Global, Passive Detection of Connection Tampering

Ram Sundara Raman[1,2] Louis-Henri Merino[3] Kevin Bock[4]
Marwan Fayed[2] Dave Levin[4] Nick Sullivan[2] Luke Valenta[2]
[1] University of Michigan [2] Cloudflare, Inc. [3] EPFL [4] University of Maryland

ABSTRACT

In-network devices around the world monitor and tamper with connections for many reasons, including intrusion prevention, combating spam or phishing, and country-level censorship. Connection tampering seeks to block access to specific domain names or keywords, and it affects billions of users worldwide with little-to-no transparency. To detect, diagnose, and measure connection-level blocking, "active" measurement techniques originate queries with domains or keywords believed to be blocked and send them from vantage points within networks of interest. Active measurement efforts have been critical to understanding how traffic tampering occurs, but they inherently are unable to capture critical parts of the picture. For instance, knowing the set of domains in a block-list (i.e., what *could* get blocked) is not the same as knowing what real users are actively experiencing (i.e., what is *actively* getting blocked).

We present the first global study of connection tampering through a *passive* analysis of traffic received at a global CDN, Cloudflare. We analyze a sample of traffic to all of Cloudflare's servers to construct the first comprehensive list of *tampering signatures*: sequences of packet headers that are indicative of connection tampering. We then apply these tampering signatures to analyze our global dataset of real user traffic, yielding a more comprehensive view of connection tampering than has been possible with active measurements alone. In particular, our passive analysis allows us to report on how connection tampering is actively affecting users and clients from virtually every network, without active probes, vantage points in difficult-to-reach networks and regions, or test lists (which we analyze for completeness against our results). Our study shows that passive measurement can be a powerful complement to active measurement in understanding connection tampering and improving transparency.

CCS CONCEPTS

• **Networks** → **Network measurement**; **Network reliability**; *Firewalls*; • **Social and professional topics** → **Censorship**.

KEYWORDS

Connection tampering; Censorship; Measurement

ACM SIGCOMM '23, September 10–14, 2023, New York, NY, USA

ACM ISBN 979-8-4007-0236-5/23/09…$15.00
https://doi.org/10.1145/3603269.3604875

ACM Reference Format:
Ram Sundara Raman, Louis-Henri Merino, Kevin Bock, Marwan Fayed, Dave Levin, Nick Sullivan, and Luke Valenta. 2023. Global, Passive Detection of Connection Tampering. In *ACM SIGCOMM 2023 Conference (ACM SIGCOMM '23), September 10–14, 2023, New York, NY, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3603269.3604875

1 INTRODUCTION

Connection tampering occurs when middleboxes (e.g., firewalls) monitor traffic between clients and servers and disrupt connections containing forbidden content, often by injecting forged TCP RSTs or directly dropping packets [9, 84]. Many distinct entities have been shown to tamper with connections for myriad reasons, including: companies and college campuses that restrict what can be accessed from their networks [74], intermediary ISPs implementing government regulations on third party copyright [57, 58], and country-level censors that seek to limit free speech [18, 35, 54, 73].

Whatever the motivations behind those employing it, connection tampering is pervasive in today's Internet, affecting billions of users around the world, and thus it is critically important to understand it. From the perspective of a large network service provider such as a content delivery network (CDN), understanding connection tampering allows them to better identify and communicate network failures to their customers. From the perspective of free speech advocates, a global understanding of connection tampering allows them to track restrictions to Internet freedom [68].

There has been extensive prior work towards measuring traffic tampering (especially censorship) around the world, particularly at the application layer (TLS and HTTP), which is the focus of this paper. Prior efforts have comprised almost exclusively of *active* measurements, in which researchers generate network traffic that traverses a suspected censor's network [12, 20, 21, 31, 54, 66, 73]. Active measurement efforts like these have shaped the research community's understanding of connection tampering for decades.

However, active measurements alone face inherent shortcomings. First, because they are not driven by real user data, they can only measure what *could* be tampered with, rather than what is *actively* tampered with. Second, active measurements generally require vantage points within the networks they are investigating, thereby restricting the study of many kinds of networks (especially cellular networks) and regions with low Internet penetration. Finally, active measurements require a list of domains and keywords to test for tampering [43], but test lists themselves are incomplete and inherently slow to react to changes in tampering policies, often requiring alerts from on-the-ground volunteers.

In this paper, we present a novel measurement study with a global CDN, Cloudflare, that addresses the above challenges by relying on *passive* measurement of real user traffic. The central mechanism behind our techniques is a set of *tampering signatures* that are highly indicative of tampering by a network intermediary.

We applied our techniques to a sample of connections from every Cloudflare web server in more than 285 points of presence (PoPs), that serve ~17–20% of the Internet's websites and services [80, 81]. Accordingly, our dataset spans virtually every network in the world, including cellular and enterprise networks, and networks in countries with low Internet penetration, all of which are challenging for active measurement-based techniques. Overall, our approach addresses the primary shortcomings facing active measurement-based approaches today: measuring connection tampering as it affects users' experience and services' reachability—without needing to procure vantage points inside the countries of study.

This is of course not to say that passive techniques ought to replace active measurements! Instead, this work shows that passive measurement can complement active efforts; only together can they obtain a more complete picture of the state of global connection tampering on the Internet. Furthermore, it is our hope that sharing the observations from this global study helps to facilitate dialogue among operators and their customers.

Contributions We make the following contributions:

- We are the first to design and deploy a fully passive technique to globally measure traffic tampering and censorship (§3).
- We perform the first *comprehensive* collection and analysis of tampering fingerprints, significantly extending prior efforts [84] (§4).
- We apply our tampering signatures to 0.01% of all traffic to Cloudflare's global network for a period of two weeks, resolving many operational challenges (§3 and §4).
- We report on a wide array of insights into global connection tampering and censorship at a scale not previously possible, including its frequency (§5.2) and the affected types of content (§5.4) across various regions.
- We evaluate the coverage of existing test lists, demonstrating that they miss many domains that are affected by connection tampering (§5.5).
- We present a case study that sheds new light onto Iranian censorship in the wake of the 2022 protests (§5.6).

Data sharing We plan to provide up-to-date datasets for the foreseeable future at `https://radar.cloudflare.com`. Due to privacy policies, we are unable to provide raw data about Cloudflare's users (i.e., specifically what IP addresses experienced tampering) nor its customers (i.e., specifically what domains were tampered with). What we lose by not being able to provide such details, we gain by providing a much wider understanding of how user traffic is affected by tampering than has been achieved before.

2 BACKGROUND AND RELATED WORK

In this section, we provide a brief background on how connection tampering occurs and review prior work that measures connection tampering primarily through the lens of Internet censorship.

2.1 How Tampering Occurs

Tampering refers to the intentional manipulation or blocking of communication by a network intermediary (a middlebox). Tampering can happen at one or more connection stages, including the DNS resolution [42, 63], TCP handshake [18, 62, 66], and the TLS handshake or HTTP request [74, 78]. In this paper, we focus mainly on the passive detection of tampering at or above the TCP layer—what we collectively refer to as *connection tampering*—though we also infer some IP address blocking.

Middleboxes identify unwanted traffic predominantly by performing deep-packet inspection (DPI) and looking for forbidden domain names in HTTP `Host` headers or TLS Server Name Indicator (SNI) fields, or keywords in HTTP `GET` requests. DPI works for these protocols because these fields are in cleartext: HTTP has no encryption, and the SNI appears unencrypted in the TLS handshake[1].

Tampering middleboxes disrupt ongoing connections in myriad ways [5]. Some simply drop unwanted packets altogether [32, 44]. Measurement studies of Internet censorship have observed that censoring middleboxes tend not to drop traffic, but rather disrupt connections by injecting tear-down (TCP `RST`) packets [17, 18, 48, 82] or manipulated responses [6, 36, 63]. This has immediate relevance to our approach; because these middleboxes permit the original offending traffic to reach the destination (our servers), we are able to observe both the censorship event (e.g., the `RST`) *and* the traffic that induced it. We are also able to observe tampering by dropping, as they appear as suspiciously short connections, but because the offending packet does not reach us, we are not able to infer what caused the blocking.

2.2 Active Measurements

The vast majority of prior work in measuring connection tampering has been done in the context of studying Internet censorship, a highly active research field [3, 4, 8, 16, 21, 25, 26, 29, 33, 34, 41, 43, 51, 62, 63, 71, 75, 78, 79, 86]. Although connection tampering is broader than censorship in the conventional sense, we focus our review of prior work on censorship given how thoroughly it has been studied. Numerous dedicated projects, such as Censored Planet [73], OONI [35], and ICLab [54] measure blocking of specific Internet content, while the IODA project [44] measures Internet shutdowns. In the broadest sense, this area seeks to understand what traffic is being censored, and to do so in a manner that is broad, longitudinal, and ethical.

Most of the above approaches rely on active measurements, in which the researchers send packets probes into, out of, or across networks suspected of censorship. Active measurements of censorship require soliciting participation and constructing test lists, which introduce their own sets of limitations:

Soliciting participation Active measurement requires observing traffic that has traversed a censoring middlebox. Some tools use machines within a country of interest to generate requests, for example via hosting [7, 8, 10, 29, 34, 35, 43, 52, 79, 86]. Others send traffic into specific countries from the outside [56, 62, 63, 73, 74]. An alternative is to solicit volunteers with informed consent [4, 35, 41, 79], while some tools have taken the controversial approach of co-opting users' machines inside censoring regions without informed consent [21, 53].

Soliciting participation from users—even when done properly, with informed consent—is both challenging at scale and risky, with

[1] TLS 1.3 offers an encrypted Client Hello, but even that has a cleartext SNI. The earlier encrypted SNI proposal was blocked by China entirely [19].

debates ongoing about best practices and ethics [46, 61]. Our measurement system is distinct from these in that it does not require active participation from users, nor client vantage points under our control. In our study, our measurement vantage points are both the destination of clients' traffic as well as passive observers. This positioning shifts the ethical considerations from "how to obtain the information" to "what information can be reported." To protect the potential privacy considerations of the clients and the CDN's customer websites, we report aggregates and broad content topics rather than any individuals.

Test lists Active measurement of censorship generally involves first constructing a *test list* of potentially blocked domains or keywords and sending traffic containing entries from them. Several published test lists are used across many studies, the most popular of which coming from the Citizen Lab [23], Herdict [39], GreatFire [38], Berkman Klein [13], as well as top-K website lists [47, 49].

However, test lists have two prominent limitations: First, one can never be certain about the list's *completeness*. Indeed, as we show in §5.5, today's test lists miss a considerable fraction of tampered domains. Second, even if test lists are complete, they are unable to measure users' experience of blocking; at best, test lists indicate what *can* be blocked, not what users are *actively* trying and unable to access. Our passive techniques address both of these limitations of test lists by detecting tampering of connections generated by real users. Our techniques can be used to learn what domains and keywords are being tampered with, which can help inform future test lists.

2.3 Passive Measurements

We are aware of two efforts to passively measure connection tampering. In 2005, Arlitt and Williamson used a year's worth of data from a campus network [9] and found ~15% of connections were terminated by `RST` packets. Closest to our work is a 2009 study by Weaver et al. that explored ways to passively detect forged and injected `RST` packets in four campus networks [84]. The authors constructed tampering signatures associated with injected `RST` packets, and then identified corresponding commercial middleboxes. What primarily distinguishes our work from these prior studies is our large-scale deployment; we are able to identify and apply tampering signatures at an unprecedented global scale. Our methodology is informed by that of Weaver et al., however the global view we obtain by deploying at a popular CDN reveals significantly more indicators of `RST`-based tampering (19 high-confidence signatures compared to 6). We additionally find signatures indicative of packet drops.

3 DESIGN

This section describes the design of our system for passively detecting connection tampering based on packets received at a large CDN. We also discuss the various constraints and ethical considerations that such a tool imposes on our data collection.

3.1 Design Rationale

A standard, successful TCP connection for HTTP or HTTPS content comprises three broad stages: (1) A TCP three-way handshake, typically comprising a `SYN` from the client, a `SYN+ACK` from the server, and an `ACK` from the client; (2) a series of `PSH`, `PSH+ACK`, and `ACK` packets to send and acknowledge data; and finally (3) a graceful connection termination by exchanging `FIN` and `FIN+ACK` packets. In the event of an error or failure, clients or servers can also send `RST` packets (or `RST+ACK` packets in response to an unsolicited `SYN`) [1]. A `RST` instructs the recipient to immediately terminate the connection and discard state.

When middleboxes tamper with a connection, they seek to stop the server from sending the requested content to the client. They do this in one of two broad ways[2]:

- **Packet drops**: The most straightforward way to cease communication between a client and server is for middleboxes to drop the packets they send to one another [32, 75]. In such cases, the server sees all packets prior to the triggering event, after which it would appear the client became unresponsive. Although highly effective, packet dropping is not universally deployed because it requires tampering middleboxes to be *in the path* between client and server, which is resource-intensive [54, 75].
- `RST` **injection**: Another common form of tampering involves middleboxes forging packets with the `RST` bit set to the client and server, leading both to believe the other wishes to immediately terminate the connection [17, 18, 75, 82, 84]. In such cases, the server sees all the packets prior to and including the triggering event, immediately followed by what appears to be the client ungracefully terminating the connection. Note that `RST` injection does not require middleboxes to be in-path; they can simply obtain copies of the client and server's packets off-path.

While packet drops and `RST`s can occur in normal client communications, it is highly unlikely for them to occur precisely when censorship would occur. Many censorship events occur very early in a connection, often in response to the first `PSH` packet, which typically contains the domain in the clear (either in a GET request for HTTP or the SNI field in the TLS handshake for HTTPS). While not impossible, it is rare for a client to issue a request and then immediately ungracefully terminate the connection with a `RST` or become unresponsive. It would also be rare for a client to accidentally mimic known, idiosyncratic behavior of specific censors, such as the Great Firewall (GFW) of China, which sends multiple `RST+ACK`s immediately following an offending `PSH` [18, 48, 82].

The rationale behind our techniques is to use abnormal packet sequence "signatures" suggesting premature connection termination (e.g., `RST`s immediately following the first `PSH`) as potential indicators of traffic tampering.

3.2 Data Collection Methodology

We uniformly sample one of every 10,000 connections from among *all* TCP connections to *all* web servers at Cloudflare's more than 285 points of presence, with a total of more than 11,500 interconnections with other networks. Collectively, the system serves more than 45M HTTP requests per second on average (61M at peak) for millions of websites and services.

From *every* connection sampled, we collect the first 10 packets received at the server (from both client and middlebox), with full

[2] Alternatively or additionally, some middleboxes inject content to the client, such as a block page. As this paper focuses on signals visible to the server, we do not discuss this further.

headers and payloads. If the tampering middlebox does not block the traffic from the client, then the data we collect contains the packet(s) that triggered the connection tampering. We emphasize that our data is a sample of *all* connections, unfiltered by domain names or by the presence of `RST`s. We used this data to derive the tampering signatures reported in this paper. The integration of a data analysis pipeline into existing systems at that scale led to the following constraints on collection methodology.

First, only inbound packets are logged. Thus, our tampering signatures characterize only the packets from the client (and middleboxes), and not from the servers. The absence of outbound traffic turns out not to be a limitation. Our results show that it is possible to recognize, even identify, tampering signatures exclusively with inbound packets (§4).

Second, packets sequences may be logged out of order because timestamps in our dataset are only at a 1-second granularity. Fortunately, this also turns out not to be a limitation; we can typically reconstruct order with packet headers and sequence numbers (e.g., `SYN`s are followed by `SYN+ACK`s).

Third, samples consist only of the first 10 packets. While this precludes analysis of tampering later in a connection, most common types of connection tampering occur within the first few packets that identify a domain or service [17, 75]. Also, as HTTPS adoption increases, barring MitM attacks [18, 72], intermediaries have no visibility beyond TLS Client Hello packets. As a result, most tampering decisions are made before the TLS handshake is complete, usually within the first three ingress packets.

Finally, the sampling rate is one in 10,000 new connections. We believe any sampling bias to be minimal, but may miss types of tampering that are infrequent. The start of a TCP connection is marked by a `SYN` packet. To avoid oversampling, we sample connections only after packets are processed by the CDN's DDoS protection services, thereby filtering out most `SYN` flooding attacks. However, some attacks and retransmissions may still influence our detection of tampering (see §4.2).

3.3 Ethical Considerations

Our study underwent detailed review to ensure that data collection and processing was compliant with the CDN's policies. We also coordinated with our institution's IRB, which determined our study was exempt. Through detailed discussion with privacy experts at the CDN and following the guidelines established by the Menlo report [28], we established the following privacy constraints:

No traffic decryption The logging system that samples traffic has no ability to decrypt traffic, meaning that our visibility into packet contents is identical to any network observer. Furthermore, as described in §3.2, we use only the minimum information parsed from packets necessary to understand tampering behavior.

Data is analyzed and reported in aggregate Our dataset includes source and destination IP addresses as well as destination domain names and requested content, if available. However, we only analyze and report aggregated information to protect the CDN's customers and clients. In particular, our findings (§5) aggregate client information to the level of AS or country, and we report on domain *categories* rather than individual domains.

Access to data is restricted Our implementation passed internal security audits of Cloudflare before being deployed on production servers. Raw data is only available to specific employees with strict access controls. All processing and aggregation were performed on the CDN premises.

3.4 What the Data Does (Not) Say

The dataset we have collected is unique, and thus it merits clarifying what it does and does not allow us to reason about.

Who and what was affected, not who performed the tampering When tampering occurs, the data tells us the affected domain and source IP of the connection, but not where the tampering happened. Since the tampering could have occurred anywhere between the client and the web server, we cannot say for certain who tampered with the connection. Nonetheless, the global distribution of Cloudflare's servers suggests that the number of networks and regions between the two endpoints is relatively small.

What triggered the tampering (though not always) For tampering middleboxes that do not drop offending traffic, we are able to see all packets from the client. In many cases, the middlebox will not trigger tampering until it has seen data from the client (e.g., an HTTP GET or TLS SNI). This means that the trigger content is visible in our dataset, allowing us to reason about the domains that are affected without requiring us to have an *a priori* test list. The data does not provide insight into the precise trigger if the middlebox drops the offending packets. Prior work has demonstrated that in large country-level censors, in-path middleboxes are rare [18, 54, 75].

What *is* blocked, not what *could be* Our collection is strictly passive, and thus our data is purely client-driven. All of the tampering that we observe represents an instance where a real client's attempt to access content was stymied. This provides a unique perspective into the actual effects of tampering that active measurements alone are not able to obtain. That said, our technique is limited to what clients request; if no clients ever try to access a given domain, then we gain no insight into whether that domain would be tampered with. Active measurements therefore remain necessary to measure what constitutes censors' block lists.

We are only able to reason about the set of domains served by Cloudflare, but we believe that the millions of domains it serves is a representative cross-section of the entire web. Nonetheless, to mitigate potential bias we focus our analysis on categories of domains rather than individual names (§5.4).

4 TAMPERING SIGNATURES

The central mechanism behind our techniques is a set of *tampering signatures*: sequences of TCP packet header flags that are indicative of a tampering event. Our system passively monitors (sampled) connections and compares the packets to our signatures. In this section, we describe the first *comprehensive* list of tampering signatures, and an evaluation of their ability to detect tampering.

Type	Signature	Description	Prior Work
Post-SYN	⟨SYN → ∅⟩	No packets after a single SYN	[16, 32, 62]
	⟨SYN → RST⟩	One or more RSTs after a single SYN	[84]*, [15, 62]
	⟨SYN → RST+ACK⟩	One or more RST+ACKs after the SYN	[84]*, [15, 62]
	⟨SYN → RST; RST+ACK⟩	One or more RST and RST+ACK after a single SYN	[20]
Post-ACK	⟨SYN; ACK → ∅⟩	No packets received after a SYN and an ACK	[10, 12, 15, 16, 75]
	⟨SYN; ACK → RST⟩	Exactly one RST after a SYN and an ACK	[84]*, [10, 12, 22]
	⟨SYN; ACK → RST; RST⟩	More than one RST after a SYN and an ACK	[15, 22]
	⟨SYN; ACK → RST+ACK⟩	Exactly one RST+ACK after a SYN and an ACK	[84]*
	⟨SYN; ACK → RST+ACK; RST+ACK⟩	More than one RST+ACK after a SYN and an ACK	—
Post-PSH	⟨PSH+ACK → ∅⟩	No packets received after PSH+ACK packets	[12, 19, 88]
	⟨PSH+ACK → RST⟩	Exactly one RST	[14, 48, 74, 82, 83]
	⟨PSH+ACK → RST+ACK⟩	Exactly one RST+ACK	[14, 48, 74, 82, 83]
	⟨PSH+ACK → RST; RST+ACK⟩	At least one RST and one RST+ACK	[20]*, [82, 83]
	⟨PSH+ACK → RST+ACK; RST+ACK⟩	At least two RST+ACKs	[20]*, [82]
	⟨PSH+ACK → RST = RST⟩	More than one RST; same ACK numbers	—
	⟨PSH+ACK → RST ≠ RST⟩	More than one RST; change in ACK numbers	[84]*
	⟨PSH+ACK → RST; RST_0⟩	More than one RST; one of the ACK numbers is zero	—
Post-Multiple Data Packets	⟨PSH+ACK; Data → RST⟩	One or more RSTs not immediately after first PSH+ACK	—
	⟨PSH+ACK; Data → RST+ACK⟩	One or more RST+ACKs not immediately after first PSH+ACK	—

Table 1: The comprehensive set of tampering signatures we identify through global passive measurements. The signature names have the format $X \rightarrow Y$, where X is the set of packets sent prior to the tampering event and Y is the set of packets sent after. "∅" denotes packet drops. Only prior work with a "*" identify the exact signature; other prior work identify the general phenomenon (e.g., "packet drops after client hello").

4.1 Signatures Detected

We first enumerate the list of packet sequences in our data where we do not see graceful termination of the TCP connection. We consider connections to have not terminated gracefully if they include a packet with the RST flag set or if they exhibit a 3-second inactivity within the recorded 10 packets without a FIN handshake. We label these as *possibly tampered connections*. This forms a *superset of the connections* that match our signatures characterized by early termination and RST injection. Possibly tampered connections account for 25.7% of all connections in our dataset for the two-week period from January 12–January 26, 2023. We group possibly tampered connections by their TCP packet flags, and investigate the groupings in decreasing order of frequency, cross-referencing these groupings with those identified in prior work that detected censorship and packet injection [20, 84].

In total, we investigated more than 700 unique packet groupings for possibly tampered connections and consolidated them into 19 unique *signatures* that are indicative of tampering. These 19 signatures cover 86.9% of all possibly tampered connections, showing that it is possible to manually enumerate the patterns observed in anomalous traffic that may be caused by tampering. Table 1 presents our comprehensive list of tampering signatures, their description, and previous works on Internet censorship that have recorded the effect indicated by these signatures. Each tampering signature ($X \rightarrow Y$) comprises of the set of packets preceding the tampering event (X) and the set of packets following it (Y).[3] We describe them in turn here, broken down by how far into the connection the tampering takes place.

[3] ∅ denotes having received no packets for more than three seconds.

Mid-handshake The first class of signatures we identify occur before the TCP three-way handshake has completed. This packet sequence type accounts for 43.2% of possibly tampered connections, of which 99.5% match one of our tampering signatures. All of these involve a SYN packet from the client, but no corresponding ACK packet. Tampering at this stage is likely triggered based on the destination IP address, as SYN packets typically do not contain any application-layer data (domain names or keywords). We observe some cases where the SYN includes the HTTP request payload, ostensibly an effort by web browsers to optimize web response times [9], or an effort to start a TCP amplification attack [14]. On January 17, 2023 we found that 38% of SYN packets on port 80 contained an HTTP request payload, with 93% of these requests to the same four domains. On port 443, only 0.02% of SYN packets contained a valid TLS Client Hello message.

We create two broad classes of signatures at this stage: (i) there are no packets observed on the connection after the SYN packet (⟨SYN → ∅⟩) and (ii) there are one or more RST or RST+ACK packets received after the SYN packet. Some tampering middleboxes such as China's GFW inject both a RST and a RST+ACK [20].

Immediately post-handshake The next set of signatures capture tampering that occurs after the TCP handshake but prior to the server receiving any further data. This packet sequence type accounts for 16.1% of the connections we denote as possibly tampered, of which 98.7% match one of our tampering signatures. Among these, we observed tampering via both packet dropping and RST injection. These typically indicate cases where the first data packet from the client, usually containing the start of a TLS handshake or HTTP GET request, is dropped by a censor.

This form of tampering has been observed in Iran. Aryan et al. [10] observed the post-handshake RST injection (⟨SYN; ACK →

RST⟩, ⟨SYN; ACK → RST+ACK⟩) in Iran in 2013. Specifically, they found that the censor dropped the client's offending request, replied to the client with a block page, and injected multiple RSTs to the server. More recently, Basso [12] observed the post-handshake packet dropping signature (⟨SYN; ACK → ∅⟩) in Iran in 2020. Our results confirm these prior findings, and also discover other regions whose traffic is subjected to the same form of tampering. For example, we show in §5 that, while these signatures constitute 34.4% of tampered connections from networks in Iran, they also constitute over 70% of tampered connections from Sri Lanka networks and over 81% from Turkmenistan networks.

We split these signatures based on the number of RST or RST+ACK packets received in order to enable the possibility of differentiating different tampering systems. However, our analysis (§B) shows this split might be of limited utility, because many tampering systems inject varying number of RST or RST+ACK packets to ensure connection termination.

After first data packet The next and most diverse form of tampering signature applies after the server receives the first data packet (PSH+ACK) from the client, which usually contains the TLS Client Hello or HTTP request. This packet sequence type accounts for 5.3% of possibly tampered connections, of which 97.9% match one of our tampering signatures. These also include both packet drops and RST injection. For the sole packet-drop signature among these (⟨PSH+ACK → ∅⟩), we observe that no packets are allowed from the client after the PSH+ACK packet.

Several of the RST and RST+ACK signatures have been identified in prior work. Bock et al. [20] reported the signatures involving multiple tear-down packets (⟨PSH+ACK → RST+ACK; RST+ACK⟩ and ⟨PSH+ACK → RST; RST+ACK⟩) when studying China's GFW. Weaver et al. [84] reported on middleboxes that inject two or more RST packets with different acknowledgment numbers (⟨PSH+ACK → RST ≠ RST⟩), ostensibly in an effort to guess the acknowledgment number of the next packet. We extend this idea to differentiate cases where one of the acknowledgement numbers is zero while another is not (⟨PSH+ACK → RST; RST_0⟩). To the best of our knowledge, the other tampering signatures we found in this set are novel.

After multiple data packets Finally, we consider connections that terminate after multiple data packets. This packet sequence type accounts for 33.0% of possibly tampered connections, of which 69.2% match one of our tampering signatures. We observe two kinds of early-terminated connections after multiple data packets from the client: those terminated by one or more RSTs and those terminated by one or more RST+ACKs. Manually investigating these cases suggests that these RSTs might be injected when triggered on specific keywords in cleartext (HTTP) connections, or by commercial devices that have visibility into encrypted traffic (e.g., firewalls in organizations that have trusted certificates installed on clients).

Other possibly tampered connections Another 2.3% of possibly tampered connections do not cleanly fall into one of the above connection stages (e.g., a connection terminated after a SYN and two ACKs). We do not cover these cases with our tampering signatures.

4.2 Validation: Threats to Validity

The tampering signatures described in Table 1 are designed to detect common forms of connection tampering. However, when applying these signatures to real-world data at large CDN scale, matching packet sequences could also be caused by other factors such as atypical clients and network attacks. Here, we describe what we anticipate to be the largest threats to validity, and empirically demonstrate that they are not significant.

Scanners One potential source of false positives for our tampering signatures is network scanners, like ZMap [30], that probe many servers [55]. Often, scanners do not comply with standards; ZMap, for instance, sends a single SYN to each target and, if the server replies with a SYN+ACK, ZMap simply replies with a RST (matching our signature ⟨SYN → RST⟩).

To gauge the effect of scanners on our tampering detection, we test what percentage of connections received by Cloudflare have the three properties known to be common in scanners, as described by Hiesgen et al. [40]: (1) Connections that have no TCP Options, (2) Connections that have a high TTL value (≥200), and (3) Connections with a fixed non-zero IP-ID. We find no connections without TCP options in our dataset, and only about 0.05% of all connections have a high TTL value (≥200). In the case of ZMap, however, we can build a finer-grained signature based on identifying static fields set in the initial SYN probe, including an IP-ID value of 54321 [40]. We find that only ~1% of the ⟨SYN → RST⟩ signature matches in our dataset can be confidently attributed to ZMap. Thus, although ZMap and other scanners do result in some level of false positives, we posit that this does not meaningfully impact our results.

SYN attacks Our data collection pipeline is executed after the CDN's DDoS protection services, but it is still possible that some attacks impact detection using our tampering signatures. The Post-SYN signatures are particularly susceptible to attacks such as SYN flooding, especially the ⟨SYN → ∅⟩ and ⟨SYN → RST⟩ signatures. It is also possible that connections matching these signatures are from spoofed source IP addresses, thus affecting our country-based results. Therefore, we restrict some of our results in §5 to our Post-ACK and Post-PSH signatures, which are more likely to be primarily caused by tampering.

Happy Eyeballs Some dual-stack clients may take advantage of the Happy Eyeballs mechanism to opportunistically make connections over both IPv6 and IPv4, the latter after a brief timeout (250–300ms), and drop or cancel one connection when the other succeeds [70, 85]. Clients using Happy Eyeballs may lead to some connections matching the ⟨SYN → ∅⟩ or ⟨SYN → RST⟩ signatures[4]. While we expect Happy Eyeball connections to have a minimal effect on our signatures because of their low prevalence [11], we again restrict some of our results in §5 to only Post-ACK and Post-PSH signatures.

4.3 Validation: Supporting Evidence

Ideally, for every tampering signature and for every client network, we would obtain a vantage point and try to trigger and verify the

[4]Upon inspection, we found that Chromium follows RFC 8305 and resets the unused connection [70], while some clients like Curl follow the older RFC 6555 and drop the unused connection [85].

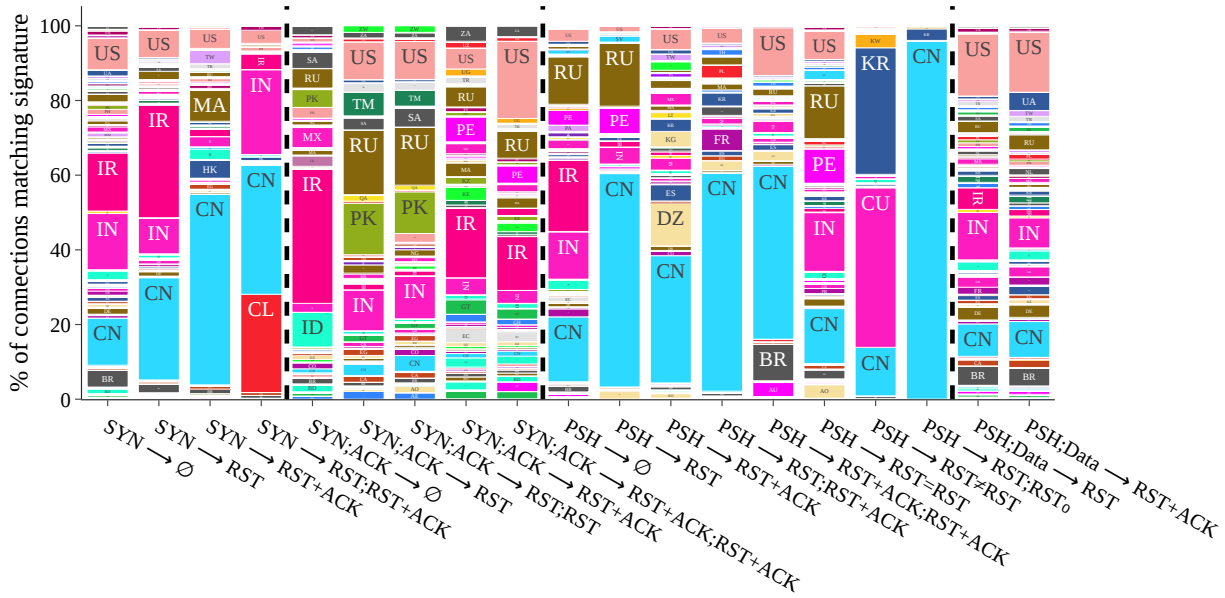


Figure 1: *Signature matching across countries*: Each column is the total global number of connections matching a specific signature. Within each column is the proportion of connections initiations from individual countries matching that signature.

behavior ourselves. The sheer scale of connections received by Cloudflare makes this infeasible, and many of the networks do not have vantage points. Instead, we seek the following supporting evidence by matching patterns in packet captures with expected behaviors in TCP/IP, as well as patterns and observations observed by previous active measurements.

Geographic distributions match censorship Figure 1 shows, for each tampering signature, the distribution of countries from where the connection was initiated for the period of January 12–26, 2023. Most tampering signatures exhibit a disproportionately large fraction of connections matching that signature from a small number of places. Moreover, these distributions do not match the baseline distribution of all connections to the CDN, indicating that the packet sequences are caused by region-specific network intermediaries, and not by some unknown client behavior. Often, the countries that originate the most signature-matching connections are those with known censorship systems, such as China (CN), Iran (IR), Russia (RU), and India (IN). In fact, certain signatures are exclusively observed only from certain countries, such as ⟨PSH+ACK → RST; RST_0⟩ in connections from China and South Korea (KR). This indicates that our signatures are capturing specific tampering behaviors of networks in these countries.

Some signatures, notably ⟨PSH+ACK; Data → RST⟩ and ⟨PSH+ACK; Data → RST+ACK⟩, match on connections from many countries. We observe that these connections frequently contain a HTTP user-agent value that indicates the presence of a commercial firewall, which are common across the globe [74]. Even in these cases, some countries appear more frequently than normal, such as Ukraine (UA) for the ⟨PSH+ACK; Data → RST+ACK⟩ signature.

Inconsistent changes in IP Identification (IP-ID) field The IP Identification (IP-ID) field in the IPv4 header was originally

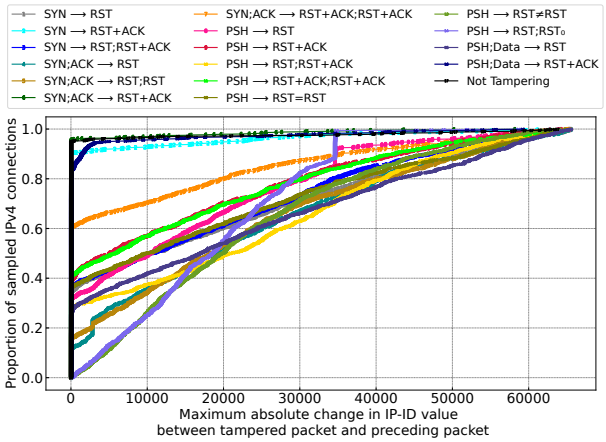


Figure 2: *IP-ID difference*: The figure shows the maximum absolute change in IP-ID value between the RST packet and the preceding non-injected packet in a connection (up to 1,000 connections for each signature).

designed to identify different fragments of a packet [64, 77]. With increasing link capacities, the use of IP-IDs for this purpose has dwindled (and has even been excluded from the IPv6 header). Most modern operating systems either set the IP-ID to zero, use a *per-connection* counter, or use a globally incrementing counter in order to differentiate packets [32, 69, 77]. In all of these cases, packets in the same connection from a client will have a change in IP-ID values that is either zero or one.

However, a middlebox that is injecting, say, a RST packet, does not employ the same counter, and could thus use an IP-ID value that is very far from the IP-ID range used by the client. We use this insight for our evaluation. Among all connections sampled on

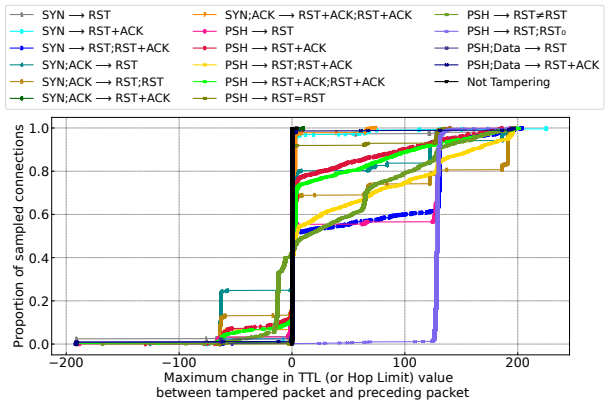


Figure 3: *TTL difference*: The maximum change in TTL (or hop limit) value between the RST packet and the preceding non-injected packet in a connection (upto 1,000 connections for each signature).

January 17, 2023, 93.4% have a *minimum* IP-ID difference of 0 or 1 and 4.2% of connections have a minimum IP-ID difference greater than 100. This indicates that IP-ID differences between packets in most connections are small, and we can use large changes in IP-ID as an indicator of injected packets.

The *maximum* IP-ID difference in a connection is similarly indicative. Figure 2 shows the maximum IP-ID difference between possibly injected RST packets and the preceding packets in the same connection for up to 1,000 connections per signature sampled on January 17, 2023. In the `Not Tampering` bucket, more than 95% of all connections have a maximum IP-ID difference less than or equal to one. In contrast, the distribution for most of our tampering signatures is significantly different. Except for three signatures (⟨SYN → RST+ACK⟩, ⟨SYN; ACK → RST+ACK⟩, and ⟨PSH+ACK; Data → RST+ACK⟩), 40%–100% of connections matching all the other tampering signatures have a maximum IP-ID difference that is greater than one. This shows that RST packets in these signatures are likely generated by a different TCP stack belonging to the injector.

A high maximum IP-ID difference indicates a positive injection outcome; conversely its absence does not indicate an *absence* of tampering. For instance, it is possible that both the client and the injector TCP stacks set the IP-ID to zero. Moreover, certain censorship systems are known to copy the IP-ID value from the IP header of the client packets [75]. Thus, we use this experiment only to show that a large portion of connections matching our signatures are positive cases of tampering.

Inconsistent changes in Time-to-live (TTL) field Most clients set their initial TTL to a constant value, commonly 64 or 128 [27]. However, when injecting RST packets, some tampering systems may initialize their TTLs with different values, thus allowing us to detect third-party injection. Similar to our investigation of IP-IDs, we first confirm that the *minimum* difference in TTL (or Hop Limit) values between packets in the same connection is less than or equal to one for more than 95% of connections.

As shown in Figure 3, most (>99%) sampled connections with no signature matches do not exhibit a large *maximum* TTL difference. In contrast, many tampering signatures, particularly those that inject RSTs after the first data packet (⟨PSH+ACK → RST; RST_0⟩, ⟨PSH+ACK → RST; RST+ACK⟩, and ⟨PSH+ACK → RST⟩), show larger maximum differences in TTL values. We notice two distinct patterns: CDF curves for tampering signatures such as ⟨SYN; ACK → RST; RST⟩ and ⟨PSH+ACK → RST⟩ show a step-like pattern, indicating different tampering systems with different initial TTL values. Others (e.g., ⟨PSH+ACK → RST ≠ RST⟩) surprisingly show a linear increase in TTL differences between injected and client packets. Upon further inspection, we see this behavior from a number of connections originating from certain countries like South Korea, where RST packets appear with seemingly random TTL values.

Collectively, these observations of supporting evidence demonstrate that our tampering signatures are highly indicative of actual connection tampering by network intermediaries.

5 GLOBAL ANALYSIS OF CONNECTION TAMPERING

We gathered our samples over a period of two weeks (January 12–January 26, 2023) to analyze connections that match our signatures. During this period, the CDN received traffic from 247 countries[5], which allows us to identify patterns of traffic tampering globally.

5.1 Global Signature Matches

We begin by demonstrating that *our passive detection methods can quantify traffic tampering across countries and autonomous systems.* Recall that our tool is only able to directly infer who is affected, and not necessarily who is actually *performing* the tampering (§3.4).

Figure 4 shows the percentage from select countries (geolocated based on source IP address) of total number of connections matching all of our signatures. We have included countries where connection tampering is a known censorship tool [37, 65], as well as six other countries of interest for comparison: the US, GB, DE, UA, PE, and MX. Collectively, this figure shows that *our results confirm prior findings while also providing insight into understudied regions.*

Confirming prior findings A large portion (84%) of connections from Turkmenistan (TM) matches one of our tampering signatures, with 66.4% of the tampered connections matching the ⟨SYN; ACK → RST⟩ signature. Previous work [73] has noted that ISPs in Turkmenistan use blanket bans on CDNs, so such high levels of blocking is expected. Neighboring countries such as Uzbekistan (UZ, 22.9% of connections matching the ⟨SYN; ACK → RST+ACK⟩ signature), Kazakhstan (KZ, 16.5% matching the ⟨SYN; ACK → RST+ACK⟩ signature), Ukraine (UA, 19.2% matching the ⟨PSH+ACK; Data → RST+ACK⟩ signature), and Russia (RU) all observe large amounts of tampered connections. Russia is a particularly interesting case, as we observe many different matching signatures, indicating that different networks perform tampering differently, as has been noted in previous work [66, 67]. We also observe a large percentage of tampered connections originating from Cuba (CU), Saudi Arabia (SA), and Iran (IR), all of which have all of which have been observed to perform country-level censorship [10, 12, 14, 16, 73].

China is known to have an extensive system for traffic tampering that blocks access to a large variety of content [19, 22, 48, 82, 87, 90], and the signatures we observe predominantly in connections from

[5] According to ISO 3166 codes for countries and their subdivisions [45].

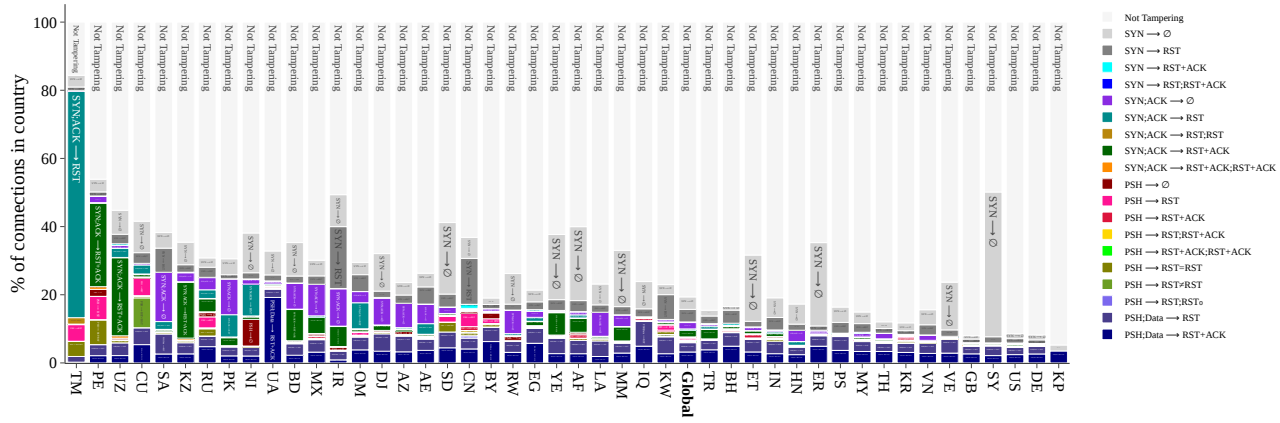


Figure 4: *Signature distribution per country*: The percentage of connections originating from select countries (and globally) that match a particular signature, or are not tampered with.

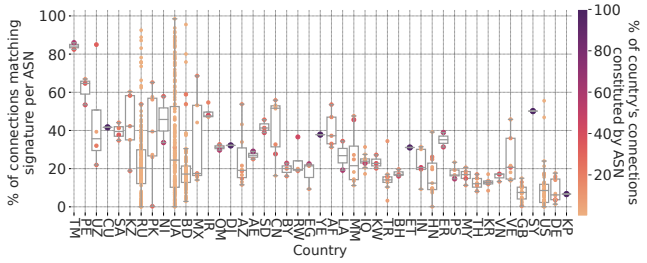


Figure 5: *Percentage of connections that match any signature in each large AS (originating the top 80% of connections in a country) in select countries*: Each dot represents an ASN and the color reflects percentage of country's traffic originating from that ASN. Entries on the x-axis mirror Figure 4.

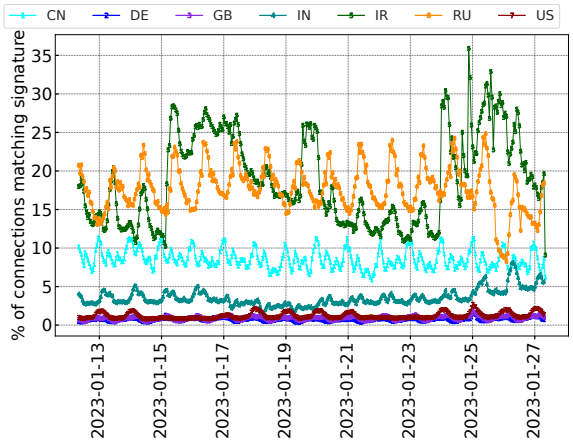


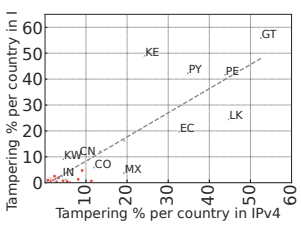
Figure 6: *Signature matches over time*: The percentage of connections matching our `Post-ACK` and `Post-PSH` signatures in some countries of interest. (x-axis denotes local time.)

China in Figure 1 and Figure 4 (⟨`PSH+ACK` → `RST+ACK`; `RST+ACK`⟩, ⟨`PSH+ACK` → `RST`; $\texttt{RST}_0$⟩, ⟨`PSH+ACK` → `RST`; `RST+ACK`⟩, ⟨`PSH+ACK` → `RST`⟩) align closely to known patterns of censorship from the GFW [18, 35, 48, 82]. ISPs in Iran are known to either drop TLS Client Hello packets silently or inject `RST+ACK`s after drops [12], matching our ⟨`SYN`; `ACK` → ∅⟩, ⟨`SYN`; `ACK` → `RST+ACK`⟩, and ⟨`SYN`; `ACK` → `RST+ACK`; `RST+ACK`⟩ signatures. Similarly, our signature matches in India (IN), Pakistan (PK), Russia (RU) all match patterns observed in previous active measurement work [12, 52, 66, 89].

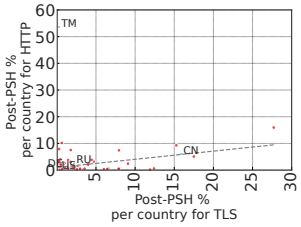
Insight into understudied regions One of the advantages of an entirely passive system is its ability to scale globally, thereby allowing us to measure regions that have not been studied by in-depth active measurements. For instance, a large percentage (33.9%) of the ⟨`PSH+ACK` → `RST` ≠ `RST`⟩ signature matches on connections from South Korea (KR) are dominated by a single ISP, from which the `RST` packets appear to have randomized TTL values (see §4). This level of observability helps operators to understand connection failures in customer traffic. Peru (PE) and Mexico (MX) have also not been widely studied, yet still originate a high percentage of tampered connections (53.9% and 30.1% tampered connections, respectively). Figure 4 also shows Germany (DE), the United Kingdom (GB), and the United States (US), all places where connection tampering is in use or under consideration for copyright and other protections [57, 58, 60].

An AS-centric view of tampering An AS view of the data is presented in Figure 5, with x-axis mirroring the content and ordering in Figure 4. Recall that we can only infer the AS from which the connection was initiated; tampering can occur anywhere along the path from the source AS to the destination. For each source AS we calculate the "match-proportion" as the percentage of connections from that AS that match any tampering signature. Figure 5 shows the distribution of match-proportions for the top ASes in a country collectively constituting 80% of connections from that country.

We observe that *tampering across ASes with centralized tampering systems tends to vary less than those with more decentralized systems.* Countries known to have centralized and coordinated censorship systems, such as China (CN) and Iran (IR) [7, 10], are largely homogeneous across ASes with respect to tampering, as evidenced by short y-axis range, regardless of the size of the ASes. Countries like Russia (RU), Ukraine (UK), and Pakistan (PK) that have been shown to have a decentralized system of information control [52, 66], show



(a) Percentage of `Post-ACK` and `Post-PSH` matches in IPv4 and IPv6 (regression slope = 0.92).



(b) Percentage of `Post-PSH` matches for TLS vs HTTP requests (regression slope = 0.3).

Figure 7: IPv4 vs IPv6 & TLS vs HTTP Tampering.

more heterogeneity in tampering as evidenced by long y-axis ranges. We also show that blocking varies across ASes in countries such as Mexico (MX) that active measurements have not thoroughly studied. Although proportions of tampering in the US, UK, and DE are overall smaller by comparison, the individual proportions vary greatly across ASes.

5.2 Longitudinal Signature Matches

One of the benefits of our approach is that *passive techniques follow shifts in user behaviors over time*. Figure 6 shows the percentage of connections matching our `Post-ACK` and `Post-PSH` signatures in six countries: CN, DE, GB, IN, IR, RU, and US. These signatures are typically triggered by the TLS Client Hello or HTTP request.

We observe that the tampering connections in each country exhibit a diurnal pattern, with spikes typically between midnight and 8AM local time. These findings indicate that tampered connections form a larger percentage of the traffic from these countries during the late night and early morning hours. We also observe that tampering percentages are generally lower during the weekends, particularly in Iran. The high variability of signature matches over time in Iran is, we believe, due to reactions to ongoing protests there; we explore this in more depth in §5.6.

5.3 IP Version and Destination Port

Our tampering signatures operate strictly over TCP headers; thus, *we are able to measure connection tampering over both IPv4 and IPv6*. This feature overcomes a limitation of active measurements focused on the IPv4 protocol [62, 73], in part because IPv6 clients are less common and harder to solicit (§2). Figure 7(a) shows the percentage of `Post-ACK` and `Post-PSH` signature matches in IPv4 vs IPv6 for all countries. Taken in aggregate, there appears to be no significant differences between tampering on IPv4 vs IPv6 traffic (the regression line has slope 0.92), but there are some disparities in certain countries. For example, the tampering rate from Sri Lanka (LK) is more than 40% in IPv4 connections and less than 25% in IPv6; conversely, tampering rates from Kenya in IPv6 are almost double the 25% rate in IPv4.

Our data also allows us to compare how frequently tampering is applied to TLS handshakes versus HTTP requests. Figure 7(b) shows the portion of `Post-PSH` signatures (when the domain or GET is observed) for HTTP and TLS requests in each region. Overall, *TLS handshakes appear to be more prone to tampering than HTTP in most places*. For instance, among connections from China (CN), around 15% of TLS handshakes match our tampering signature, compared to 7% of HTTP requests. Connections from Turkmenistan (TM) stand out as an exception, where over 50% of HTTP requests match our tampering signature, but virtually no TLS handshakes.

Source Region	Most Affected Categories	% of all Tampered Connections	% of all Domains seen in Category
Global	Adult Themes	13.25	10.77
	Content Servers	8.47	1.13
	Technology	6.46	0.91
CN	Adult Themes	17.96	50.99
	Content Servers	4.92	3.09
	Education	3.40	21.28
DE	Content Servers	12.4	0.48
	Business	10.51	0.36
	Technology	7.57	0.2
GB	Content Servers	16.53	0.41
	Business	7.93	0.21
	Technology	7.10	0.18
IN	Adult Themes	40.31	18.33
	Chat	13.23	3.4
	Content Servers	12.57	2.37
IR	Content Servers	27.84	30.23
	Technology	26.05	2.17
	Business	3.66	1.42
KR	Adult Themes	39.46	37.58
	Gaming	7.14	1.53
	Login Screens	5.58	30.48
MX	Advertisements	23.26	12.62
	Technology	17.75	3.42
	Business	15.83	2.86
PE	Advertisements	22.06	61.5
	Business	3.03	5.91
	Technology	3.00	8.52
RU	Hobbies & Interests	18.97	28.08
	Business	10.27	2.91
	Advertisements	8.49	7.40
US	Content Servers	15.55	0.59
	Technology	11.09	0.35
	Business	10.20	0.29

Table 2: A view of `Post-PSH` connection tampering as it affects clients and categories. The second column shows the top-3 domain categories most affected. The third column is the category's proportion of all tampered *connections* from the corresponding region. The last column is coverage, specifically the proportion of all domains in the category affected by tampering. For example, globally, `Post-PSH` tampering affects 1.13% of all domains in the Content Servers category, but is responsible for 8.47% of all `Post-PSH` tampering.

5.4 Tampering and Users' Experience

Next, we investigate how tampering affects users' experience by investigating what kinds of content is actively being tampered with. We first bucket domains into subject categories. We categorize[6] the

[6]Given the difficulty involved with domain categorization, some domains may fit inside multiple categories.

List Name	# Entries	Global	CN	IN	IR	KR	MX	PE	RU	US
Tranco_1K	1,000	4.7%	1.7%	8.3%	0.0%	33.3%	9.1%	0.0%	20.8%	17.0%
Tranco_10K	10,000	20.1%	6.8%	39.2%	12.5%	44.4%	45.5%	23.5%	50.0%	69.5%
Tranco_100K	100,000	47.0%	16.7%	76.3%	31.3%	55.6%	81.8%	52.9%	87.5%	86.44%
Tranco_1M	1,000,000	69.8%	45.4%	97.9%	43.8%	72.2%	100.0%	94.1%	95.8%	93.22%
Majestic_1K	1,000	2.5%	1.0%	4.1%	0.0%	11.1%	0.0%	0.0%	4.2%	10.17%
Majestic_10K	10,000	7.5%	2.4%	9.3%	0.0%	16.7%	9.1%	0.0%	20.8%	28.8%
Majestic_100K	100,000	15.3%	5.1%	20.6%	0.0%	33.3%	18.2%	11.8%	33.3%	45.8%
Majestic_1M	1,000,000	31.2%	13.0%	66.0%	12.5%	33.3%	36.4%	23.5%	54.2%	62.7%
Greatfire_all	214,406	22.7%	10.9%	43.3%	6.3%	44.4%	27.3%	5.9%	50.0%	54.2%
Greatfire_30d	22,427	10.1%	5.5%	22.7%	0.0%	27.8%	9.1%	0.0%	20.8%	11.9%
Citizenlab	23399	7.5%	3.1%	12.4%	0.0%	22.2%	18.2%	0.0%	25.0%	11.9%
Citizenlab_global	1,388	3.6%	1.7%	5.2%	0.0%	11.1%	9.1%	0.0%	12.5%	10.2%
Citizenlab_country	Variable	-	0.7%	2.1%	0.0%	0.0%	0.0%	0.0%	8.3%	0.0%
Union: Citizenlab + Greatfire	233,359	23.1%	10.9%	43.3%	6.3%	44.4%	27.3%	5.9%	50.0%	54.2%
Union: All lists	1,627,447	71.5%	48.1%	99.0%	43.8%	72.2%	100.0%	94.12%	95.8%	93.2%
Substring: Citizenlab + Greatfire	-	53.6%	36.9%	62.9%	56.3%	61.1%	54.5%	35.3%	58.3%	71.2%
Substring: All lists	-	87.7%	77.5%	100.0%	93.8%	83.3%	100.0%	100.0%	95.8%	96.6%

Table 3: `Post-PSH` *domains in published lists* when comparing eTLD+1 regionally. Each cell in row x column y denotes the percentage of all domains observed to be tampered in region y that would have been captured by active scanning using test list x. Rows marked 'Substring' indicate percentages if domains in tampered connections are treated as substrings in lists.

domain name of each connection matching any `Post-PSH` signature using the CDN's third-party vendor stream [50]. For confidence, we count a domain as being tampered within a region only if it exceeds 100 `Post-PSH` matches in a one-day period. Other threshold values did not significantly change the results summarized below.

Table 2 shows the top three categories of domains in `Post-PSH` matches for a selection of regions. *The three categories most subject to tampering are "Adult Themes," "Content Servers," and "Technology".* Globally, the top-three most affected categories are more than 28% of all `Post-PSH` signature matches. Much of this traffic originates from China and India, where prior active measurements have noted that domains hosting Adult Theme content are usually censored [54, 73]. Content Servers include CDNs and sites with content frequently retrieved by other applications; Technology is a broad category of product and services related sites.

Table 2 also includes (in the third column) the *coverage* of the category within the region: that is, the proportion of tampered connections in that category over all tampered connections from that region. As an example comparison, consider connections from China and India where Adult Themes are known to be censored [54, 73]. Adult Themes content accounts for almost 18% of all tampered connections from China, and more than 40% of all tampered connections initiated in India.

Finally, Table 2's fourth column shows how *complete* the tampering of a given category is: that is, the proportion of tampered domains in a category over all domains in the category observed to be accessed from a given region. We posit that greater completeness indicates a concerted effort to block a given category of websites. Continuing the above examples, ~51% of requests from China for Adult Themes content are tampered with, and ~18% for India.

The separations between domain categories and connections are particularly stark in DE, GB, and US, where connection tampering is not as widespread. For instance, the top three tampered categories in the US have only 0.41% average coverage but account for almost 37% of all tampered connections, meaning that the majority of tampering is associated with a relatively small number of domains within each category.

Collectively, these results demonstrate one of the most powerful features of applying passive measurements to measuring connection tampering: *because passive measurements are driven by real user behavior, they allow us to not only identify blocked content, but also the effect of that blocking on users' experience.*

5.5 Comparison with Test Lists

We compare the domains matching our `Post-PSH` signatures to active measurement test lists containing popular domains such as Tranco [47] and Majestic [49] and test lists curated for censorship measurements such as GreatFire [38] and Citizen Lab [23]. Table 3 shows the coverage of various active measurement test lists over various regions (using a threshold of 100 connections per domain).

Our system can identify tampered domains that are not included in active measurement test lists. We find that test lists specially curated for censorship measurements do not contain a large percentage of the domains for which we observe `Post-PSH` signatures. For instance, only 10.9% of domains that observe `Post-PSH` signatures matches in CN are contained within the union of the Citizen Lab and GreatFire lists (fourth-to-last row in Table 3). Domain lists based on popularity such as Tranco and Majestic contain more tampered domains due to their larger size and applicability across a wider range of countries. Recall that, while our data can help characterize tampered domains that test lists miss, passive measurements have some limitations compared to active measurements. In particular, our dataset is limited to the domains that are both proxied by the CDN and actively requested by users; active measurements can of course initiate queries for any domain. Moreover, our view into which domains are tampered with is limited in regions whose `PSH` packets containing the TLS Client Hello or HTTP GET request are dropped, such as by the censorship policy in Iran [10, 12].

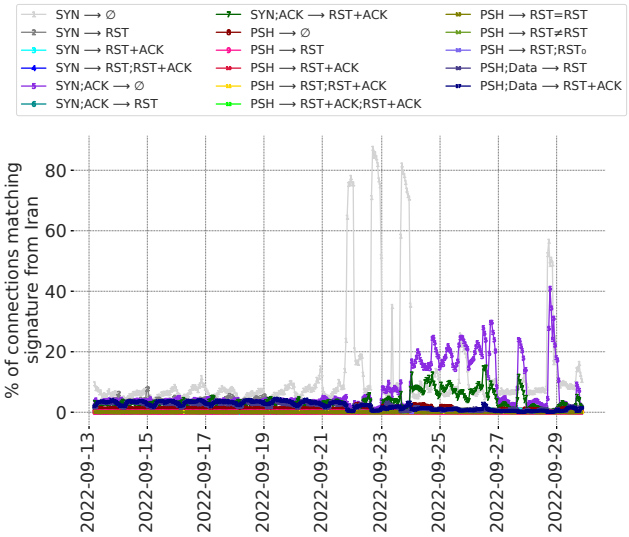


Figure 8: *Signature match rates longitudinally in Iran* during a period of nation-wide protests. (*x*-axis is local time.)

Several prior studies have observed *over-blocking* of domain names, potentially caused by erroneous regular expression [2, 42, 56]. For instance, Nourin et al. [56] observed that Turkmenistan censors *any* domain that includes the substring `wn.com`, potentially over-blocking hundreds of thousands of domains. By looking for exact matches between the domains we observe and those in test lists, we are potentially under-reporting on how well the test lists actually perform. To account for this, the last two rows of Table 3 provide a best-case scenario for test lists; in these rows, we report on the percentage of tampered domains that are a substring of any domain in the test list. Even in this best-case scenario, the curated test lists do not capture all of the domains our passive measurements detect. These results collectively show that passive techniques can be valuable in informing the construction of test lists.

5.6 Case Study: Tampering in Iran

Longitudinal passive measurement can provide insights into tampering around noteworthy events. For example, large ongoing protests sparked on September 13, 2022, in Iran led to aggressive Internet blocking practices [59, 76]. A 17-day timeseries of various hourly signature matches on connections coming from Iran is plotted in Figure 8. During this interval, signature match rates increase significantly, particularly for the ⟨`SYN` ⟶ `RST`⟩, ⟨`SYN`; `ACK` ⟶ `∅`⟩ and ⟨`SYN`; `ACK` ⟶ `RST+ACK`⟩ signatures; in certain instances, more than 40% of all connections exhibited timeouts after the handshake. This behavior is consistent with previous work showing Iran's firewall systems drop TLS Client Hello packets, forcing a timeout [12, 73]. Furthermore, the vast majority of connections and signature matches during this period appeared on connections from two mobile ISPs. The tampering appears to peak in the (late) evening hours.

6 CONCLUDING REMARKS

In this paper, we presented the first global passive study on connection tampering, building upon previous insights that censors' actions can be fingerprinted and detected passively and efficiently [84]. This novel deployment led to myriad insights that are unavailable with active measurements alone, including analyses of the impact of tampering on users' experience, not just the set of domain names that *could* be tampered with.

Are tampering signatures stable? Past evidence shows that tampering signatures are stable, indicating that drastic changes are unlikely in the near future [74]. For instance, the GFW was first observed sending a pattern of three `RST` packets in 2006 [24], and this pattern has been measured in many independent projects since [18, 82].

One possible reason that censors are slow in updating such behavior is that their censorship infrastructure is made up of multiple middleboxes [17, 75], often from third-party vendors. Thus, altering their censorship mechanisms often requires deploying new physical middleboxes, which is costly and time-consuming. Moreover, there are finite ways to tamper with connections, as evidenced by our ability to enumerate them. Indeed, researchers often associate new censorship fingerprints directly with the deployment of new middleboxes themselves [19].

Nonetheless, it is useful to ask: How *could* a censor evade passive detection? First, they would have to stop using uncommon sequences of packets, like the GFW's multiple `RST`s. The ideal tampering strategy would involve *blocking* content from the server to the client (so the client does not get any objectionable content), while *continuing* the connection to the server as if it were the client (so the server does not detect any immediate connection teardowns). Such a strategy would only be possible when the tampering middlebox can drop packets, which is uncommon in practice (§2).

Is this the end of active measurements? *No!* Passive and active approaches complement one another. This work shows many instances where passive techniques could gain insights that are unavailable to active techniques, but do not diminish active measurements' unique strengths. For instance, our passive system limits our measurements to the domains using the CDN's services; active measurements do not have that limitation. Active measurement can also trigger events and test hypotheses to understand tampering in ways that passive measurement cannot. These explanations are crucial not just to clients and websites, but also for the service operators that connect them. We hope that our study encourages the combined use of active and passive measurements for obtaining a holistic view of connection tampering.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their helpful feedback. We thank the many people at Cloudflare who helped inform and shape this work including but not limited to Syed Suleman Ahmad, David Belson, Kristin Berdan, Alex Forster, John Graham-Cumming, Arian Akhavan Niaki, Sudheesh Singanamalla, Alissa Starzak, Avani Wildani, and Peter Wu. Also, thanks to Cloudflare's Radar team for their engineering and design to support on-going data sharing. This research was supported in part by NSF grant CNS-1943240.

REFERENCES

[1] Transmission Control Protocol. RFC 793, RFC Editor, September 1981.

[2] Access Now. The return of digital authoritarianism: Internet shutdowns in 2021. https://www.accessnow.org/wp-content/uploads/2022/05/2021-KIO-Report-May-24-2022.pdf, april 2022.

[3] Maria Agrabeli. Internet Censorship in Iran: Findings from 2014-2017. https://blog.torproject.org/internet-censorship-iran-findings-2014-2017, 2017.

[4] Collin Anderson, Philipp Winter, and Roya. Global Network Interference Detection over the RIPE Atlas Network. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2014.

[5] Daniel Anderson. Splinternet Behind the Great Firewall of China. *ACM Queue*, 10(11), 2012.

[6] Anonymous. The Collateral Damage of Internet Censorship. *ACM SIGCOMM Computer Communication Review (CCR)*, 42(3):21–27, 2012.

[7] Anonymous. Towards a Comprehensive Picture of the Great Firewall's DNS Censorship. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2014.

[8] Anonymous, Arian Akhavan Niaki, Nguyen Phone Hoang, Phillipa Gill, and Amir Houmansadr. Triplet Censors: Demystifying Great Firewall's DNS Censorship Behavior. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2020.

[9] Martin Arlitt and Carey Williamson. An Analysis of TCP Reset Behaviour on the Internet. *ACM SIGCOMM Computer Communication Review (CCR)*, 35(1):37–44, 2005.

[10] Simurgh Aryan, Homa Aryan, and J. Alex Halderman. Internet Censorship in Iran: A First Look. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2013.

[11] Vaibhav Bajpai and Jürgen Schönwälder. Measuring the Effects of Happy Eyeballs. In *Applied Networking Research Workshop*, 2016.

[12] Simone Basso. Measuring SNI based blocking in Iran. https://ooni.org/post/2020-iran-sni-blocking/, 2020.

[13] Berkman Klein Test Lists. https://github.com/berkmancenter/url-lists.

[14] Kevin Bock, Abdulrahman Alaraj, Yair Fax, Kyle Hurley, Eric Wustrow, and Dave Levin. Weaponizing Middleboxes for TCP Reflected Amplification. In *USENIX Security Symposium*, 2021.

[15] Kevin Bock, Pranav Bharadwaj, Jasraj Singh, and Dave Levin. Your Censor is My Censor: Weaponizing Censorship Infrastructure for Availability Attacks. In *USENIX Workshop on Offensive Technologies (WOOT)*, 2021.

[16] Kevin Bock, Yair Fax, Kyle Reese, Jasraj Singh, and Dave Levin. Detecting and Evading Censorship-in-Depth: A Case Study of Iran's Protocol Whitelister. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2020.

[17] Kevin Bock, George Hughey, Louis-Henri Merino, Tania Arya, Daniel Liscinsky, Regina Pogosian, and Dave Levin. Come as You Are: Helping Unmodified Clients Bypass Censorship with Server-side Evasion. In *ACM SIGCOMM*, 2020.

[18] Kevin Bock, George Hughey, Xiao Qiang, and Dave Levin. Geneva: Evolving Censorship Evasion. In *ACM Conference on Computer and Communications Security (CCS)*, 2019.

[19] Kevin Bock, iyouport, Anonymous, Louis-Henri Merino, David Fifield, Amir Houmansadr, and Dave Levin. Exposing and Circumventing China's Censorship of ESNI. https://geneva.cs.umd.edu/posts/china-censors-esni/esni/, 2020.

[20] Kevin Bock, Gabriel Naval, Kyle Reese, and Dave Levin. Even Censors Have a Backup: Examining China's Double HTTPS Censorship Middleboxes. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2021.

[21] Sam Burnett and Nick Feamster. Encore: Lightweight Measurement of Web Censorship with Cross-Origin Requests. In *ACM SIGCOMM*, 2015.

[22] Zimo Chai, Amirhossein Ghafari, and Amir Houmansadr. On the Importance of Encrypted-SNI (ESNI) to Censorship Circumvention. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2019.

[23] CitizenLab Censorship Test Lists. https://github.com/citizenlab/test-lists.

[24] Richard Clayton, Steven J. Murdoch, and Robert N. M. Watson. Ignoring the Great Firewall of China. In *Privacy Enhancing Technologies Symposium (PETS)*, 2006.

[25] Alexander Darer, Oliver Farnan, and Joss Wright. FilteredWeb: A Framework for the Automated Search-Based Discovery of Blocked URLs. In *Network Traffic Measurement and Analysis Conference (TMA)*, 2017.

[26] Alexander Darer, Oliver Farnan, and Joss Wright. Automated Discovery of Internet Censorship by Web Crawling. In *WebSci*, 2018.

[27] Default TTL (Time To Live) Values of Different OS. https://subinsb.com/default-device-ttl-values/, 2014.

[28] D. Dittrich and E. Kenneally. The Menlo Report: Ethical Principles Guiding Information and Communication Technology Research. Technical report, U.S. Department of Homeland Security, Aug 2012.

[29] Arun Dunna, Ciarán O'Brien, and Phillipa Gill. Analyzing China's Blocking of Unpublished Tor Bridges. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2018.

[30] Zakir Durumeric, Eric Wustrow, and J Alex Halderman. ZMap: Fast Internet-wide Scanning and Its Security Applications. In *USENIX Security Symposium*, 2013.

[31] Roya Ensafi, David Fifield, Philipp Winter, Nick Feamster, Nicholas Weaver, and Vern Paxson. Examining How the Great Firewall Discovers Hidden Circumvention Servers. In *ACM Internet Measurement Conference (IMC)*, 2015.

[32] Roya Ensafi, Jeffrey Knockel, Geoffrey Alexander, and Jedidiah R. Crandall. Detecting Intentional Packet Drops on the Internet via TCP/IP Side Channels. In *Passive and Active Network Measurement Conference (PAM)*, 2014.

[33] Oliver Farnan, Alexander Darer, and Joss Wright. Analysing Censorship Circumvention with VPNs via DNS Cache Snooping. In *IEEE International Workshop on Traffic Measurements for Cybersecurity*, 2019.

[34] David Fifield and Lynn Tsai. Censors' Delay in Blocking Circumvention Proxies. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2016.

[35] Arturo Filasto and Jacob Appelbaum. OONI: Open Observatory of Network Interference. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2012.

[36] fqrouter. Detailed GFW's three blocking methods for SMTP protocol. https://web.archive.org/web/20151121091522/http://fqrouter.tumblr.com/post/43400982633/%E8%AF%A6%E8%BF%B0gfw%E5%AF%B9smtp%E5%8D%8F%E8%AE%AE%E7%9A%84%E4%B8%89%E7%A7%8D%E5%B0%81%E9%94%81%E6%89%8B%E6%B3%95, 2015.

[37] Freedom on the Net. https://freedomhouse.org/report/freedom-net.

[38] GreatFire. Censorship of Blocked in China | GreatFire Analyzer. https://en.greatfire.org/search/blocked, 2019.

[39] Herdict Test Lists. https://cyber.harvard.edu/research/herdict, 2017.

[40] Raphael Hiesgen, Marcin Nawrocki, Alistair King, Alberto Dainotti, Thomas C Schmidt, and Matthias Wählisch. Spoki: Unveiling a New Wave of Scanners through a Reactive Network Telescope. *USENIX Security Symposium*, 2022.

[41] Nguyen Phong Hoang, Sadie Doreen, and Michalis Polychronakis. Measuring I2P Censorship at a Global Scale. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2019.

[42] Nguyen Phong Hoang, Arian Akhavan Niaki, Jakub Dalek, Jeffrey Knockel, Pellaeon Lin, Bill Marczak, Masashi Crete-Nishihata, Phillipa Gill, and Michalis Polychronakis. How Great is the Great Firewall? Measuring China's DNS Censorship. In *USENIX Security Symposium*, 2021.

[43] Austin Hounsel, Prateek Mittal, and Nick Feamster. Automatically Generating a Large, Culture-Specific Blocklist for China. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2018.

[44] IODA: Internet Outage Detection and Analysis. https://ioda.caida.org/, 2023.

[45] The International Standard for country codes and codes for their subdivisions. Standard, International Organization for Standardization, Geneva, CH.

[46] Ben Jones, Roya Ensafi, Nick Feamster, Vern Paxson, and Nick Weaver. Ethical Concerns for Censorship Measurement. In *NS Ethics*, 2015.

[47] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. In *Network and Distributed System Security Symposium (NDSS)*, 2019.

[48] Fangfan Li, Abbas Razaghpanah, Arash Molavi Kakhki, Arian Akhavan Niaki, David Choffnes, Phillipa Gill, and Alan Mislove. lib.erate, (n): A library for exposing (traffic-classification) rules and avoiding them efficiently. In *ACM Internet Measurement Conference (IMC)*, 2017.

[49] Majestic Million: The million domains we find with the most referring subnets. https://majestic.com/, 2021.

[50] Matthew Prince. The Mistake that Caused 1.1.1.3 to Block LGBTQIA+ Sites Today. https://blog.cloudflare.com/the-mistake-that-caused-1-1-1-3-to-block-lgbtqia-sites-today, april 2020.

[51] Allison McDonald, Matthew Bernhard, Luke Valenta, Benjamin VanderSloot, Will Scott, Nick Sullivan, J. Alex Halderman, and Roya Ensafi. 403 Forbidden: A Global View of CDN Geoblocking. In *ACM Internet Measurement Conference (IMC)*, 2018.

[52] Zubair Nabi. The Anatomy of Web Censorship in Pakistan. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2013.

[53] NetBlocks. https://netblocks.org/, 2023.

[54] Arian Akhavan Niaki, Shinyoung Cho, Zachary Weinberg, Nguyen Phong Hoang, Abbas Razaghpanah, Nicolas Christin, and Phillipa Gill. ICLab: A Global, Longitudinal Internet Censorship Measurement Platform. In *IEEE Symposium on Security and Privacy*, 2020.

[55] Nmap. TCP Idle Scan. https://nmap.org/book/idlescan.html, 2023.

[56] Sadia Nourin, Van Tran, Xi Jiang, Kevin Bock, Nick Feamster, Nguyen Phong Hoang, and Dave Levin. Measuring and Evading Turkmenistan's Internet Censorship. In *International World Wide Web Conference (WWW)*, 2023.

[57] European Union Intellectual Property Office. Live event piracy: discussion paper: challenges and good practices from online intermediaries to prevent the use of

their services for live event piracy. 2023.
[58] European Union Intellectual Property Office, Giancarlo Frosio, and Oleksandr Bulayenko. Study on dynamic blocking injunctions in the European Union. 2021.
[59] Technical multi-stakeholder report on Internet shutdowns: The case of Iran amid autumn 2022 protests. `https://rsf.org/en/index`, 2022.
[60] Open Rights Group. Blocked! `https://www.blocked.org.uk/about`.
[61] Craig Partridge and Mark Allman. Ethical Considerations in Network Measurement Papers. *Communications of the ACM*, 59(10):58–64, sep 2016.
[62] Paul Pearce, Roya Ensafi, Frank Li, Nick Feamster, and Vern Paxson. Augur: Internet-Wide Detection of Connectivity Disruptions. In *IEEE Symposium on Security and Privacy*, 2017.
[63] Paul Pearce, Ben Jones, Frank Li, Roya Ensafi, Nick Feamster, Nick Weaver, and Vern Paxson. Global Measurement of DNS Manipulation. In *USENIX Security Symposium*, 2017.
[64] Jon Postel. Internet Protocol. RFC 791, September 1981.
[65] Press Freedom Index. `https://rsf.org/en/index`.
[66] Reethika Ramesh, Ram Sundara Raman, Matthew Bernhard, Victor Ongkowijaya, Leonid Evdokimov, Annie Edmundson, S. Sprecher, Muhammad Ikram, and Roya Ensafi. Decentralized Control: A Case Study of Russia. In *Network and Distributed System Security Symposium (NDSS)*, 2020.
[67] Reethika Ramesh, Ram Sundara Raman, Apurva Virkud, Alexandra Dirksen, Armin Huremagic, David Fifield, Dirk Rodenburg, Rod Hynes, Doug Madory, and Roya Ensafi. Network Responses to Russia's Invasion of Ukraine in 2022: A Cautionary Tale for Internet Freedom. In *USENIX Security Symposium*, 2023.
[68] Reporters Without Borders. Enemies of the Internet 2013 Report. `https://surveillance.rsf.org/en/wp-content/uploads/sites/2/2013/03/enemies-of-the-internet_2013.pdf`, March 2013.
[69] Flavia Salutari, Danilo Cicalese, and Dario J Rossi. A Closer Look at IP-ID Behavior in the Wild. In *Passive and Active Network Measurement Conference (PAM)*, 2018.
[70] David Schinazi and Tommy Pauly. Happy Eyeballs Version 2: Better Connectivity Using Concurrency. RFC 8305, December 2017.
[71] Rachee Singh, Rishab Nithyanand, Sadia Afroz, Paul Pearce, Michael Carl Tschantz, Phillipa Gill, and Vern Paxson. Characterizing the Nature and Dynamics of Tor Exit Blocking. In *USENIX Security Symposium*, 2017.
[72] Ram Sundara Raman, Leonid Evdokimov, Eric Wustrow, Alex Halderman, and Roya Ensafi. Investigating Large Scale HTTPS Interception in Kazakhstan. In *ACM Internet Measurement Conference (IMC)*, 2015.
[73] Ram Sundara Raman, Prerana Shenoy, Katharina Kohls, and Roya Ensafi. Censored Planet: An Internet-wide, Longitudinal Censorship Observatory. In *ACM Conference on Computer and Communications Security (CCS)*, 2020.
[74] Ram Sundara Raman, Adrian Stoll, Jakub Dalek, Reethika Ramesh, Will Scott, and Roya Ensafi. Measuring the Deployment of Network Censorship Filters at Global Scale. In *Network and Distributed System Security Symposium (NDSS)*, 2020.
[75] Ram Sundara Raman, Mona Wang, Jakub Dalek, Jonathan Mayer, and Roya Ensafi. Network Measurement Methods for Locating and Examining Censorship Devices. In *ACM Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, 2022.
[76] Internet Shutdowns and Censorship, in Iran and Beyond. `https://techpolicy.press/internet-shutdowns-and-censorship-in-iran-and-beyond/`, 2022.
[77] Joe Touch. Updated Specification of the IPv4 ID Field. RFC 6864, February 2013.
[78] Benjamin VanderSloot, Allison McDonald, Will Scott, J. Alex Halderman, and Roya Ensafi. Quack: Scalable Remote Measurement of Application-Layer Censorship. In *USENIX Security Symposium*, 2018.
[79] John-Paul Verkamp and Minaxi Gupta. Inferring Mechanics of Web Censorship Around the World. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2012.
[80] W3Techs Web Technology Surveys. Usage statistics and market share of Cloudflare. `https://w3techs.com/technologies/details/cn-cloudflare` last accessed 01/2022.
[81] W3Techs Web Technology Surveys. Usage statistics of Cloudflare Server. `https://w3techs.com/technologies/details/ws-cloudflare` last accessed 01/2022.
[82] Zhongjie Wang, Yue Cao, Zhiyun Qian, Chengyu Song, and Srikanth V. Krishnamurthy. Your State is Not Mine: A Closer Look at Evading Stateful Internet Censorship. In *ACM Internet Measurement Conference (IMC)*, 2017.
[83] Zhongjie Wang, Shitong Zhu, Yue Cao, Zhiyun Qian, Chengyu Song, Srikanth V. Krishnamurthy, Kevin S. Chan, and Tracy D. Braun. SymTCP: Eluding Stateful Deep Packet Inspection with Automated Discrepancy Discovery. In *Network and Distributed System Security Symposium (NDSS)*, 2020.
[84] Nicholas Weaver, Robin Sommer, and Vern Paxson. Detecting Forged TCP Reset Packets. In *Network and Distributed System Security Symposium (NDSS)*, 2009.
[85] Dan Wing and Andrew Yourtchenko. Happy Eyeballs: Success with Dual-Stack Hosts. RFC 6555, April 2012.
[86] Philipp Winter and Stefan Lindskog. How the Great Firewall of China is Blocking Tor. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2012.
[87] Xueyang Xu, Morley Mao, and J. Alex Halderman. Internet Censorship in China: Where Does the Filtering Occur? In *Passive and Active Network Measurement Conference (PAM)*, 2011.
[88] Diwen Xue, Benjamin Mixon-Baca, ValdikSS, Anna Ablove, Beau Kujath, Jedidiah R. Crandall, and Roya Ensafi. TSPU: Russia's Decentralized Censorship System. In *ACM Internet Measurement Conference (IMC)*, 2022.
[89] Tarun Kumar Yadav, Akshat Sinha, Devashish Gosain, Piyush Kumar Sharma, and Sambuddho Chakravarty. Where The Light Gets In: Analyzing Web Censorship Mechanisms in India. In *ACM Internet Measurement Conference (IMC)*, 2018.
[90] Zhi-Jin Zhong, Tongchen Wang, and Minting Huang. Does the Great Fire Wall Cause Self-Censorship? The Effects of Perceived Internet Regulation and the Justification of Regulation. *Internet Research*, 27(4):974–990, 2017.

APPENDIX

In these appendices, we present two additional results: the aggregate rate of signature matches over time, and the (low) variability of tamper signatures applied to any given IP-domain pair. We note that appendices are supporting material that has not been peer-reviewed.

A LONGITUDINAL SIGNATURE MATCHES

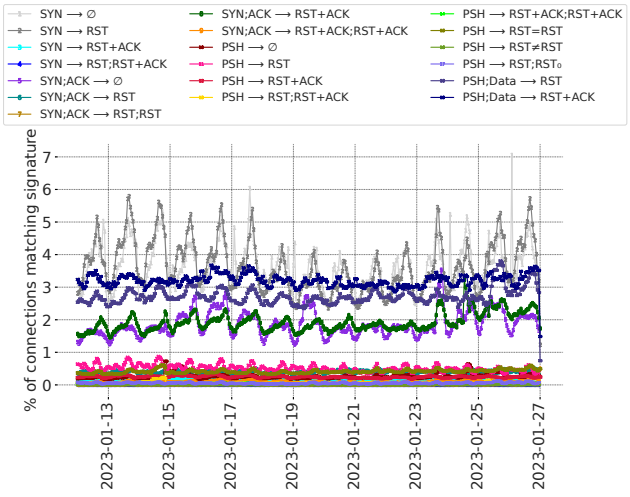


Figure 9: *Signature matches over time*: The percentage of connections matching each signature.

Figure 9 shows the percentage of connections that match each of our tampering signatures over the two week period that we consider. Many tampering match rates follow a diurnal pattern, especially those (such as ⟨PSH+ACK → RST⟩ and ⟨SYN → RST⟩) that are more prevalent in connections that originate from a few countries. Signatures that match on connections originating from many countries (such as ⟨PSH+ACK; Data → RST⟩ and ⟨PSH+ACK; Data → RST+ACK⟩) show lesser diurnal variance.

B SIGNATURE OVERLAPS

Signature matching is largely consistent, with certain exceptions. Consistent tampering between a source IP address and a domain is an indication that the domain caused the tampering. To evaluate consistency, we focus on Post-PSH matches between IP-domain pairs; since the domain is visible in the data, it was also observable

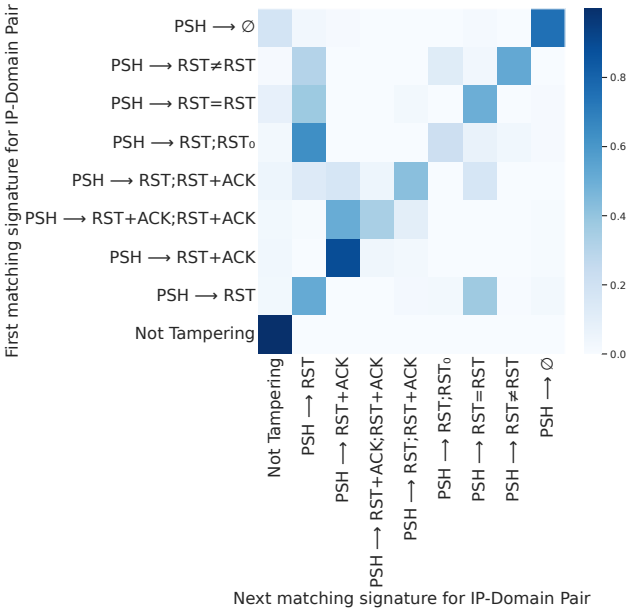


Figure 10: *Signatures that occur for the same IP and domain pair*: Each cell represents the fraction of connections from the same IP address for the same domain that matches the signature on the y-axis first and then the signature on the x-axis. Most IP-Domain connections experience the same tampering signature both times.

on the path to the middlebox. Results are shown in Figure 10, which shows the fraction of connections with the same IP-domain pairs that first match the signature on the Y-axis, and then match the signature on the X-axis. The high values in the diagonal show that signature matches are largely consistent. However, we observe that many of the signatures that match two or more `RST` packets (such as ⟨`PSH+ACK` ⟶ `RST` = `RST`⟩) later match the signature that matches one `RST` packet (⟨`PSH+ACK` ⟶ `RST`⟩), and vice-versa. There are a many reasons why this may happen: (1) Certain tampering systems are known to inject multiple packets with `RST` flags to ensure connection termination [20], (2) This could represent a form of residual blocking, where the tampering entity starts behaving differently for a limited time once triggered, and (3) This could represent cases where one or more `RST` packets are lost in the network. In summary, we do not observe a significant difference in tampering patterns by observing a single `RST` packet versus multiple `RST` packets.