Programming Assurance with Network Assistance (PrANA) Responses

Roger Dingledine

The Tor Project

Q1: What's a concrete example of one or more security goals that are difficult to achieve using the abstractions and functionality provided by today's Internet?

Concrete example #1: Securing communications metadata. The internet community is getting better at integrating end-to-end encryption into application-level protocols like Signal and WhatsApp, but most communication on the internet still leaks traffic information like who you're talking to, when you're talking, and how much you're talking.

Adversaries can collect this traffic information from a variety of vantage points: watching the end user's network connection, watching (or being) the destination site, or observing backbone internet links.

Protecting the content of the traffic is important, but in many situations the most sensitive information comes from communication partners (connecting to a cancer survivor support forum, talking to a journalist) or locations (connecting to Facebook from a hotel room in a different city than usual). We worry both about targeted individual attacks (e.g. using network metadata to discover the identity of the whistleblower) and also broad collection of browsing and communication habits across populations, which can be used to 'go back in time' and answer questions once laws or norms change.

Concrete example #2: Preventing browser (application level) tracking. In addition to the network level communications metadata, we also need to protect application level communications metadata. In the browser case, this information includes the usual example of cookies, but also includes many more subtle issues like which fonts are installed, the size of the browser window, which languages are preferred, processor (and thus timing) characteristics, and more.[1]

Popular browser vendors like Mozilla and Google have rolled out 'private browsing' or 'incognito' modes, but the key flaw in their approach is that they only consider a local adversary looking at the computer after a browsing session: they do not consider a network adversary, or a destination site, who attempts to learn about or track users. In fact, Duck Duck Go did a user study asking people what they thought private browsing mode protects, and found that the majority of people expect it to protect against a network adversary when the reality is that it does not.[2]

Separating and isolating browsing activity to different destinations during a browsing session should consider correlation on both the network layer (e.g. separate paths) and the application layer (e.g. first party isolation for third party cookies, and bundling all requests for a given page onto the same path).

Concrete example #3: Blocking resistance at the network level. Around the world, people are prevented from reaching a growing number of ordinary websites, with website categories ranging from news to political information to entertainment.

Many people assume the problem is isolated to authoritarian regimes, but the reality is that network censorship is becoming commonplace in 'Western' countries like England, Australia, and Sweden.

Worse, because designers of modern networks value control, they tend to design with centralization in mind. Not only do these network bottlenecks make large-scale censorship more straightforward to deploy (and more consistent when it is deployed), but they also have second order effects like giving nation-states the tools to do country-wide 'internet shutdowns' whenever they want.

These network chokepoints are also the perfect locations for surveillance and censorship middleboxes. Modern vendors, including those from the United States and Europe, offer "Deep Packet Inspection" features, flow reconstruction, quick and frequent updates with new rulesets, and all the other control features that modern network operators have come to expect.

[1] https://www.torproject.org/projects/torbrowser/design/

[2] https://duckduckgo.com/download/Private_Browsing.pdf

Q2: If we had the ability to change the way that networks were built and operated, how could we achieve these goals in a fundamentally better way?

Two pieces to solving the network-level metadata problem are **indirection** and **address rewriting**. That is, you need to send your traffic to an address that doesn't give away your intended actual destination, and somewhere along the path we need to rewrite the source address so the destination doesn't learn it. Notice that these words sound a lot like Network Address Translation (NAT).

The current best-of-breed solution to all three of these security goals is the Tor ecosystem, which includes:

- tor, the network level daemon
- Tor Browser, a fork of Firefox that isolates activity between Tor circuits and disables a broad set of browser tracking vectors
- Pluggable transports, which offers a modular process for "plug-ins" that transform Tor traffic to make it harder for censors to recognize or block by destination or traffic characteristics

Users get all three of these components when they download the Tor Browser. But each component can also be useful individually; for example, other censorship circumvention tools use our pluggable transports to transform their own traffic to make it harder to block.

One of the critical security properties of the Tor approach is "distributed trust": there is no point in the network that can match up users to their destinations. That security property makes it fundamentally more powerful than the traditional Virtual Private Network (VPN) approach, where a centralized provider redirects traffic but still gets to watch both sides of the traffic flows.

But it is not straightforward to scale the Tor overlay network approach to full internet size: each of the messy pieces still has its own open research questions, like how clients should maintain knowledge of available relays and their keys, how to load balance properly and assess relay capacity as the network scales, how to handle congestion and denial-of-service issues, how to assess and maintain diversity of relay locations as the network grows, and how to incentivize enough network operators to provide capacity.[3]

To tie it all together, these three concrete steps will get us closer to solving these security goals at the network layer:

- For network metadata protection: we need more research attention on taking the lessons we've learned from Tor as an overlay network and applying them to an approach in the network layer itself.
- For browser tracking: the end goal is to integrate Tor into browsers, so they can provide the Private Browsing Mode that their users already think they are getting. Brave already integrates Tor, and Firefox wants to.[4]
- For censorship: having a proper infrastructure for protecting communications metadata is key, because *censorship implies surveillance*: if the network can't learn where you're going, it can't decide whether you're allowed to get there. But the existence of that infrastructure somewhere on the internet isn't enough: censors who can block the entire metadata security mechanism can deny users its security properties. Current known techniques to ensuring that users can reliably reach that infrastructure involve using protocols and destination addresses that blend in with "expected" background traffic, both in terms of content (e.g. distinguishing Tor's handshake from other expected protocols) and by traffic characteristics (packet size, volume, and timing).

In closing, we should bear in mind that these next steps involve solving not just the scientific problems, but also tackling social problems, commercial business incentives, national security desires to retain centralized control and knowledge, etc.

[3] https://blog.torproject.org/tors-open-research-topics-2018-edition#performance
[4] https://mozilla-research.forms.fm/mozilla-research-grants-2019h1/forms/6510