the project was provided by me. During my sophomore year in college, I presented some of the resulting research at OU's graduate CS conference; my work received the 2nd place award, even though it was going against grad students. I believe this demonstrates that I am skilled at independent work, and that lack of a mentor's Namecoin expertise will not be a problem for me.