In October 2019, the OONI team completed its 1-year contract with the OTF (which started on 1st November 2018).

We would like to thank the OTF for its generous support, which has helped the OONI-verse grow and allowed us to build new and exciting tools for the internet freedom community!

Stable release of OONI Probe Desktop App for Windows and macOS

We released the stable version of the OONI Probe desktop app for Windows and macOS: https://github.com/ooni/probe-desktop/releases/tag/v3.0.0-rc.3

From version 3.0.0-rc.2 onwards, auto-update support is enabled and working on macOS and Windows. Users of the older beta versions may have to do a hard reset of OONI Probe from the About screen page.

The new OONI Probe desktop app supports running the Websites, Instant Messaging, Performance, and Middlebox tests, as well as configuring the maximum number of tested URLs for the Websites test (which defaults to the full test list).

We aim to publicly launch the new OONI Probe desktop app over the next weeks, once we have received more feedback from community members and have done more extensive testing of the app on more platforms.

Stable release of OONI Probe Command Line Interface (CLI)
We released the stable version of the OONI Probe Command Line Interface (CLI):
https://github.com/ooni/probe-cli/releases/tag/v3.0.0-rc.3

Prior to this, we also made other releases during October 2019:
https://github.com/ooni/probe-cli/releases/

Stable release of OONI Probe engine

To enable the integration of tests and to otherwise improve OONI Probe sustainability, we have been re-writing the OONI Probe measurement engine (the software that implements OONI Probe nettests) from C++ to golang.

The new OONI Probe measurement engine is available via the following repository:
https://github.com/ooni/probe-engine

During October 2019, we made several releases: https://github.com/ooni/probe-engine/releases

The latest release of the OONI Probe engine is stable: https://github.com/ooni/probe-engine/releases/tag/v0.2.0

Release of OONI Probe Mobile 2.2.0

We released the 2.2.0 version of the OONI Probe mobile app.

Android: https://github.com/ooni/probe-android/releases/tag/v2.2.0

iOS: https://github.com/ooni/probe-ios/releases/tag/v2.2.0

The latest 2.2.0 version:

- Uses less storage by automatically deleting old measurements and fetching them from OONI Explorer
- Enables users to copy and share OONI Explorer links based on their testing results

Seamless integration between mobile and desktop apps

To have a more seamless integration between the OONI Probe mobile and desktop apps, we integrated react-native (https://facebook.github.io/react-native/) into the probe-android and probe-ios apps.
This allows us to maximize code sharing between the OONI Probe desktop and the mobile apps, as well as making it possible for us to use our in-house React expertise for OONI Probe Mobile development as well -- hence reducing the ongoing maintenance effort.

We used the About screen as an example to understand the possible limitations to this approach and implemented it as a react-native screen: https://github.com/ooni/probe-react-native/blob/master/views/About.js

Some research was done into converting all of the ooni-components part of our design system (https://github.com/ooni/design-system) to react-native and we have understood that we will have to make some changes to how we use the styled-components module. Specifically, we have understood that we will have to refactor our code to use the react-native specific styled-components wrapped when importing modules into the react-native views.

We also spent some time researching the performance impact and application size overhead associated with using react-native. See: https://github.com/ooni/probe/issues/868#issuecomment-547720048

Based on all of this research, we have decided to postpone shipping the react-native integration until we have more views implemented and we have assessed the impact that the increased app size will have on our user-base.

We do still think this is something we should be doing, as it opens up opportunities for us to start sharing code between desktop and mobile apps for specific components, such as charts and tables. This will also help ensure that the UX and design is more similar and familiar across platforms.

Relevant code:
https://github.com/ooni/probe-react-native
https://github.com/ooni/probe-ios/pull/313
https://github.com/ooni/probe-android/pull/266
https://github.com/ooni/probe/issues/868

Made OONI Probe testing more resilient to network outages

To make OONI Probe testing more resilient to network outages, we:

- used jafar to stress test netx and implemented retransmission of DNS to improve DNS robustness
- made the “nervous resolver” fallback to DoH using Cloudflare, which is also a way to increase robustness
The automatic fallback to DNS over HTTPS / DNS over TLS has been designed in such a way that it will be triggered (without touching the scoreboard) also in the event that the master DNS fails. This will be useful to improve reliability when the master DNS used by the device is either down or overloaded. We experienced this kind of transient outage both in Cuba and in Italy, during a six-hour incident where Vodafone Italy DNS infrastructure was down.

We also ensured that the new Go code has timeout settings in line with the Go standard library and browsers like Firefox, thus further increasing OONI Probe reliability.

A detailed explanation of what this work entailed (including relevant code repositories) is available here: https://github.com/ooni/probe-engine/issues/88#issuecomment-548895700

**Implemented new OONI Probe nettests**

To improve and expand upon OONI Probe measurement methodologies, we implemented the following new nettests:

- Update to the latest ndt7: https://github.com/ooni/probe-engine/issues/70
- DASH rewrite that uses HTTPS: https://github.com/ooni/probe-engine/issues/51
- Telegram re-write using github.com/ooni/netx: https://github.com/ooni/probe-engine/issues/54

We have also integrated automatic follow-up measurements for:

- DNS over TLS / DNS over HTTPS fallbacks: https://github.com/ooni/netx/pull/108
- SNI blocking detection: https://github.com/ooni/netx/pull/109

The DNS over TLS / DNS over HTTPS fallback is performed anytime we resolve a domain name and we get a bogon IP (i.e. an IP that should not be public, such as, for example 10.0.0.1 or 127.0.0.1). In such cases, we record into a scoreboard structure that we were served a bogon IP and fallback to DNS over TLS or DNS over HTTPS.

SNI blocking detection is performed after a URL measurement; whenever we see a TLS handshake failure that leads us to think that there may be SNI injection, we run an additional follow-up test with a test helper (example.com:443 in the current experimental code) with the SNI we suspect being blocked and we also record the result into the scoreboard.

This design for follow-up experiments is also based on conversations with Vinicius Fortuna (Jigsaw) regarding how to improve the OONI testing methodology.
This work intersects with the "Improve basic on device data analysis" work. We chose a design where these new nettests are follow-ups, because this enables us to collect richer measurements without the need to prompt users to manually run follow-ups.

The above work was tracked through this master ticket:
https://github.com/ooni/probe-engine/issues/86

We implemented these new nettests in probe-engine and we plan to ship them as part of the OONI Probe apps over the next months.

**Ingested new metadata produced by new set of tests**

Previously, the OONI data processing pipeline only ingested metadata from OONI's Web Connectivity test.

We have now added support for ingesting and analyzing the metadata produced by most OONI Probe tests:

- Facebook Messenger test
- Telegram test
- Whatsapp test
- HTTP header field manipulation test (for middlebox detection)
- Vanilla Tor connectivity test
- TCP connect test
- DASH test
- Meek test

We have also implemented basic handling (without scoring) of the following OONI Probe tests:

- Psiphon test
- NDT test

This work was implemented as part of the fast-path pipeline in the following pull request: https://github.com/ooni/pipeline/pull/242

**Improved basic on-device data analysis**

To improve basic on-device data analysis (as mentioned above in "Implementing new OONI probe nettests") we added support for the following:

- OONI Probe can detect bogus DNS replies and fallback to DNS over HTTPS / DNS over TLS
- OONI Probe flags the event where the TLS handshake failed with connection reset and runs follow-up tests to determine whether it's a case of SNI-based filtering.

This work was tracked through the following ticket: https://github.com/ooni/probe-engine/issues/87

We also built Jafar (https://github.com/ooni/jafar) -- a tool that simulates a censored environment -- to ensure that our methods work as expected.

**Reduced the latency of publication of uploaded measurements to less than daily**

To analyze and publish OONI measurements from around the world faster, we built a fast-path pipeline: https://github.com/ooni/pipeline/pull/213

To reduce the latency of publication of uploaded measurements to less than daily, we implemented fast-path support in the OONI API. This work is available through the following pull request: https://github.com/ooni/api/pull/115

The database tables for the fast-path (which publish measurements daily) are already getting populated and published.

The OONI API related changes (which will expose the most recent measurements in OONI Explorer) are going to be shipped in production sometime next week, as we are trying to avoid deploying too many services at once.

**Added data processing pipeline heuristics for flagging relevant blocking events for the RSS feed**

To flag relevant blocking events and expose them via the RSS feed, we added data processing pipeline heuristics to the fast-path pipeline and ingested the metadata of most OONI Probe tests.

Our work on adding an event detector and RSS feed generation to the fast-path pipeline is available via the following pull request: https://github.com/ooni/pipeline/pull/228

Based on this, we have an internal web app that generates charts based on blocking events. The event detector is currently running.
Exposed blocking events in RSS feed hosted on OONI Explorer

Based on the event detector that we added to the fast-path pipeline (https://github.com/ooni/pipeline/pull/228), we created an RSS feed for exposing blocking events on OONI Explorer.

OONI Explorer now provides links to RSS feeds on events detected by the fast-path pipeline event detector. The following feeds are to be made available on OONI Explorer:

- Global feed (added to all pages)
- Country-specific feed (added to relevant country pages)

So far, we have deployed the global RSS feed which shows events on a global scale and is available here: https://explorer.ooni.org/rss/global.xml

Soon thereafter, we are going to be adding the country-specific feed on the OONI Explorer country pages.

This work was implemented through the following pull request: https://github.com/ooni/explorer/pull/342

For the time being, we are not going to be promoting the RSS feed feature of OONI Explorer extensively as we need to use it more ourselves and get feedback from trusted community members to avoid mis-interpretation of the data.

Ensured all our infrastructure is redeployable

We worked on ensuring that all our infrastructure is re-deployable, making it possible for us to recover from a disaster (a machine being terminated) with minimal amount of downtime and be able to migrate to up-to-date versions of Operating Systems to keep our infrastructure secure.

All of our work on ensuring that our infrastructure is re-deployable was completed in October 2019 and tracked via this ticket: https://github.com/ooni/sysadmin/issues/356

As part of this work, we also coordinated with Greenhost to ensure the redeployed hosts are using the eclips.is platform in the most resource-conscious way possible. For example, several unnecessary hosts have been terminated and some that did not require to be located in Hong Kong or Amsterdam have been migrated to the Miami location which has more availability of resources.

Improved the availability of our infrastructure
We improved the availability of our infrastructure via the "oonification of probe services".

The oonified bouncer, collector and orchestra backend are deployed and serving clients. The pipeline is using the new collector and we rolled-over all the collectors to the new probe-services oonified host.

Our work on oonifying the probe services is available via this pull request: https://github.com/ooni/sysadmin/pull/361

All of our work on improving the availability of our infrastructure was tracked through the following ticket: https://github.com/ooni/sysadmin/issues/359

Published report on the blocking of abortion rights websites

In collaboration with Coding Rights (Brazilian partner) and Women on Waves / Women on Web, we published a report that examines the blocking of abortion rights websites womenonwaves.org and womenonweb.org around the world.

The report is available on the following sites:

- Coding Rights: https://medium.com/codingrights/on-the-blocking-of-pro-choice-websites-women-on-waves-and-women-on-web-505ed6f17b63

VPNCompare wrote an article about our report: https://www.vpncompare.co.uk/block-abortion-websites/

In this report, we share OONI data confirming the blocking of womenonwaves.org in Brazil, as well as in Iran and Turkey.

We also share OONI data on the blocking of womenonweb.org in South Korea, Turkey, and Saudi Arabia.

We corroborated OONI findings by looking at Censored Planet data (https://censoredplanet.org/) as well, shared in the report.

Created community resources

To strengthen existing partnerships and better support OONI community engagement efforts, we created several new resources, listed below.
Created OONI Frequently Asked Questions (FAQ) section

To better support the OONI community, we created a Frequently Asked Questions (FAQ) section: https://github.com/TheTorProject/ooni-web/pull/301

The OONI FAQ can also be viewed here: https://deploy-preview-301--ooni.netlify.com/about/faq/

The new OONI FAQ includes answers to 67 questions that fall under the following themes:

- About OONI
- OONI Probe
- Testing websites
- OONI data
- OONI Explorer

We included the questions that we have most frequently been asked over the last years, as well as questions that have been raised by community members during meetings, hackathons, workshops, usability studies, and during other community interactions.

The new FAQ (along with all other content on the OONI website) is published on github, and we welcome feedback and community contributions. We aim to publish the FAQ on the OONI website and promote it over the next few weeks.

As OONI tools and methodologies evolve, we will continue to update the FAQ on an ongoing basis.

Created OONI Glossary

To empower broader community participation in censorship measurement research, we created a glossary which explains the terms used in the OONI apps, methodologies, and research reports.

The new OONI Glossary is available on github (and we encourage review, feedback, and community contributions): https://github.com/TheTorProject/ooni-web/pull/298

The glossary can also be viewed here: https://deploy-preview-298--ooni.netlify.com/about/glossary/

We aim to publish the glossary on the OONI website over the next few weeks and we will promote it through our partner networks.

Published OONI workshop slides
To enable our partners and the broader community to teach OONI Probe and engage their local communities with censorship measurement research and advocacy, we created and published OONI workshop slides:
https://docs.google.com/presentation/d/1UAxGeF1NhCXc8pT7cfWTp0NPdkWB5LInBkGfg!usp=sharing

These workshop slides cover the basics with regards to:

- OONI Probe test methodologies
- Running OONI Probe
- Potential risks associated with running OONI Probe
- How to reduce potential risk associated with running OONI Probe
- Contributing to test lists
- Customized website testing
- OONI Run: Coordinating custom OONI Probe testing
- Using OONI Explorer
- OONI API
- Interpreting OONI data
- Using OONI data
- How to get involved with the OONI community

We have published these OONI workshop slides on Google Slides so that community members can:

- download the slides in the format of their choice
- edit and customize the slides (so that they're more relevant to their local context and to the needs of the communities they teach)

Our edits to the OONI website to accommodate the publication of the OONI workshop slides are available here: https://github.com/TheTorProject/ooni-web/pull/302

The OONI workshop slides can be found through the Get Involved section of the OONI website: https://ooni.org/get-involved/

**Published document for OONI Probe community engagement**

Our partners have expressed the need for a document that they can share to introduce new people to the use of OONI Probe.

To this end, we created a PDF document that summarizes key information for getting started with running OONI Probe and engaging your community with censorship measurement research:

Specifically, this document covers:

- About OONI Probe
● Running OONI Probe
● Testing websites
● Testing specific websites of your choice
● Testing a custom list of websites

The above information is shared through instructions, synthesized information, and links for further reading.

Our edits to the OONI website to accommodate the publication of this document are available here: https://github.com/TheTorProject/ooni-web/pull/302/commits/6f61b9e9b672443ee151c1925b8a5ea47b7966a6

This document can be found through the Get Involved section of the OONI website: https://ooni.org/get-involved/

**Updated OONI Run button page**

Often when censorship events emerge (leading up to or during political events, such as protests or elections), social media, news media, and VPN sites are commonly blocked.

To enable the testing of specific types of websites and to support rapid response to emergent censorship events, we set up a page which includes OONI testing buttons: https://ooni.org/get-involved/run/

These buttons enable the OONI Probe testing of:

● Social media websites
● News media websites
● VPN sites
● Wikipedia sites
● Human rights sites
● Environmental sites
● LGBTQI sites

Rather than having to identify which social media websites (for example) to test during an urgent censorship event, the OONI community can refer to the buttons on this page for targeted testing. Apart from supporting efforts related to rapid response, these OONI Run buttons can also support research and advocacy efforts related to the customized testing of the above categories of sites over time.

**Community use of OONI data**

**Investigating the blocking of sci-hub and libgen in France**
Community members hosted an OONI workshop in France to engage locals with the use of OONI Probe in order to test the blocking of sci-hub and libgen across different networks.

They shared their OONI Probe methodologies and findings through a series of tweets: https://twitter.com/tenacioustek/status/1183715300283502592

They also shared the script they wrote to query results directly from the OONI API: https://github.com/Te-k/analyst-scripts/blob/master/ooni/get_ooni_website_status.py

Community activities

**HacktoberFest 2019**

OONI participated in HacktoberFest (https://hacktoberfest.digitalocean.com/):
https://twitter.com/OpenObservatory/status/1179033711557263361

The OONI github issues that were part of HacktoberFest are available here:
https://github.com/search?q=org%3Aooni+label%3Ahacktoberfest

We received valuable contributions from HacktoberFest participants! Across projects, we received 6 pull requests. Notably, the newly launched OONI Explorer received 4 pull requests, one of which was a major refactoring exercise:
https://github.com/ooni/explorer/pull/336

**OONI workshop in Taiwan**

Our Taiwanese partners, the Open Culture Foundation (OCF), hosted an OONI workshop in Taipei on 17th October 2019.

Information about their workshop is available here:
https://ocftw.kktix.cc/events/iffcc201910

**Manual for OONI Probe rapid response in Chile**

Our Chilean partners, Derechos Digitales, created and published a manual (in Spanish) which explains how to use OONI Probe:

The use of OONI Probe was heavily promoted by Derechos Digitales during the protests in Chile: https://twitter.com/derechosdigital/status/1187101399001260032

**Blog post about OONI Probe**
The Civilsphere Project (dedicated to supporting NGOs, journalists, and human rights defenders) published a blog post (also in Spanish) where they explain OONI and how to use OONI Probe (likely in response to the protests in Chile): https://www.civilsphereproject.org/blog/2019/10/22/que-es-ooni-y-como-usarlo

In their blog post, Civilsphere discuss why it's important to measure internet censorship, what OONI Probe can measure, how to run OONI Probe (and associated potential risks), how to configure OONI Probe settings, and how to find OONI data.

Press coverage

CNET published an article about OONI (titled: "These watchdogs track secret online censorship across the globe"): https://www.cnet.com/features/the-watchdogs-tracking-secret-online-censorship-across-the-globe-ooni/#ftag=COS-05-10aaa0j

OONI Probe was also listed as a tool to fight authoritarianism in the following article: https://toolsandnotions.comprehensophy.org/technologies-that-counter-authoritarianism/

Userbase

In October 2019, OONI Probe was run 485,623 times from 5,561 different vantage points in 214 countries around the world.

This information can also be found through our stats: https://api.ooni.io/stats

~ The OONI team