Award identification number: S-LMAQM-17-GR-1121
Recipient Organization: Tor Project Inc.
DUNS Number: 809211100
EIN: 1208096820A1
Period covered by report: FYQ1 2018

Tor Project

Progress Report Q4 2017 - S-LMAQM-17-GR-1121

Isabela Bagueros
Project Manager
The Tor Project
isabela@torproject.org

Table of Contents

Project Information

Grantee:	The Tor Project, Inc.
Project Title:	Tor Browser for Android
Award Number	S-LMAQM-17-GR-1121
Period of performance:	Q4 of the year -October 1st 2017 - December 31st 2017
Reporting date:	January 31st, 2018
Reporting frequency:	Quarterly
Email contact:	isabela@torproject.org

Summary

Work related to Tor Browser for Android

Our newly formed mobile browser team start to work on our strategy to get a browser for Android. The team start working in December and reviewed the current code base of Orfox, met with Guardian Project developers to discuss best approaches. Also spent a great time reviewing previews work and also met with Mozilla developers to discuss their mobile strategy so we can make sure we plan accordingly.

As for what is coming up next, most of the implementation proposals are already under discussion in Q1 and currently we are discussing an alpha release of Orfox to start merging the work being done by the team. This follows our strategy of doing incremental work to achieve our final product. This way users has a chance to use some of the features as soon as they are ready.

Work related to Tor network optimization for mobile users

Our network team has managed to measure our performance on mobile devices, this way we can identify metrics to improve and measure our impact as well. For instance, this quarter we worked on CPU load on mobile devices and were able to identify where in the code should be changed and with those fixes decreased significantly the load.

Improvements on other metrics identified as important by mobile apps developers interviewed at the beginning of this project, such as storage space, memory consumption and battery consumption. We have improved the developers ability to control Tor's functions and manage better its performance. And continue to modify the code so it's easier to have Tor working as a library within your app.

All our efforts are towards making it easier for developers to adopt Tor by meeting the standards requested by them. For Q1 we continue to work on these goals and we are also hoping to have some face to face feedback from developers at our bi-annual meeting in March.

Objective 1: Build a Tor Browser for Android to empower mobile users to safely, anonymously, and securely interact with Internet resources and services.

Activity O1.1: Build a Tor Browser for Android with functionality and build processes in parity with desktop Tor Browser.

Progress Report:

Our first step was to look at tickets created in our issue tracker system that are related to building the Tor Browser bundle to Android. Our Browser team investigated it before and had lined up some ideas on how to better approach such project. Our new mobile team organized this work and start to address some of the lower hanging fruits to start bringing functionality and build processes that we have on our desktop browser to our Android one.

Ticket #5709[1]: "Tor Browser Bundle build for Android" was created by Roger Dingledine and Mike Perry based on investigation done a few years ago that line up an approach to use Proxy

[1] https://trac.torproject.org/projects/tor/ticket/5709

Mobile as the proxy forcer addon. And with that follow a series of incremental steps to add Torbutton features into the mobile build. Some of this work has already been done years ago.

One of the subtasks here is to merge Orfox patches into Tor Browser[2], this mean to have one single tree of code for desktop and Android and therefore make it easier to keep track of all Tor Browser related things. This work involves many steps, but the first one was to bring Orfox (using version 38) into ESR parity with Tor Browser desktop (using version 52)[3].

Risks Assessments:
So far for this part of the work is just time. We know we started late in our timeline, but we also see progress happening for this activity. So everything is looking quite positive despite this bump at the beginning.

Next Report [Q1 2018]:
One of the things coming up on Q1 2018 is a proposal[4][5] for porting Tor Button extension to mobile. The work related to this proposal has also being organized on our issue tracker and can be followed under our master ticket #24855[6].

Activity 01.2: Research and develop Android specific fingerpriting defenses for Tor Browser.

Progress Report:

For this beginning of the project we are auditing the android-specific code for proxy-bypass and fingerprintability. We are also comparing the work done on Tor Browser desktop to build defenses for those to see if can be used for Android. This will give us an idea of how much work will need to be redone for Android specifically needs.

Risks Assessments:

One of our concerns is amount of work that will need to be re-done in comparison to what we can actually use from the desktop protections.

Next Report [Q1 2018]:

Continue our investigation, including talking with other mobile developers who have worked on this area during our bi-annual meeting in Rome in March.

[2] https://trac.torproject.org/projects/tor/ticket/19675
[3] https://trac.torproject.org/projects/tor/ticket/23144
[4] https://storm.torproject.org/shared/fchH0_Liol-cWFQwH1dgQsETUC7whKb7Hy46cPYZLrS
[5] https://lists.torproject.org/pipermail/tbb-dev/2018-January/000723.html
[6] https://trac.torproject.org/projects/tor/ticket/24855

Activity 01.3: Work with Mozilla to merge and built defenses back into Firefox Mobile

Progress Report:

In early December we joined Mozilla All Hands in Texas and there we met with the following teams:

- Fusion project - responsible for most uplifting to Firefox desktop browser.
- Tor proxy support - a small team investigating how to make it easier for Firefox to use Tor as a proxy.
- Focus team - mobile lightweight client from Mozilla with high standards for privacy.

Besides spending a week with Mozilla folks organizing projects we have in common, we also have monthly meetings for check-ins and also plan on having Mozilla folks join us at our bi-annual meeting in Rome, in March.

Risks Assessments:

Mozilla has different browser dev kit they are using. Which means at any moment they could pick one over the other and our contributions get lost. Part of our discussions during Mozilla All Hands were about this possibility and to decide which browser dev structure to build ours. We ended up deciding to stay with the one used by Orfox and which we can upstream to mobile ESR

Next Report [Q1 2018]:

Continue working with Mozilla to merge and built defenses back into Firefox mobile.

Objective 2: Specifically Build a Tor Browser for Android to empower mobile users to safely, anonymously, and securely interact with Internet resources and services.

Activity O2.1: Enable standalone use of Tor Browser for Android without separate Orbot installation

Progress Report:

We spend some time investigating the code to draft a proposal for porting Tor Launcher, we also created a main ticket to track this project, #24856[7].

[7] https://trac.torproject.org/projects/tor/ticket/24856

Risks Assessments:

If we want Tor Browser to be usable, then tor integration must be seamless, and the complexity must be hidden from the user. For that we need the design to be such that it balances the isolation/usability/complexity differential of the implementation.

Next Report [Q1 2018]:

In Q1 2018 we have been discussing a proposal that was submitted to our development team email list[8], a second version[9] has already been shared based on the first round of feedback. We hope to finalize this and the implementation plan by our face to face meeting in Rome, in March.

Activity 02.2: Improve usability of Tor Browser for Android, relative to Orfox, including anti-censorship bridges.

Progress Report:

We haven't started this work yet. This part comes after the Tor Launcher porting work.

Risks Assessments:

n/a

Next Report [Q1 2018]:

n/a

Activity 02.3: Improve software and Tor Network architecture to improve usability for low-speed networks and low-power, low-RAM devices.

Progress Report:

The goal of this activity is to reduce resources and performance bottlenecks so we can optimize Tor for mobile clients. This quarter we start to measure Android performance metrics we want to improve on Tor to make it meet the standards from third party app developers.

The areas we focused on this quarter are based on the reports from Android developers to what are the key metrics they care about. We still have other areas to approach, which we will do during the curse of this project.

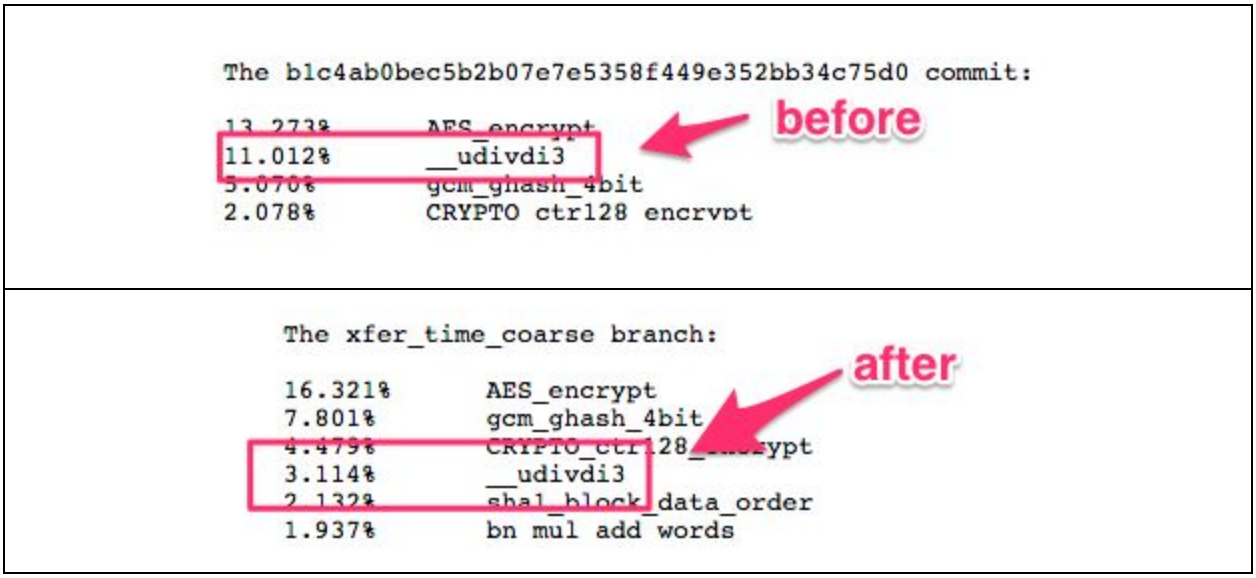
CPU load

[8] https://lists.torproject.org/pipermail/tbb-dev/2018-January/000735.html
[9] https://lists.torproject.org/pipermail/tbb-dev/2018-January/000743.html

Based on the investigation we did in Q3 on how to use the Android development environment to collect measurements. We chose to use simpleperf[10][11], a command line tool to get CPU profiling information from Tor via the Orbot application.

This tool will help us collect CPU measurements in a reproducible manner on a set of Android devices, this way we can build a baseline and then measure if our improvements are having any effect or not.

First we worked on CPU profiling for Android[12] and published our initial reports here[13]. The preliminary profiling identified emulated 64-bit division on 32-bit platforms as a bottleneck for some mobile devices, leading us to refactor some of our timing logic code (24374[14], 24613[15]) to fix that and take some load off from the CPU. We reduced it from 11.012% to 3.114%[16]:



Storage reduction

For storage usage reduction, we've divided Tor's cached documents, sensitive keys, and other state files into separate configuration options[17][18][19], so that mobile embedders can tell the mobile OS which of them can be deleted when under pressure.

[10] https://trac.torproject.org/projects/tor/ticket/24062
[11] https://gitweb.torproject.org/tor.git/tree/doc/HACKING/android/Simpleperf.md
[12] https://trac.torproject.org/projects/tor/ticket/24061
[13] https://people.torproject.org/~ahf/sponsor8/20171116/
[14] https://trac.torproject.org/projects/tor/ticket/24374
[15] https://trac.torproject.org/projects/tor/ticket/24613
[16] https://anubis.0x90.dk/~ahf/udivdi3.txt
[17] https://trac.torproject.org/projects/tor/ticket/22703
[18] https://trac.torproject.org/projects/tor/ticket/24268
[19] https://trac.torproject.org/projects/tor/ticket/24272

Tor network wakeups

These network wakeups will consume device memory, so we want to gain a better understanding on why we are waking up and to be able to compare our progress with waking up less.

For that we have written support for instrumenting the Tor main event loop in #24605[20] which allows us to baseline progress and catch regressions.

Tests with Shadow

While simpleperf allows us to perform tests and measurements on device, we still need to test our code changes on Tor's behavior in relationship to the Tor network in general, and we will use Shadow to do that.

During our Tor Meeting in Montreal in November we met with Shadow developers from NRL to set experiments using it. This allows us to test things while simulating time, which will speed things up with experiments, for instance instead of running an experiment for a week we could simulate a week of having that code running in the network and collect the results right away.

Risks Assessments:

For this stage of the work we must be very diligent with our measurements so we are not mislead by wrong data.

Another risk we are trying to mediate with our experiments on Shadow are the impact of our changes in the code when it's running as part of our network. Tor configuration file allows the same code base to function as a client or a node in the network (between other functionalities).

Our work here aims to improve the client performance on a mobile device, but while doing that we must make sure that it is not hurting the performance of running it on a relay part of the network.

Next Report [Q1 2018]:

Our work on Q4 reduced the heavy amount of calls to the compiler-generated udivdi3 function, for Q1 2018 we plan to to continue improving on this, and for that we are currently investigating the remaining usage here. One of our suspecting offenders is the timer wheel code (24688[21]).

[20] https://trac.torproject.org/projects/tor/ticket/24605
[21] https://trac.torproject.org/projects/tor/ticket/24688

Additionally we identified that we should look into:

- Are we are doing unneeded SHA-3 operations.
- Are we are doing any unneeded RSA / DH operations.

And hope it will help us continue to optimize Tor for mobile adoption.

Activity 02.4: Improve the Tor Network's controller interface to allow mobile apps to reduce bandwidth and battery use.

Progress Report:

During Q4 we worked on a proposal for controller interface to allow developers to better manage battery consumption of Tor in mobile devices:

Proposal 286: "Controller APIs for hibernation access on mobile"[22][23]

This proposal describes controller APIs for better management of Tor's hibernation mechanisms, and extensions to those mechanisms, for better power management in mobile environments.

We are proposing to introduce new hibernation states: "IDLE", "IDLE_UPDATING", "SLEEP", and "SLEEP_UPDATING". To give more options to developers, but also including certain rules to make sure the network can't be impacted by it, such as not allowing relays and bridges to automatically become IDLE on their own.

The proposal is under discussion on our development email list[24] and a final review meeting is scheduled for Q1 so it can be finalized and we can move on with its implementation.

Make it easier for mobile app developers to embed tor

We made great progress on a crucial requirement that came from our mobile developers community, which is the ability to run Tor inside another process rather than as a process of its own, or as we call it, to embed Tor.

Notable achievements so far include:

- Designing and publishing a stable API for tor embedding[25].
  - For the usage in the mobile context it's useful to have a documented tor main function to call to start it and exposed as C headers
- Making Tor not shut down the process when it exits[26].

[22] https://gitweb.torproject.org/torspec.git/tree/proposals/286-hibernation-api.txt
[23] https://trac.torproject.org/projects/tor/ticket/24107
[24] https://lists.torproject.org/pipermail/tor-dev/2017-November/012634.html
[25] https://trac.torproject.org/projects/tor/ticket/23845
[26] https://trac.torproject.org/projects/tor/ticket/23848

    - Currently tor uses inside of various places the exit() call, but this is not nice when you are using it as a library as it leads to the whole app crashing, while we would rather just get an exception or a proper error code.
- Let programs call tor_main with a preconstructed control socket[27].
    - Have a way for programs that want to call tor_main to pass a control socket to tor_main, so that they don't need to have tor listening on a control port at all.
- Make signal handlers optional, for starting Tor in-process[28].
    - When Tor starts, it installs handlers for a bunch of signals. But if you're running Tor in its own thread, there's a good chance you don't actually want that behavior.

Another piece we need to accomplish in order to make it easier for apps to have Tor built-in is the ability to make sure Tor can shut down via control port, and start again in same process.

The tasks related to this work finished this quarter were:

- Don't crash when restarting Tor in the same process[2930] - this was the first thing we had to address
- Make it easy to debug restart-in-process[31] - developer only feature designed to help us catch bugs while implementing this work.
- Fix memory-leaked event_base_once() users.[32] - The event_base_once() function is prone to leak memory on event loop exit. This becomes a bigger problem once this work is in place.

Risks Assessments:

Right now we need to make sure that all optimization we are doing in the code does not break other parts of it and most important does not creates security issues for the network.

All the testing process we created in the past couple of years are very helpful for us to be able to catch bugs or security issues before any release. Mitigating this risk.

Next Report [Q1 2018]:

Controller APIs for better battery management our next directions are to improve the Proposal 286 and examine ways in which hibernation logic needs to improve to use less battery while running Tor.

[27] https://trac.torproject.org/projects/tor/ticket/23900
[28] https://trac.torproject.org/projects/tor/ticket/24588
[29] https://trac.torproject.org/projects/tor/ticket/24581
[30] https://trac.torproject.org/projects/tor/ticket/24337
[31] https://trac.torproject.org/projects/tor/ticket/24583
[32] https://trac.torproject.org/projects/tor/ticket/24584

Will continue to work on embedding Tor tasks documented at our main ticket[33] for this project and the tasks related to restarting Tor as part of the same process[34].

Activity 02.5: Enable better reporting of network and connection errors to apps that use Tor Network

Progress Report:

Past quarter we worked on bootstrap errors. The bootstrap part is very important because the messages about it are related to what Tor is doing to connect to the rest of the network. The better the reporting here the better will be for the user experience. Because we will be able to better deal with troubleshooting the problem and figuring out what is the solution. With that information we can either solve it ourselves, making it seamless for the user or give precise instructions to them to know how to proceed.

Many of these bootstrap errors we investigated over this quarter are often seen when a client's clock (user computer clock) is several hours off from UTC.

We have seen many users come to IRC and to our support forums reporting clock skew results from incorrect time zone settings (usually a time zone set to UTC when that's not the local time zone but it displays correctly so the user doesn't think it's wrong).[353637383940414243]

Risks Assessments:

So far at Tor, we haven't really analyzed or documented security considerations very well at all. So we must dedicate time to do so, to ensure that changes in this area will not create a security problem.

Next Report [Q1 2018]:

For Q1 2018, we will study the security impact of changes in this area, especially about making enforcement of consensus expiration more lenient.

---

[33] https://trac.torproject.org/projects/tor/ticket/23684
[34] https://trac.torproject.org/projects/tor/ticket/23847
[35] https://trac.torproject.org/projects/tor/ticket/2878
[36] https://trac.torproject.org/projects/tor/ticket/23508
[37] https://trac.torproject.org/projects/tor/ticket/23605
[38] https://trac.torproject.org/projects/tor/ticket/23565
[39] https://trac.torproject.org/projects/tor/ticket/24300
[40] https://trac.torproject.org/projects/tor/ticket/24367
[41] https://trac.torproject.org/projects/tor/ticket/24392
[42] https://trac.torproject.org/projects/tor/ticket/24486
[43] https://trac.torproject.org/projects/tor/ticket/24661