

The Future of HTTP and Anonymity on the Internet

Mike Perry
The Tor Project, Inc
mikeperry@torproject.org

Georg Koppen
The Tor Project, Inc
gk@torproject.org

Abstract

The Tor Project has a keen interest in the development of future standards involving the HTTP application layer and associated transport layers. At minimum, we seek to ensure that all future HTTP standards remain compatible with the Tor Network, avoid introducing new third party tracking and linkability vectors, and minimize client fingerprintability. We also have a strong interest in the development of enhancements and/or extensions that protect the confidentiality and integrity of HTTP traffic, as well as provide resistance to traffic fingerprinting and general traffic analysis. In fact, we are presently researching these areas.

1 Introduction

The Tor Project is a United States 501(c)(3) non-profit dedicated to providing technology, research, and education to support online privacy, anonymity, and censorship circumvention. Our primary software products are the Tor network software, and the Tor Browser, which is based on Firefox. The Tor Project is actively collaborating with Mozilla to ensure that our modifications to Firefox are merged with the official Firefox distribution, with the long-term goal of providing an optional Tor-enabled mode of operation for native Firefox users.

In this position paper, we describe the concerns and interests of the Tor Project with respect to future HTTP standardization. These concerns and interests span six areas: identifier linkability, connection usage linkability, fingerprinting linkability, traffic confidentiality and integrity, traffic fingerprinting and traffic analysis, and Tor network compatibility.

Each of these areas is communicated in a separate section of this position paper. We have also performed a preliminary review of HTTP/2 with respect to these areas, and have noted our comments in each section. We will be performing a more in-depth review of HTTP/2 for client fingerprinting and other tracking issues in the coming months as we modify the Firefox implementation for deployment in Tor Browser.

2 Identifier Linkability Concerns

Identifier linkability is the ability to use any form of browser state, cache, data storage, or identifier to track (or link) a user between two otherwise independent actions. For the purpose of this position paper, we are specifically concerned with any stateful information residing in the browser's HTTP layer that persists beyond the duration or scope of a single connection.

For background, the Tor Project has designed Tor Browser with two main properties for limiting identifier-based tracking: First Party Isolation, and Long Term Unlinkability.

First Party Isolation is the property that a user's actions at one top-level URL bar domain must not be correlated or linked to their actions on a different top-level URL bar domain. We maintain this property through a number of patches and modifications to various aspects of browser functionality and state keeping[1].

Long Term Unlinkability is the property that a user's future activity must not be linked or correlated to any prior activity after that user's explicit request to sever this link. Tor Browser provides Long Term Unlinkability by allowing the user to clear all browser tracking data in a single click (called "New Identity")[2]. Our eventual

goal is to allow users to define their relationship with individual first parties, and alter that relationship by resetting or blocking the associated tracking data on a per-site basis.

2.1 Identifier Linkability in HTTP/2

The Tor Project is still in the process of evaluating the stateful nature of HTTP/2 connections and their associated streams and settings. It is likely that we will be able to isolate the usage of HTTP/2 connection state in a manner similar to the way we currently isolate HTTP/1.1 connection state. However, it is not clear yet at this point how complicated this isolation will be.

2.2 Avoiding Future Identifier Linkability

We feel that it is very important that mechanisms for identifier usage, storage, and connection-related state keeping be cleanly abstracted and narrowly scoped within the HTTP protocol. However, we also recognize that to a large degree identifier storage and the resulting linkability is primarily an implementation detail, and not specific to the protocol itself.

Identifier linkability will become a more serious problem in future HTTP versions if the server is allowed to specify a setting or configuration property for a client that must persist beyond the duration of the session. In the case of Tor Browser, we will most likely clear this state immediately upon connection close.

3 Connection Usage Linkability Concerns

Connection usage linkability arises from the use of the same underlying transport stream for requests that would otherwise be independent due to the first party isolation of their associated identifiers and browser state.

Tor Browser currently enforces connection usage unlinkability at the HTTP layer, by creating independent HTTP connections to third party hosts that are sourced from different first party domains, even if HTTP requests are issued simultaneously to the same third party site. These connections also use separate, isolated paths through the Tor network based on the domain of the first party that sourced them.

3.1 Connection Usage Linkability with HTTP/2

The heavy use of connection multiplexing in HTTP/2 may present additional complexities for ensuring that requests are isolated. Unfortunately, unlike identifier usage, connection usage linkability is encouraged by the HTTP/2 specification in Section 9.1 (in the form of specifying that clients SHOULD NOT open more than one connection to a given host and port)[4].

The Tor Browser will ignore this recommendation in its HTTP/2 implementation, and create an independent HTTP/2 connections to third parties for every first party domain that sources them.

3.2 Avoiding Future Connection Usage Linkability

In the future, connection usage linkability may become a more serious problem if the notion of a connection becomes disassociated from the application layer, and instead is enforced through a collection of identifiers or stateful behavior in the browser (for example, in the case of a connectionless datagram transport layer). This may tend to encourage implementations that make it difficult to decouple the notion of a session from the notion of a destination address, which will serve to entrench and cement cross-site third party tracking capabilities. Some user agent vendors already have implicit or explicit monetary incentives to make these implementation tradeoffs.

Connection (and even identifier) linkability could similarly arise if implementations were required to remember which endpoints supported which HTTP versions, to avoid wasting round trips determining this information

in-band. Implementations that choose not to store this state (to prevent the associated tracking vectors) may end up at an inherent performance disadvantage.

For these reasons, consideration should be taken to ensure that the specification does not encourage implementations to bake in deep assumptions about providing only a single connection instance per site, or otherwise implicitly encourage the browser to store information about a site's capabilities for long periods of time.

4 Fingerprinting Linkability Concerns

User agent fingerprinting arises from four primary sources: end-user configuration details, device and hardware characteristics, operating system vendor and version differences, and browser vendor and version differences.

The Tor Project is primarily concerned with minimizing the ability of websites to obtain or infer end user configuration details and device characteristics. We concern ourselves with operating system fingerprinting only to the point of removing ways of detecting a specific operating system version. We make no attempt to address fingerprinting due to browser vendor and version differences[3].

Under this model, it is unlikely that very many fingerprinting vectors that concern us will arise in the HTTP layer. However, the possibility for end user configuration details to leak into behaviors of the HTTP layer is still a possibility.

4.1 Fingerprinting Linkability in HTTP/2

The Tor Project is still in the process of evaluating client fingerprintability in HTTP/2. The largest potential source of fingerprinting appears to be in the SETTINGS frame. If clients choose setting values depending on end-user configuration, local network or related hardware capabilities, or operating system version, we may alter our implementation's behavior accordingly to remove this logic.

4.2 Avoiding Future Fingerprinting Linkability

It is conceivable that more fingerprinting vectors could arise in future versions of HTTP, especially if more flow control and stream multiplexing decisions are delegated to the client, and depend on things like local link layer properties, available system memory, available CPU cores, or other system details. Care should be taken to avoid these situations, especially in the specification of any highly-tuned datagram-based transport layer.

5 Traffic Confidentiality and Integrity Concerns

The Tor Project is very interested in any efforts to improve the confidentiality and integrity of the session layer of HTTP/3.

In particular, we are strong advocates for mandatory authenticated encryption of HTTP/3 connections. The availability of free and automated entry-level authentication through the Let's Encrypt Project[5] should eliminate the remaining barriers to requiring authenticated encryption. The creation of the Let's Encrypt certificate authority also causes us to strongly favor mandatory authenticated encryption over opportunistic unauthenticated or unauthenticated to authenticated upgrade mechanisms, despite our concerns with the certificate authority authentication model.

We are also interested in efforts to encrypt the ClientHello and ServerHello messages using an initial ephemeral handshake, as described in the Encrypted TLS Handshake proposal[6]. If SNI, ALPN, and the ServerHello can be encrypted using an ephemeral key exchange that is authenticated later in the handshake, the adversary loses a great deal of information about the user's intended destination site. When large scale CDNs and multi-site load balancers are involved, the ultimate destination would be impossible to determine with this type of handshake in place. This will aid in defenses against traffic fingerprinting and traffic analysis, which we describe in detail in the next section.

6 Traffic Fingerprinting and Traffic Analysis Concerns

Traffic fingerprinting is the process of using machine learning to classify web page visits based on their encrypted traffic patterns. It is most effective when exact request and response lengths are visible, and when the classification domain is limited by knowledge of the specific site being visited.

In the case of Tor, this attack is most commonly considered with respect to the client's connection to their Guard node (the entry into the Tor network). There, Tor's fixed 512 byte packet size, link encryption, and stream multiplexing go a long way to impede this attack for minimal overhead. The fixed 512 byte packet size helps to obscure some amount of request and response length information. Tor's link encryption also conceals the destination website from the Guard node observer, which reduces classifier accuracy and capabilities by increasing the size of the classification domain. Tor's stream multiplexing causes concurrent web page loads to blend together. In the face of concurrent multiplexed web page loads, the accuracy of these attacks drops considerably.

There was some initial controversy in the research literature as to the exact degree to which the classification domain size, the base rate fallacy, and other machine learning issues applied to website traffic fingerprinting of Tor traffic, but after we publicly requested that these effects be studied in closer detail[7], recent results have confirmed and quantified the benefits conferred by Tor's unique link encryption[13].

Tor's link properties are by no means a complete defense, but they show that there is room to develop defenses that specifically aim to increase the size of the classification domain and the base rate. Additionally, with a large base rate, it is our belief that minimal padding and clever use of request and response behavior will increase the false positive rate enough to prevent these attacks from being practical, even when some amount of prior information about the website in question is available. For this reason, we have been encouraging continued study of low-overhead defenses against traffic fingerprinting[9].

6.1 Traffic Analysis Improvements and Issues with HTTP/2

When website traffic fingerprinting was first studied in Tor, we developed an experimental defense against it that attempted to use HTTP/1.1 pipelining to randomize pipeline depth and request ordering to reduce the information available to classifiers[8]. Unfortunately, cursory experiments have revealed that this defense appears to provide questionable benefit, though exactly why has not yet been thoroughly investigated. We suspect it may be due to the lack of support for large pipeline depths (or any reliable HTTP/1.1 pipelining at all) on many sites.

We are hopeful that HTTP/2 will enable better request and response size and ordering randomization through the use of HTTP/2's client-configurable frame size and stream multiplexing properties, in addition to frame padding. Leveraging these features is high on the list of low-overhead defense experiments that the Tor Project is interested in evaluating when we pick up the Firefox implementation of HTTP/2 as part of our rebase to Firefox 38-ESR in the coming months.

However, in our preliminary investigation of HTTP/2, we also discovered that certain aspects of the protocol may aid certain types of traffic analysis attacks.

In particular, the PING and SETTINGS frames are acknowledged immediately by the client, which might give servers the ability to collect information about a client's location and/or routing via timing side-channels. They also allow the server to introduce an active traffic pattern that can be used for end-to-end traffic correlation or confirmation.

In Tor Browser, we will likely introduce delay or jitter before responding to these requests, and close the connection after receiving some rate of unsolicited PING or SETTINGS updates. However, lack of explicit guidance in the specification about this issue raises concerns about what frequencies of these frames are likely to represent normal server behavior in the wild due to overly-aggressive HTTP/2 implementations, as opposed to actual attacks.

It is true that there are other mechanisms that an attacker could use for the same purpose (such as redirects or Javascript), but these mechanisms can either be disabled by the user, discovered by UI indicators, or otherwise mitigated by Tor Browser.

6.2 Future Traffic Analysis Resistance Enhancements for HTTP/3

With the aid of an encrypted TLS handshake (to increase the classification domain and associated base rate), along with some additional padding features, we are hopeful that defenses against traffic fingerprinting will also be applicable to non-Tor TLS sessions as well. In addition to protecting the communications of non-Tor users from traffic fingerprinting, the application of these defenses to the HTTP layer will also serve to increase the difficulty of end-to-end traffic correlation and general traffic analysis of Tor exit node traffic.

In terms of assisting traffic analysis defenses, we would like to see capabilities for larger amounts of per-frame padding, and more fine-grained client-side control over frame sizes. Unfortunately the 256 bytes of padding provided by HTTP/2 is likely to be inconsequential when combined with the minimum frame size the client can request (16 kilobytes), unless we are additionally able to take advantage of Tor's 512 byte cell size in tandem.

In combination with researchers at the University of Leuven, the Tor Project has also developed a protocol[10] and prototype implementation[11] for communicating statistical schedules for asynchronous padding from Tor clients to Tor relays. The research community is currently in the process of evaluating the efficacy of this protocol against traffic fingerprinting and other traffic analysis attacks.

Pending the results of this analysis, these padding commands could form the basis of new HTTP/3 frame commands for communicating more sophisticated (yet still traffic-bounded) padding schedules to HTTP/3 servers.

7 Tor Network Compatibility Concerns

Our final area of concern is continued compatibility of the Tor network with future versions of the HTTP protocol. It is our understanding that there is a desire for future versions of HTTP to move to a UDP transport layer so that reliability, congestion control, and client mobility will be more directly under control of the client user agent.

At present, the Tor Network is only capable of carrying TCP traffic. While it will be possible to support the transit of UDP datagrams using our existing TCP overlay network without significant anonymity risks within a year's time or sooner, it is unlikely that this level of support will be sufficient to warrant the use of a finely-tuned UDP version of HTTP rather than a TCP variant.

Long term, our goal is to transition the entire Tor network to our own datagram protocol with custom congestion and flow control to better support both native datagram transport and end-to-end flow control. However, additional research is still needed to examine the anonymity implications associated with this transition[12]. Our present estimate is that a full network transition to UDP is at least five years away.

We are also concerned that even after a full network transition to a datagram transport, it is likely that the congestion, flow, and reliability control of a UDP version of HTTP may still end up performing poorly over higher-latency overlay networks such as ours.

For these reasons, we are especially interested in ensuring that overlay networks are taken into account in the design of any UDP-based future versions of HTTP, and also prefer to retain the ability to use future HTTP versions over TCP, should the UDP implementations prove sub-optimal for our use case.

References

1. Mike Perry, Erinn Clark, Steven Murdoch The Design and Implementation of the Tor Browser. Section 4.5: Cross-Origin Identifier Unlinkability. <https://www.torproject.org/projects/torbrowser/design/#identifier-linkability>.

2. Mike Perry, Erinn Clark, Steven Murdoch The Design and Implementation of the Tor Browser. Section 4.7: Long-Term Unlinkability via "New Identity" button. <https://www.torproject.org/projects/torbrowser/design/#new-identity>.
3. Mike Perry, Erinn Clark, Steven Murdoch The Design and Implementation of the Tor Browser. Section 4.6: Cross-Origin Fingerprinting Unlinkability. <https://www.torproject.org/projects/torbrowser/design/#fingerprinting-linkability>.
4. M. Belshe, R. Peon, M. Thomson Hypertext Transfer Protocol version 2. <https://http2.github.io/http2-spec/>.
5. Mozilla, Akamai, Cisco, EFF, Identrust, Automattic Let's Encrypt Certificate Authority. <https://letsencrypt.org/>.
6. M. Ray Transport Layer Security (TLS) Encrypted Handshake Extension. <https://tools.ietf.org/html/draft-ray-tls-encrypted-handshake-00>.
7. Mike Perry A Critique of Website Traffic Fingerprinting Attacks. <https://blog.torproject.org/blog/critique-website-traffic-fingerprinting-attacks>
8. Mike Perry Experimental Defense for Website Traffic Fingerprinting. <https://blog.torproject.org/blog/experimental-defense-website-traffic-fingerprinting>
9. Mike Perry, Erinn Clark, Steven Murdoch The Design and Implementation of the Tor Browser. Section 4.8: Other Defenses. <https://www.torproject.org/projects/torbrowser/design/#traffic-fingerprinting-defenses>
10. Mike Perry, Marc Juarez Multihop Padding Primitives <https://gitweb.torproject.org/user/mikeperry/torspec.git/plain/proposals/ideas/xxx-multihop-padding-primitives.txt?h=multihop-padding-primitives>
11. Marc Juarez WFPadTools https://bitbucket.org/mjuarezm/obfsproxy_wfpadtools/
12. Steven J. Murdoch Comparison of Tor Datagram Designs <https://research.torproject.org/techreports/datagram-comparison-2011-11-07.pdf>
13. Marc Juarez and Sadia Afroz and Gunes Acar and Claudia Diaz and Rachel Greenstadt A Critical Evaluation of Website Fingerprinting Attacks Proceedings of the 21th ACM conference on Computer and Communications Security (CCS 2014) <https://www.eecs.berkeley.edu/~sa499/papers/ccs-webfp-final.pdf>