Q1 project timeline (April 1, 2016 – June 30, 2016): activity summary

Pluggable transport integration

Project objectives

Outputs for the quarter

Task 1 (SponsorT, aka the work we did via the Internews contract):

Task 2 (SponsorS, aka this direct contract): Maintain and extend obfsproxy, obfs4proxy, obfsclient and other Pluggable Transport codebases as needed, and assist developers and researchers who wish to use Tor frameworks to do relevant research.

Task 3 (SponsorS): Tor side pluggable transport related (and other) improvements.

Task 4 (SponsorS): Pluggable transport R&D (catchall)

Challenges

Testing and network simulation improvements

Project objectives
Outputs for the quarter

III. Results and recommendations

1. Testing is helpful (big surprise).

Challenges

Enhanced outreach

Project objectives

Outputs for the quarter

Direct work with organizations

Outreach materials

Media strategy:

Social media

We tried something new this quarter, we initiated a rapid response for censorship events to alert users they can use Tor to circumvent it.

Challenges

Press

Next quarter’s planned activities

Pluggable Transport integration

Testing and network simulation improvements

Enhanced outreach
Q1 project timeline (April 1, 2016 - June 30, 2016): activity summary

Pluggable transport integration

Project objectives

The goal of this project is to separate Tor's anonymity and privacy properties from its censorship-resistance properties: the core Tor software focuses on building Tor circuits and getting the multi-layer encryption right, while the transport layer focuses on preventing an attacker from recognizing or blocking the client's connections to the rest of the Tor network.

This modular approach lets us "plug in" new transports as needed; plus, since the transport layer is a separate program, it can be written in whatever rapid prototyping language is most convenient, allowing Tor to adapt much more quickly to a censor's new Deep Packet Inspection (DPI) tactics without needing to touch the core Tor program at all.

Outputs for the quarter

Task 1 (SponsorT, aka the work we did via the Internews contract):

We finished our 6 month extension contract with Internews.

For this extension we decided to focus on communication with the broader community. We identified 3 audience types:

- PT end user;
- PT bridge operator;
- PT developer;
For the end user we created an easy “how to” graph and animated gif. This work goes along with the animation video we are doing for our contract with DRL and aims to help the Awareness phase of the user lifecycle:

**Awareness** -> **Onboarding** -> **Retention** -> **Resurrection**

**Awareness** is the moment where the user learns about the product or feature and decides to try it, which leads to the **onboarding** moment, where you are teaching the user how the product works. With that in mind we are teaching the user why they might need a PT and how to use it:
How To graph (download original) -animated gif can be found here.

For the PT operators, we created a section for them with tutorials on how to run different PT bridges.

We also created a section for developers, in which we linked to all the documentation produced last year:

- Pluggable Transports Evaluation Guidelines
- Tech Spec
- Guidelines for deploying a PT on Tor Browser

And finally organized all PTs by their status:

- Deployed on Tor Browser
- Deprecated from Tor Browser - removed from Tor Browser
- Undeployed Transports - exist but not deployed on Tor Browser
- Under development
We linked to PT’s evaluations done last year, following our Evaluation Guidelines, plus two new preliminary ones done under this extension:

- **Basket2 Evaluation** - previously named as obfs5
- **SnowFlake Evaluation**

Both transports are still under development, which is why we are calling them ‘preliminary’ evaluations.

**Task 2 (SponsorS, aka this direct contract):** Maintain and extend obfsproxy, obfs4proxy, obfsclient and other Pluggable Transport codebases as needed, and assist developers and researchers who wish to use Tor frameworks to do relevant research.

We decided to change the name of obfs5 to **Basket2** since it derives inspiration primarily from obfs4 and predecessors, and incorporates ideas initially prototyped in the experimental basket transport.

This transport features:

- Authentication, data integrity, and confidentiality.
- Active probing resistance.
- Passive fingerprinting resistance, improved over obfs4.
- Client driven dynamic negotiation of runtime padding to better suit various adversary models.
- Better separation between the handshake obfuscation and the authenticated key exchange mechanisms.
- Significantly improved link layer framing.
- Optional user authentication.
- Post-quantum forward secrecy.
- License switch from 3BSD to AGPL for more Freedom.

We are testing this transport internally and we are happy with the progress so far, however, there are still development tasks and documentation to be completed which we will carry on in the next quarter.

**Task 3 (SponsorS): Tor side pluggable transport related (and other) improvements.**

- **Tor Browser 6.0a5 - April 2016**
  - On this release we shipped with a new Tor alpha version, 0.2.8.2 (reported last quarter), which makes meek usable again and contains a number of other improvements/stability fixes.
- **Tor 0.2.8.3-alpha - May 2016**
  - Minor bugfixes (pluggable transports):
    - Avoid reporting a spurious error when we decide that we don’t need to terminate a pluggable transport because it has already exited. Fixes bug 18686; bugfix on 0.2.5.5-alpha.
- **Tor Browser 6.0 release - May 2016**
- Update meek to 0.22 (tag 0.22-18371-3)
  - Bug 18371: Symlinks are incompatible with Gatekeeper signing
  - Bug 18904: Mac OS: meek-http-helper profile not updated
- Tor Browser 6.5a1-hardened
  - Bug 18976: Remove some FTE bridges - requested by FTEproxy team.

Task 4 (SponsorS): Pluggable transport R&D (catchall)

Roger Dingledine and Nick Mathewson have contributed reviewing papers for the Privacy Enhancement Technology Symposium (PETS), which brings together privacy experts from around the world to present and discuss recent advances and new perspectives on research in privacy technologies.

We are very fortunate to have a great number of researchers who participate at PETS\(^1\) and are presenting papers related to Tor.

One that we would like to highlight which is related to censorship circumvention technology is the "Salmon: Robust Proxy Distribution for Censorship Circumvention" paper\(^2\), written by Frederick Douglas (University of Illinois Urbana-Champaign), Rorshach (unaffiliated), and Weiyang Pan and Matthew Caesar (University of Illinois Urbana-Champaign).

What is interesting with this paper is that it addresses an old design we had for bridge distribution where there is a ranking classification according to the user lifetime. New ones are allocated to a set of bridges only, and as they stay longer connected to a bridge they move up in this ‘trustable place’ in the rank.

This is an attempt to isolate malicious actors who are just trying to find out our bridges so they can censor them. With this rank, it would take longer for them to be given another set of bridges, keeping a greater part of the network safe from being blocked and also increasing the cost for the attacker.

Another big thing we did in the R&D front against censorship was to release the new "Fallback directories" design as a big feature for Tor 0.2.8 (which is now the stable release).

This will allow users in situations where the main Tor directory authorities have been blocked to able to automatically bootstrap into the Tor network using other locations.

\(^1\) [https://www.securityweek2016.tu-darmstadt.de/pets-2016/program/](https://www.securityweek2016.tu-darmstadt.de/pets-2016/program/)
Challenges

While writing our final report to Internews, we mentioned our reflections on one of the main pain points of the PT implementers meetings, which is to determine ownership of carrying on those discussions and driving them into actual decisions and implementations.

We know we play a big role in the ecosystem, therefore we want to help solve this problem and we presented what we can commit moving forward. Our goal is to host some of the discussions related to process at our Dev Meeting at the end of September.

Parts of the process that we believe should be owned by Tor:
- Changes to the PT specification
- Implementation of those changes in currently Tor-maintained PTs
- Any changes to Tor

For changes to the PT specification, we propose a process similar with our code/review process:
- For each additional specification document:
  - Determine who needs to give buy-in as ‘reviewers’:
    - (we assume) Orthogonal specs will need buy-in just from whoever plans to implement and use their functionality
    - (we assume) Additional-functionality specs will need buy-in from Tor, plus whoever plans to implement and use the functionality.
  - Review comments guidelines:
    - Give a number for each change in the current spec;
    - Discuss it;
    - Decide who will implement it;

Testing and network simulation improvements

Project objectives

Tor will improve the correctness and stability of the core Tor software by streamlining and automating the process of launching a complete test; designing and scripting an automated test suite to exercise and stress as much of Tor’s functionality as possible; and extending Tor’s controller interface to allow better monitoring.

Outputs for the quarter

We have been good on keeping up with the test coverage\(^3\) of our code base. And we continue to maintain and improve our testing infrastructure, as new requests or bugs comes out.

---

\(^3\) [https://people.torproject.org/~nickm/tor-auto/coverage/cov-check.html](https://people.torproject.org/~nickm/tor-auto/coverage/cov-check.html)
Tor 0.2.8.3-alpha release⁴:

- Major bugfixes (testing):
  - Fix a bug that would block 'make test-network-all' on systems where IPv6 packets were lost. Fixes bug 19008; bugfix on tor-0.2.7.3-rc.
  - Avoid "WSANOTINITIALISED" warnings in the unit tests. Fixes bug 18668⁵; bugfix on 0.2.8.1-alpha.

- Minor bugfixes (testing):
  - Allow directories in small networks to bootstrap by skipping DirPort checks when the consensus has no exits. Fixes bug 19003⁶; bugfix on 0.2.8.1-alpha. Patch by teor.
  - Fix a small memory leak that would occur when the TestingEnableCellStatsEvent option was turned on. Fixes bug 18673; bugfix on 0.2.5.2-alpha.

During this quarter we also released a report where we analyzed the bugs we saw on 0.2.5 or latter. We select high priority and high severity bug tickets as well as bugs listed on changelogs of the following releases.

Since this was the first time we ever did such analyses, we documented the methodology used for it together with the results on a blog titled: "Mid-2016 Tor bug retrospective, with lessons for future coding."

These analyses gave us some great insights and recommendations, and you can read the full post at our blog⁷. To avoid extending ourselves too long with this report, we will just add the part related to the impact of our Testing effort in this program:

III. Results and recommendations

1. Testing is helpful (big surprise).

We’ve believed for a while that we can reduce the number of bugs that make it into the wild by using more tests on our codebase. This seems broadly true, but incomplete.

First, it seems that only about half of our severe bugs appeared to be the kind of thing that better tests would have caught. The other half involved logic errors and design oversights that would probably have made it through testing.

Second, it seems that in some cases, our existing tests were adequate to the job, if only we had automated them better, or had run them more consistently, more rigorously, or under more conditions.

In all cases, of course, automation isn’t quite enough. We must also have the automated tests run regularly (daily?), and make sure that the results are available to developers in a convenient way.

**Recommendation 1.1 Run our automated unit tests under more code-hardening methodologies.**

This includes --enable-expensive-hardening under GCC and clang, valgrind with leak checking turned on, and anything else we can find.

Bugs where running tests under hardening or valgrind might have helped include: #13104,#14821, #17401, #17404, #18454.

---

⁴ [https://blog.torproject.org/blog/tor-0283-alpha-released](https://blog.torproject.org/blog/tor-0283-alpha-released)
⁵ [https://trac.torproject.org/projects/tor/ticket/18668](https://trac.torproject.org/projects/tor/ticket/18668)
⁶ [https://trac.torproject.org/projects/tor/ticket/19008](https://trac.torproject.org/projects/tor/ticket/19008)
**Recommendation 1.2:** Also run test-network and test-stem in an automated environment.

These checks can detect a lot of problems, but right now we only try the stem tests in automated builds, and we don't try them with hardening.

Cases where a suitably extended (or completely vanilla) stem or chutney test case might have helped include: #8746, #9296, #10465, #10849, #11200, #13698, #15245, #15801, #16247, #16248, #17668, #17702, #17772, #18116, #18318, and #18517.

**Recommendation 1.3:** Automate use of static analysis tools with Tor.

There were some cases where we found a bug using a static analysis tool later than we might have, because the static analysis tool had to be hand-launched. We can get faster bug resolution by automatically running all the static analysis tools we use. (We've already done this.) Static analyzers might have caught: #13477 and #18454, at very little effort on our part.

**Recommendation 1.4:** Continue requiring unit tests for new code, and writing unit tests for old code.

Untested code had bugs at a higher rate than tested code.

Bugs where plain old unit tests might have helped include: #11824, #12195, #13066, #13151, #15083, #16400, #17041, #17668, #17702, #17772, #17876, #18162, and #18318.

**Recommendation 1.5:** Get more users to try out our nightly builds.

Having more users of our nightly builds would help us notice more bugs on the git master branch before those bugs appear in stable or alpha releases.

Having users for our nightly builds would have prevented #11200 entirely.

**Recommendation 1.6:** Whenever possible, write integration tests for new features.

Features that lack integration tests via Chutney or some other mechanism tend to have bugs that last longer than other bugs before anybody notices them.

(See stem/chutney list above.)

**Recommendation 1.7:** We need more tests about shutting down busy clients and relays.

Our code tends to have a fair number of corner cases concerning shutting down at the wrong time, and crashing or asserting rather than exiting cleanly.

Careful shutdown tests might have caught #8746 and #18116.

---

**Building sustainability for the program**

As we mentioned on our last report, we want to work on ensuring the continuity of the best practices we have created with this program.

Since the beginning of the year the Network Team has applied a new code review process with first and second reviewers. This technique helps divide the workload of reviewing code and ensures that we have reviewers for all the code coming in. It is during code review that we can ensure that tests are applied to our code.

Not only that but also helps developers improve their skills, where one can graduate to become second reviewers of subsystems they master on. This will also help remove a bottleneck we have today where only the Tech Lead of the team is in the position of second reviewer for all the systems of core tor.
This concentration of work in one person can lead to slow down on code merging and doesn’t create a space for others to grow. We hope with our new process to have more subsystems maintainers that can become second reviewers of those subsystems and therefore remove this bottleneck.

Challenges

Management of the team capacity has become one of the challenges we are facing while adding these best practices of code review.

It is important that a developer has a good understanding of how much they can commit for the sprint (our sprints are one month long). Not all of their time will be spent on writing code -- they should also count their time reviewing other developers’ code, as well as other work such as writing technical design proposals.

We must also ensure that developers are not overcommitting themselves, which can lead to delays with deliverables and lack of time for code review.

We are adopting scrum techniques and incorporating points for tasks as well as for code review. This way a developer can plan accordingly within their capacity. The developer can better organize their time because they will have points that can tell how long it will take to either do a task or review someone else's code.

Enhanced outreach

Project objectives

We seek to make more people aware of the benefits of Tor, especially in the scenario that censorship circumvention needs to be combined with privacy and anonymity to help civil society members can work in Internet-repressive environments.

Outputs for the quarter

Direct work with organizations

Google Summer of Code (GSoC)

As mentioned last quarter, we joined GSoC and we are happy to say that we currently have 7 students working on projects related to Tor and being mentored by Tor people:

- Amogh Pradeep - Introduction
  - Project: Orfox
  - Mentor: Nathan / Hans

---

8 [https://blog.torproject.org/blog/gsoc-2016-projects](https://blog.torproject.org/blog/gsoc-2016-projects)
● Huy Vu - Introduction
  ○ Project: CONIKS for Tor Messenger
  ○ Mentor: Marcela / Arlo

● Ismael Riahi - Introduction
  ○ Project: Ahmia search engine for hidden services
  ○ Mentor: Juha Nurmi / George

● Mridul Malpotra - Introduction
  ○ Project: Exitmap improvements project
  ○ Mentor: Philipp / Damian

● Pierre Laperdrix - introduction
  ○ Project: The Torprinter project
  ○ Mentor: Georg / Günes / Nicolas

● Sambuddha Basu - introduction
  ○ Project: Expand Nyx
  ○ Mentor: Damian / Sebastian

● segfault - introduction
  ○ Project: Tails Server
  ○ Mentor: anonym / George

We also partnered with EFF and a broader set of organizations and companies to organize the 'Day of Action: Stop Changes to Rule 41'⁹:
Facebook

Facebook announced\(^\text{10}\) that 1 Million users are connecting to Facebook using Tor!

\(^{10}\)https://www.facebook.com/notes/facebook-over-tor/1-million-people-use-facebook-over-tor/865624066877648
Outreach materials

Simply Secure hired a designer who did a very similar project for Mozilla, his name is Elio Qoshi and he has a long experience working as designer and with the FOSS community.

We are super happy to have him onboard, as well as the continued help and guidance from Simply Secure with this project. We have already achieved a lot of progress, is a hard project because Tor’s original colors are hard to work with (purple and green), but our goal is not to do a rebranding but to build a guideline based on Tor’s current brand. This is just the beginning of a lot of great work, the team understands the challenge ahead and is very excited to work on it.

We are sharing a screenshot from Elio’s personal blog\(^\text{11}\) announcing that he joined our team which we believe it express the sentiment of everyone in the team as well:

---

\(^{11}\) [https://elioqoshi.me/en/2016/07/joining-tor/](https://elioqoshi.me/en/2016/07/joining-tor/)
Joining Tor as a Brand Designer

I’m deeply humbled and excited to start my work as a contracted Brand Designer at The Tor Project, joining a young and talented group of people who aim to improve Tor’s UX and visual identity across its communication channels. My first duty at Tor will be to unify a consistent style guide for Tor’s visual identity, so we can stop fragmenting the Tor brand across various places and create the first step for fellow designers to join the UX team. In the following some rough early drafts during research.

Thanks to Open Source Design and Simply Secure for suggesting this position to me. It’s a privilege to work on software which can affect history. Watch this space.

Blog post from Elio

Tactical Tech

Create a beautiful video\(^\text{12}\) to be used as outreach for more people to collaborate with the Tor Network. We published it on our Youtube channel and shared it over social media and other platforms.

\(^{12}\) https://www.youtube.com/watch?v=h114LK71GQo
Media strategy:

As part of our media education efforts we are starting to published Q&A with our developers on our blog to help reporters (as well as users) to better understand new features and technology we are working on.

The Q&A with our Tor Browser team lead explaining Selfrando\textsuperscript{13} is a great example of this effort:

\textsuperscript{13} https://blog.torproject.org/blog/selfrando-q-and-georg-koppen
Selfrando: Q and A with Georg Koppen

Georg Koppen is a longtime Tor browser developer. He and Tor developer Mike Perry worked to integrate Selfrando into Tor browser.

Tell us about Selfrando, the new code being tested for Tor Browser.

Selfrando randomizes Tor browser code to ensure that an attacker doesn’t know where the code is on your computer. This makes it much harder for someone to construct a reliable attack—and harder for them to use a flaw in your Tor Browser to de-anonymize you.

How were you and Tor’s Mike Perry involved in the project?

We mainly worked on integrating Selfrando in Tor Browser where needed and tested it as well as we could. We closely read the paper and helped to improve it. The bulk of the work was done by the other researchers. These are Mauro Conti, Stephen Crane, Tommaso Frassetto, Andrei Homescu, Per Larsen, Christopher Liebchen, and Ahmad-Reza Sadeghi.

Can you talk about Tor’s relationship with the research community?

You can see some of examples of great coverage we got from the media about Selfrando at our Press section in this report.

Social media

We tried something new this quarter: we initiated a rapid response for censorship events to alert users when they can use Tor to circumvent it.

For instance, during Uganda’s election internet services were censored in an attempt to impede free speech and debates, as our metrics site registered:
The blue and red dots are censorship events detected by our metrics.torproject.org website. OONI (Open Observatory of Network Interference) also published a more detailed report on how the Ugandan government actually blocked social media, you can read the full report here.

![Graph showing censorship events](https://metrics.torproject.org/)

During those days of censorship, we used our Twitter account to let people know how they could use Tor to bypass the censorship. We used the hashtag #NetFreedomUG which started to trend around the event, and “@mentioned” accounts of Ugandans activists and influencers so they could help spread the information:

---

14 [https://twitter.com/torproject/status/735774467234762753](https://twitter.com/torproject/status/735774467234762753)
16 [https://twitter.com/torproject/status/730514181468893184](https://twitter.com/torproject/status/730514181468893184)
17 [https://twitter.com/torproject/status/730526616573595649](https://twitter.com/torproject/status/730526616573595649)
We did the same work when Brazil blocked WhatsApp, you can review our blog post documenting the impact of it on Tor:\footnote{https://blog.torproject.org/blog/tracking-impact-whatsapp-blockage-tor}

\footnote{https://twitter.com/torproject/status/730514777240383489}

Use Orbot free (safe) VPN to reach social media in \#Uganda guardianproject.info/apps/orbot/ cc @Opimva @nickopiyo @asia_ilse
Challenges

We want to scale our rapid response outreach campaigns against censorship. For that we need to build a database of messages explaining how to use Tor, or how to get Tor in case our site is censored, or how to use a PT in case our network is censored and many other possible scenarios for our top 12 languages (EN, AR, FA, ES, RU, zh-CN, PT-BR, FR, DE, KO, TR, IT).

We must increase our reach by making sure we spread the word to more organizations involved with Internet Freedom efforts in order to generate the ‘snowball’ effect.

Develop a checklist of places to send our messages. And build a rapid response analyses team that can look into different data points to build the story of how the event happened and why. This means working with OONI, journalists, social scientists and etc.

It will be really hard to get all this done in a quarter but we want to build as much we can so this is a sustainable program that can scale and help other members of our community to fight against internet censorship.
### Press

<table>
<thead>
<tr>
<th>Date Published</th>
<th>Media Outlet</th>
<th>Article Title</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>06/28/2016</td>
<td>Ghakhs.net</td>
<td><strong>Selfrando improves Tor Browser security</strong></td>
<td></td>
</tr>
<tr>
<td>06/24/2016</td>
<td>The Merkel</td>
<td><strong>Selfrando Technique Makes Tor Nearly Invulnerable To De-anonymization Attacks</strong></td>
<td>The Merkle is cryptocurrency news site</td>
</tr>
<tr>
<td>06/23/2015</td>
<td>The Inquirer</td>
<td><strong>Mozilla throws $385k at open source projects including Tor and NVDA</strong></td>
<td></td>
</tr>
<tr>
<td>06/23/2016</td>
<td>Register</td>
<td><strong>What do Tor, Tails and Caddy have in common? Mozilla bucks</strong></td>
<td></td>
</tr>
<tr>
<td>06/22/2016</td>
<td>ZDNet</td>
<td><strong>Tor Project teams up with researchers to toughen up Tor browser</strong></td>
<td></td>
</tr>
<tr>
<td>06/22/2016</td>
<td>Tech Eye</td>
<td><strong>Human Rights groups furious about new US warrant law</strong></td>
<td></td>
</tr>
<tr>
<td>06/21/2016</td>
<td>The Daily Dot</td>
<td><strong>Internet-rights groups launch protest over expanded FBI hacking powers</strong></td>
<td></td>
</tr>
<tr>
<td>06/16/2016</td>
<td>BBC-Persia</td>
<td><strong>Tor developer discusses privacy tools with BBC-Persia</strong></td>
<td></td>
</tr>
<tr>
<td>06/02/2016</td>
<td>dna</td>
<td><strong>Tor Browser update adds improved HTML5 support and DuckDuckGo search engine</strong></td>
<td>dna is Indian news site</td>
</tr>
<tr>
<td>05/31/2016</td>
<td>Softpedia</td>
<td><strong>Tor Browser 6.0 Based on Firefox 45-ESR Released with Updated Security Features</strong></td>
<td></td>
</tr>
<tr>
<td>05/25/2016</td>
<td>Softpedia</td>
<td><strong>Tor to Use Never-Before-Seen Distributed RNG to Generate Truly Random Numbers</strong></td>
<td></td>
</tr>
<tr>
<td>05/19/2016</td>
<td>The Merkle</td>
<td><strong>Tor Technology Can Bring Much Needed Encryption To Healthcare</strong></td>
<td>The Merkle is cryptocurrency news site</td>
</tr>
<tr>
<td>05/12/2016</td>
<td>Committee to Protect Journalists</td>
<td><strong>How SecureDrop helps CPJ protect journalists</strong></td>
<td></td>
</tr>
</tbody>
</table>
Next quarter’s planned activities

Pluggable Transport integration

Our goals for the next quarter are to:

● Complete the development of Basket2 and its documentation.
● Organize discussions with the community about PT specification changes at Tor Dev Meeting in September.
● Continue to provide support for R&D related to censorship circumvention studies.
● Provide all the necessary support for PTs (bug fixes and improvements) on Core Tor and Tor Browser.
● Metrics of Pluggable Transports usage per country
Testing and network simulation improvements

- We will continue to work on team processes to ensure the good practices created during this program.
- We will continue to improve our testing infrastructure and fix any bug related to it.
- We will work on maintaining our testing coverage by implementing tests for the patches and features we are working on for Core Tor 0.2.9 release (next stable release planned for mid October)

Enhanced outreach

- Promote our outreach video explaining Bridges and Pluggable Transports for users.
- Document our rapid response program, identify areas of improvement and build a plan for scaling it.
- Continue our work with the media and organizations to help improve education about Tor and how our technology works.
- Publish an updated version of Tor Browser Manual
- Work to revive our support system - the first step will be to map all information that such a system should have to help our users. This will help us build the information architecture of this system.