Tor Metrics Roadmap

karsten@torproject.org

January 17, 2012

Metrics products comprise all software that is used to collect and process data about the Tor network and its usage. The following table lists the metrics products with the number of source and test commands and the third-last commit date as rough estimates of complexity, test coverage, and development activity. Commands are only counted in Java source files as the number of semicolons outside of comments.

Number	Product name	Source commands	Test commands	Third-last commit
1	Metrics descriptor library	1506	930	Jan 16, 2012
2	Metrics website	3724	7	Jan 13, 2012
3	Metrics data processor	1541	32	Jan 14, 2012
4	Torperf	–	–	Jul 28, 2011
5	TorStatus	670	0	Jan 15, 2012
6	Webstats	183	0	Dec 30, 2011
7	DocTor	472	0	Jan 10, 2012
8	Metrics utilities	309	0	Jan 4, 2011

This document only describes Metrics products that are services provided by the Tor Project. It gives estimates of the effort to advance the products and what developer type is required. This document does not, however, cover the software that is required to run custom analyses on the metrics data.

1 Metrics descriptor library

The metrics descriptor library is one the most recent metrics products. It reads metrics-related descriptors from disk or downloads them from the Tor directories and provides their content to other metrics products in a convenient manner. The metrics descriptor library is becoming a major building block of most other metrics products, which all implement their own reading from disk, downloading from directories, and descriptor parsing. Having a single library avoids code duplication and facilitates testing.

As of today, about 50 % of the targeted functionality is implemented, about half of which is covered by unit tests. The library is used by TorStatus and DocTor and is designed to be used by the Metrics data processor and the Metrics website in the future. Implementing the remaining functionality and unit tests is high priority, because it helps reduce complexity of other metrics products and makes them more robust. **(Java developer, 2 months, high priority)**

2 Metrics website

The metrics website makes aggregate statistics about the Tor network available in customizable graphs and tables. The metrics website further provides (mostly static) information for researchers who are interested in researching the Tor network. The metrics website also contains a couple of services which should really be distinct services.

The currently deployed Metrics website is difficult to maintain and eats a lot of developer time. It should be a high priority to split the Metrics website software into the following products:

1. Metrics statistics aggregator: The current way of providing aggregate statistics about the Tor network is to import descriptors into a database and aggregate them there. The result is stored in tables that can be quickly queried to display a graph, table, CSV output, etc. This functionality should be implemented in a single software product, and the aggregate statistics should be provided to other products. The Metrics statistics aggregator should make use of the Metrics descriptor library. Most of the code can be re-used from the current Metrics website code. Separating the data processing and graph generation helps others visualize metrics data using third-party tools. **(Java developer, 1 month, high priority)**

2. Metrics website: The "Graphs" part of the Metrics website should be provided by a second software that obtains its data from the Metrics statistics aggregator. Most of the code can be re-used from the current Metrics website code. The

only change is that the database connection will be replaced with the interface that the Metrics statistics aggregator provides. Alternatively, a new website can be developed using the data that the Metrics statistics aggregator provides, which requires quite some more developer time, though. **(Web developer, 1 month, high priority)**

3. ExoneraTor and Relay Search website: The "ExoneraTor" and "Relay Search" parts of the current Metrics website should be implemented in a separate software product. In particular, they shouldn't use the same database that the Metrics statistics aggregator uses, which causes problems like tables with 67M rows that need to be partitioned using somewhat experimental PostgreSQL features. ExoneraTor and Relay Search can be implemented as one website with a single database. That website could also offer any other type of archived descriptor, which can be useful for debugging by Tor developers. The database importer should make use of the Metrics descriptor library. **(Web/Database developer, 1–2 months, high priority)**

3 Metrics data processor

The metrics data processor collects data and makes it available to other metrics services and to other researchers and developers. The metrics data processor also sanitizes bridge descriptors and bridge pool assignments. metrics-db currently has no public user interface. Its collected and sanitized descriptors are made available via HTTP (linked from the metrics website) and via rsync.

The currently deployed Metrics data processor is reasonably stable, but somewhat difficult to maintain and extend. In the future, the Metrics data processor should make use of the Metrics descriptor library, which is going to reduce its code complexity to about 40–60 %. The result will be better maintainability. However, the migration will be time-consuming. **(Java developer, 3 months, medium priority)**

4 Torperf

Torperf measures Tor's performance as experienced by clients. It establishes a circuit, downloads a file from itself via Tor, and notes timestamps of a few substeps.

Torperf is rather inconvenient to install and maintain by less technical users; which may be a problem. Setting up multiple Torperf instances, e.g., for experiments, is highly inconvenient. Torperf development is not taking place right now. Once that changes, the focus should be to improve its usability, both for non-technical users and for performance experiments. **(Python developer, 2 months, medium priority)**

5 TorStatus

TorStatus provides current relay and bridge information to be displayed by other websites or clients. TorStatus uses the Metrics descriptor library. The first version of the TorStatus code is almost feature-complete. **(Java developer, 1 month, medium priority)**

6 Webstats

Webstats is the software that collects and sanitizes Apache web server logs. The current Webstats code is deployed and feature-complete. It is expected to be rather low maintenance.

7 DocTor

The consensus-health checker, a.k.a. DocTor, downloads the current consensuses and votes from the directory authorities and looks for potential problems with consensus generation. The consensus-health checker has become an important tool to keep the Tor network working. It uses the Metrics descriptor library and is more or less feature-complete.

8 Metrics utilities

There are currently two tools in the Metrics utilities repository. The Java and Python versions of ExoneraTor enable technical users to look up relays in the Tor network without giving away the IP addresses they are looking for. VisiTor is a small tool for web server operators to compare web server logs to Tor's exit lists and learn what fraction of their users are Tor users.

Development on Metrics utilities is basically not taking place. There are hardly any users of these tools, and the few users never come talk to us to ask for new features or report bugs.