

Introduction

The Tor Project started work on our awarded program, "Tor Measurement and Metrics Improvement," on July 1st, 2016.

This summary lists work accomplished to date by Tor Metrics Team. The project roadmap, linked at the end of this document, describes planned work through June 2017 (subject to change).

Our first step in this project was to promote the MOSS program and publicize our project with a blog post: <https://blog.torproject.org/blog/tors-innovative-metrics-program-receives-award-mozilla>

Milestone 1: Complete

Milestone: 1. CollecTor

1.1. Define a release process for CollecTor and put out one or more releases.

Complete. We created a process for it and you can see our releases here:
<https://dist.torproject.org/collector/>

1.2. Provide user-friendly documentation that empower users to independently operate CollecTor instances.

Complete. Our latest release contains an improved operator's guide:
<https://gitweb.torproject.org/collector.git/tree/INSTALL.md?id=collector-1.1.0>

1.3. Enable CollecTor to synchronize Tor network data from other CollecTor instances.

Complete. This is the major change in the latest 1.1.0 release:
<https://gitweb.torproject.org/collector.git/tree/CHANGELOG.md?id=collector-1.1.0>

1.4. Set up a second CollecTor instance and enable synchronization with the first CollecTor instance.

Complete. We are running such a CollecTor instance that synchronizes descriptors from the Tor's primary CollecTor instance.

Progress toward other milestones:

Milestone 2: metrics-lib/DescripTor

2.2 Put out one or more metrics-lib/DescripTor releases.

Released metrics-lib/DescripTor version 1.3.0 [1] to add parsing support for various new fields in Tor descriptors and to make some minor bugfixes.

[1] <https://lists.torproject.org/pipermail/tor-dev/2016-July/011201.html>

Released metrics-lib/DescripTor version 1.3.1 [2] with a hot fix to account for a change in CollecTor's directory listings and metrics-lib/DescripTor version 1.4.0 [3] with new approaches for logging and for fetching descriptors from the CollecTor service.

[2] <https://lists.torproject.org/pipermail/tor-dev/2016-August/011250.html>

[3] <https://lists.torproject.org/pipermail/tor-dev/2016-August/011334.html>

Released metrics-lib/DescripTor version 1.5.0 [4] that makes the DescriptorCollector implementation that uses CollecTor's index.json file to determine which descriptor files to fetch the new default.

[4] <https://lists.torproject.org/pipermail/tor-dev/2016-October/011584.html>

Milestone 3: Onionoo and Atlas

3.1. Define a release process for Onionoo and put out one or more releases.

Created release process for Onionoo and released the first release following it. [1]

Released Onionoo 3.1.-1.0.0 The version of Onionoo [2] consists of the current protocol version, which is 3.1 and the software's version 1.0.0

(The first Onionoo protocol was already released five years ago.)

[1] <https://lists.torproject.org/pipermail/tor-dev/2016-November/011665.html>

[2] <https://dist.torproject.org/onionoo/3.1-1.0.0/>

3.2. Provide user-friendly documentation that empowers users to independently operate Onionoo instances.

Published our 'Onionoo Operator's Guide' [3] as part of the Onionoo 3.1-1.0.0 release.

[3] <https://gitweb.torproject.org/onionoo.git/tree/INSTALL.md>

Milestone: 6 Metrics

6.1 Conduct a usability analysis of the current Tor Metrics website to identify the most pressing usability issues.

We are working with the Tor Project's UX team on this task and the project is been documented here:

<https://trac.torproject.org/projects/tor/wiki/doc/UX/MetricsRedesign>

We are currently finalizing wireframes. Prior to that we discussed different use cases based on personas we identified as possible audience for the site. Once we have wireframes, we will pass these resources to our designer. They will create mockups for our new site design. We hope these will address the main points identified at our usability analysis.

We worked on some backend improvements to our metrics website, we made user graph updates on Metrics [2] faster by roughly 3 seconds by switching the underlying data format from comma separated values files to the RData format.

[2] <https://metrics.torproject.org/?tag=cl&type=qr>

6.3. Add a visualization of Tor Browser downloads to Metrics.

Created a database schema and importer[1] for sanitized Tor webserver logs [2].
Wrote two prototypes using RStudio's Shiny[3] to explore Tor website hits and Tor
Browser/Messenger downloads and updates. [4, 5]

[1] <https://gitweb.torproject.org/metrics-tasks.git/tree/task-20008>

[2] <https://webstats.torproject.org/>

[3] <http://shiny.rstudio.com/>

[4] <https://tor-metrics.shinyapps.io/webstats/>

[5] <https://tor-metrics.shinyapps.io/webstats2/>

6.4. Add at least one visualization to Metrics that is requested by the community.

Added a new graph to Tor Metrics that shows a possible range of the number of
clients by country and transport and which reveals the most popular pluggable
transports in any given country [1] (#19544).

[1] <https://metrics.torproject.org/userstats-bridge-combined.html?country=us>

Additional Work

On our contract we also committed on performing a series of good practices for all our objectives
in order to make it easier for other organizations to run Tor network-observation software or
contribute to our code base. One of these practices is:

A. Increase software stability, scalability, security, and maintainability by improving code
quality, testability, and testing practices.

For this we did:

Added tests to CollecTor's bridge descriptor sanitizer [1], raising its line/branch
coverage from 0%/0% to 87%/73%.

[1] <https://trac.torproject.org/projects/tor/ticket/19755>

Next steps

Our roadmap for the rest of 2016 and through July 2017 (subject to change) is available at:

<https://trac.torproject.org/projects/tor/wiki/org/teams/MetricsTeam#RoadmapfromOctober2016toJune2017>