Measuring and Evading Turkmenistan's Internet Censorship

A Case Study in Large-Scale Measurements of a Low-Penetration Country

Sadia Nourin
University of Maryland

Van Tran
University of Chicago

Xi Jiang
University of Chicago

Kevin Bock
University of Maryland

Nick Feamster
University of Chicago

Nguyen Phong Hoang
University of Chicago

Dave Levin
University of Maryland

ABSTRACT

Since 2006, Turkmenistan has been listed as one of the few Internet enemies by Reporters without Borders due to its extensively censored Internet and strictly regulated information control policies. Existing reports of filtering in Turkmenistan rely on a handful of vantage points or test a small number of websites. Yet, the country's poor Internet adoption rates and small population can make more comprehensive measurement challenging. With a population of only six million people and an Internet penetration rate of only 38%, it is challenging to either recruit in-country volunteers or obtain vantage points to conduct remote network measurements at scale.

We present the largest measurement study to date of Turkmenistan's Web censorship. To do so, we developed **TMC**, which tests the blocking status of millions of domains across the three foundational protocols of the Web (DNS, HTTP, and HTTPS). Importantly, **TMC** *does not require access to vantage points in the country*. We apply **TMC** to 15.5M domains, our results reveal that Turkmenistan censors more than 122K domains, using different blocklists for each protocol. We also reverse-engineer these censored domains, identifying 6K over-blocking rules causing incidental filtering of more than 5.4M domains. Finally, we use Geneva, an open-source censorship evasion tool, to discover five new censorship evasion strategies that can defeat Turkmenistan's censorship at both transport and application layers. We will publicly release both the data collected by **TMC** and the code for censorship evasion.

CCS CONCEPTS

• **Social and professional topics** → **Censorship**; • **General and reference** → **Measurement**.

KEYWORDS

Web Filtering, Turkmenistan, Censorship Measurement

ACM Reference Format:

Sadia Nourin, Van Tran, Xi Jiang, Kevin Bock, Nick Feamster, Nguyen Phong Hoang, and Dave Levin. 2023. Measuring and Evading Turkmenistan's Internet Censorship: A Case Study in Large-Scale Measurements of a Low-Penetration Country. In *Proceedings of the ACM Web Conference 2023 (WWW '23), April 30–May 04, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3543507.3583189

WWW '23, April 30–May 04, 2023, Austin, TX, USA

ACM ISBN 978-1-4503-9416-1/23/04…$15.00
https://doi.org/10.1145/3543507.3583189

1 INTRODUCTION

Internet censorship by powerful nation-states threatens free and open communication for those living within their borders [43, 54]. For decades, researchers and practitioners have focused considerable efforts towards measuring, understanding, and circumventing censorship across the globe, with particular focus on the largest and most powerful censoring regimes, like China [11, 29, 31], Iran [10, 12, 16, 56], and India [61]. The methods developed to meet the massive scales of these efforts range from recruiting participants to deploying "probes" within the censoring regimes [25] to finding active "echo servers" [49], VPNs [32, 33, 45], or other responsive devices [58] to receive censored traffic.

Although previous efforts have been effective at measuring censorship in different regions of the world, they face many challenges when it comes to *small countries*, especially those with a low Internet penetration rate. For instance, it can be difficult or risky to recruit local volunteers to test "potentially censored" websites in repressive countries with a small population as their network probes will likely stick out from other "allowed" network traffic. For countries with a low Internet penetration rate, it is similarly challenging to acquire in-country vantage points or identify viable VPNs or responsive servers for remote network measurements.

In this paper, we introduce techniques for measuring and evading censorship of countries with low Internet penetration rates, without relying on traditional in-country resources like vantage points or volunteers. In contrast to previous measurement techniques that require servers or participants within a censoring nation-state, our techniques exploit the fact that some smaller countries' censorship infrastructure can be tricked into believing that an external host has connected to an internal IP address, even if that IP address is not actually in use. Bock et al. [14] used this characteristic to launch amplification attacks; we use it to develop techniques to measure and evade censorship.

We focus our study on Turkmenistan for several reasons. Most importantly, Turkmenistan's Internet censorship behavior presents a rare opportunity for scalable remote measurements to investigate network filtering across all three foundational protocols of the Web: DNS, HTTP, and HTTPS. Second, there have been recent reports of more restrictive Internet policies in the country [50, 51], resulting in sudden increases in the number of clients seeking to use anonymous network relays such as Tor and I2P since 2021 [1, 46]. Finally, numerous anecdotes have reported instances of some

popular websites being censored [2, 3] while there has yet to be a large-scale and systematic study on the country.

Motivated by these developments, our paper seeks to systematically answer the following questions about Turkmenistan's censorship: (1) What websites are censored, and over what protocols? (2) How does the censorship infrastructure work? and (3) How can we evade Turkmenistan's censorship?

To answer these questions, we present the design and implementation of **TMC**, a large-scale measurement system capable of testing millions of domains from outside a censoring nation-state without having access to internal vantage points (§3). **TMC** takes advantage of an important characteristic of Turkmenistan's censorship that is common (though not pervasive) among nation-state censors: it employs "bi-directional" censorship. Bi-directional censors act on traffic the same way regardless of whether the client or server is within their borders. It applies to all traffic even if the connection did not originate within the censored country. In contrast, "uni-directional" censors apply filtering policy solely on network traffic originated from within their jurisdictions.

Turkmenistan's bi-directional censorship was discovered through anecdotal accounts by many users [3]. Due to its bi-directional nature, we can originate all the measurement traffic from machines we control outside the country. However, bi-directional censorship alone was not enough for us to perform our measurements to *every* IP address within Turkmenistan's borders. To do so, we needed to develop additional, novel techniques to trick the censor into believing we are communicating with an arbitrary IP address—even if that address is not responsive to us—and then detecting censorship had taken place. We summarize our empirical findings as follows:

- Using **TMC**, we examine the blocking status of more than 15.5M fully qualified domain names (FQDNs) and detect a total of 122K censored domains (§5).
- Using these censored domains, we reverse engineer the actual blocklists used by Turkmenistan's filters, finding 6K over-blocking regular expressions that can cause large collateral damage to more than 5.4M domains unrelated to the domains that we believe Turkmenistan intended to block (§5).
- We use Geneva [18], an automated evasion tool, to discover novel censorship evasion strategies. In addition to finding that some evasion techniques that work in China [19] and Iran [16] also work in Turkmenistan, we discover five new strategies that can defeat Turkmenistan censorship at both transport and application layers (§6).

These contributions not only close the gap in the community's understanding of Web censorship in Turkmenistan but also come up with effective censorship evasion strategies that will hopefully assist in the development of circumvention tools to bypass the country's censorship at different layers of the network stack. The datasets collected by **TMC** and the code for evasion strategies that we discovered will be made publicly available.

2 BACKGROUND AND MOTIVATION

In this section, we first provide an overview of Turkmenistan's information control policies. We then discuss how the country uses different techniques for Web filtering, the challenges we initially faced when attempting to measure censorship, and how they have motivated us to conduct this study.

2.1 Turkmenistan's Information Controls

From a sociopolitical perspective, Turkmenistan has a freedom score of only 2/100 (1 is the lowest) ranked by the Freedom House in 2022 [26]. This score is reflective of a series of suppressive activities by the Turkmen government, including the suppression of press freedom, strict control of all broadcast and print media, as well as state-owned Internet service providers [27].

The government has been using different tactics to keep an inclusive and freely accessible Internet at check. Specifically, Internet access is expensive due to a state monopoly while broadband speed is among the World's slowest [47, 51]. Moreover, authorities strictly monitor communications [47] and ban "uncertified" encryption software. For instance, VPN users may face a penalty of seven years in prison [40] and Turkmen citizens have reported that they were required to swear on the Quran not to install a VPN [50].

2.2 Turkmenistan's Censorship Mechanisms

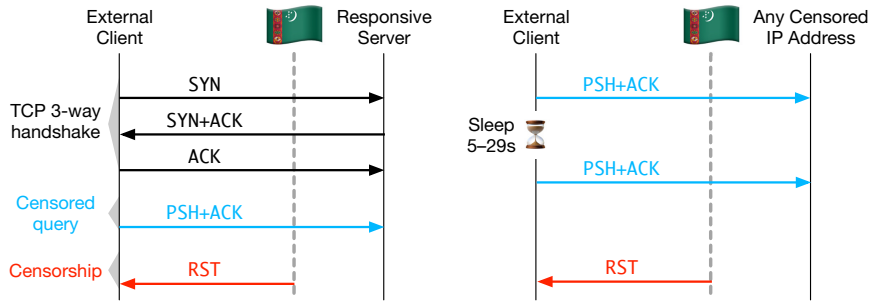
Together with restrictive Internet regulations, the Turkmen government also makes use of different network interference techniques for Web censorship. In August 2021, researchers reported that Turkmenistan was employing significant censorship of the Net4People community [3], targeting all three foundational protocols of the Web: DNS, HTTP, and HTTPS. Because Turkmenistan was applying this censorship in a bi-directional manner—that is, it was censoring traffic regardless of whether it was the client or the server inside their borders—we were able to reproduce and understand how they were censoring each of these three protocols:

DNS DNS tampering works by taking advantage of the race condition of the DNS protocol [44] (when a query is sent over UDP) to inject a forged DNS response containing wrong resource records of the domain being queried. To trigger a DNS injection from outside Turkmenistan, one only needs to send a DNS query containing a DNS Query for a censored domain (e.g., `twitter.com`) to an IP located inside the country. The censor will inject a DNS response packet containing `127.0.0.1` as the resource record for the censored domain (see Figure 1(a)).

HTTP For HTTP blocking, we can test if a domain is censored by initiating a TCP connection with an HTTP server followed by an `HTTP GET` request containing a censored domain in the `Host` field. Upon detecting the censored domain, the censor will inject one `RST` (reset) packet to tear down the connection (see Figure 1(b)).

HTTPS HTTPS censorship can be triggered in much the same way. First, we can complete a TCP connection with an HTTPS server located inside Turkmenistan. Then, in the very next `PSH+ACK` packet—corresponding to the TLS Client Hello message—we set the Server Name Indication (SNI) field to a censored domain. This causes the Turkmen censor to inject a `RST`, also shown in Figure 1(b).

Unlike the original Net4People report, we find that censorship for all three protocols is not restricted to the protocols' traditional ports. For example, although HTTP traditionally runs on port 80, we can trigger HTTP censorship to *any* destination port.



(a) DNS filtering over UDP. (b) Standard TCP censorship (HTTP and HTTPS). (c) Triggering TCP censorship without a server.

Figure 1: Turkmenistan's firewalls can be triggered from outside due to their bidirectional blocking behavior. The `PSH+ACK` contains the censored domain in the GET request for HTTP and the SNI field for HTTPS. TMC exploits the fact that censorship can be triggered and sends a second `PSH+ACK` after waiting for 5-29s after the first one containing the censored domain was sent.

We find that both HTTP and HTTPS filters exhibit *residual censorship* [15]. After triggering censorship by including a censored domain in the HTTP `Host` header or in the TLS SNI field, *any* subsequent packet matching the same TCP four-tuple (source IP:port, destination IP:port) will cause the censor to inject a `RST` packet. We determined that this residual blocking behavior stops after 30 seconds from the last injected packet.

2.3 Measurement Challenges

Despite the ease confirming bidirectional censorship, we face several challenges when trying to examine censored domains *at scale*.

Scarcity of Volunteers and Vantage Points. The Open Observatory of Network Interference (OONI) [25], ICLab [45], and Censored Planet [49] are active censorship measurement platforms capable of monitoring censorship across many regions of the world. However, across all three platforms, there are relatively few data points on Turkmenistan to provide a comprehensive picture of the country's Web censorship. (We perform a more detailed comparison to related work in §7). Understandably, given how slow, expensive, and strictly regulated the Internet is in Turkmenistan, recruiting local users to run Web connectivity tests with adequate frequency from inside the country is difficult and potentially risky: network probes will likely stick out from other "allowed" traffic. Furthermore, since VPN usage is forbidden [40, 50], a measurement system like ICLab [45] that largely depends on commercial VPNs will have very few viable vantage points in the country for running measurements. Further, with only 22.7K IPv4 addresses allocated for six autonomous systems (ASes), finding enough responsive servers (e.g., open DNS resolvers, and HTTP(S) servers) from public infrastructure for remote measurements is unlikely to succeed.

Measurement Machines Being Blocked. Inspired by earlier work on investigating China's bidirectional censorship [31], we tried checking which domains are censored via DNS tampering by sending DNS queries to a non-responsive IP located inside Turkmenistan in late 2021. This worked well for a day, but we then found the censor stopped injecting forged responses to our measurement machine: the IP of our probing machine was effectively "banned". Even at the time of writing this paper, probes originated from that IP still do not trigger any injections. To the best of our knowledge, this is the first time we observe an adversarial censor that intentionally hinders censorship measurements by ignoring probing traffic for such a long time (now more than half a year).

Inconsistent Blocking Across Different IPs. To reduce the likelihood of our measurement infrastructure getting blocked, we tried reducing the amount of probes sent to each IP while also originating our probes from multiple different IPs. Surprisingly, even when testing from the same source IP address, we discovered that filtering policies are not applied equally to all destinations within Turkmenistan. Not only does censorship vary within different IP prefixes in the same AS, but we also find variability at the granularity of different IP addresses within the same `/24` subnet (discussed in §4). This effect is visible even in the AS (AS20661) studied in the original Net4People discussion. If we send a censored DNS query (`twitter.com`) to the IP address `95.85.96.36`, we can trigger a DNS injection; however, changing the destination to the adjacent `95.85.96.35`, we find no censorship at all.

Collectively, these observations point to a fascinating question: if every IP within Turkmenistan is potentially censored in different ways, then how can we measure how *every* IP with the country is being censored? However, they also point to challenges that, when combined with those from §1, motivated us to design a measurement platform that can sustainably (i.e. **TMC** is not adversely affected if Turkmenistan blocks our probe machines' IP) and exhaustively measure Turkmenistan's Web censorship infrastructure.

3 TMC DESIGN

Taking into account the aforementioned challenges, we design **TMC** with the following objectives in mind. The system should be able to **i)** confirm which IP addresses are actively being filtered, **ii)** sustainably probe as many domains as possible across all three protocols (DNS, HTTP, and HTTPS) to detect censored domains, and **iii)** reverse-engineer the blocking rules of censored domains.

3.1 Probing Mechanisms

First and foremost, our measurement system has to achieve the above objectives without relying on local volunteers or having access to vantage points inside Turkmenistan. **TMC** addresses this by sending carefully crafted probes that elicit censorship without requiring any participation from within Turkmenistan.

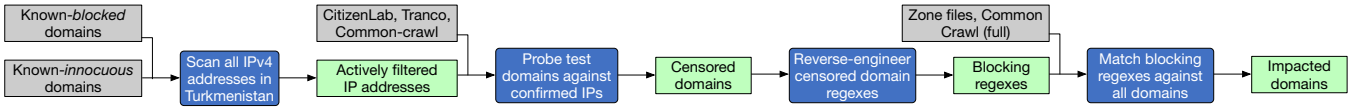


Figure 2: An overview of TMC design. Gray boxes denote external datasets; blue boxes denote actions taken by TMC; and green boxes denote TMC's findings.

DNS As shown in Figure 1(a), probing the DNS filter can easily be done due to the stateless nature of DNS-over-UDP, which is still the dominant protocol used for DNS resolutions to date [28]. To do so, we simply send a DNS query to an arbitrary (possibly even unused) IP address within Turkmenistan.

HTTP and HTTPS Probing the HTTP and HTTPS filters at scale is more challenging. Recall from Figure 1(b) that, traditionally, these protocols require completing a TCP three-way handshake, but this would restrict us to only studying responsive IP addresses.

In 2021, Bock et al. [14] showed that some stateful censors could be tricked into responding without a complete TCP handshake: by foregoing the handshake altogether and simply sending the `PSH+ACK` containing a censored domain. In our initial experiments, we tried to send a single `PSH+ACK` packet with a forbidden `Host` header, but we did not get a response from the censor. However, after repeatedly running these tests, we discovered that by sending one probing packet, waiting for 5 to 29 seconds, and then sending the same probing packet again, we *can* trigger both HTTP and HTTPS censorship, as shown in Figure 1(c). The "sleep" is the time period of the residual censorship, which is the reason why sending the second probing packet triggers a RST from the censor. These bounds are tight: if we sleep less than 5 seconds or more than 29, we do not observe any injected tear-down packets.

Upon further testing, we observed that the first probing packet must have the censored domain; the second packet can be any non-`RST`[1] packet with the same TCP 4-tuple. If the second probing packet is a `SYN+ACK` or a `SYN`, the filtering middleboxes will inject a `RST+ACK` instead of a `RST`.

These findings align with the residual blocking behavior that we noticed in §2.2. More specifically, it appears that the first `PSH+ACK` does not elicit a `RST`, but it does engage residual censorship, which results in a `RST` in response to the second packet. This corroborates one of our transport-layer evasion strategies in §6.1.

We believe the cause of this strange blocking behavior is the Turkmen censor trying to be tolerant to packet loss or asymmetric routes. In this manner, if the censor does not see the three-way handshake complete, it will still be able to make a censorship decision if a forbidden connection continues. This is not unusual: prior studies have shown sophisticated censors around the world often have more than one filtering system in place as a backup to cope with failures of other filtering systems [11, 16, 19, 31].

In all the above scenarios, we could confirm that injected packets are truly from Turkmenistan's firewalls: they all share the same distinctive and consistent signature in the IP header with (1) the `IP.ID` field set to `30000` and (2) initial `IP.TTL` value of 128. Injected packets observed at our probing machines will thus have `IP.TTL` values equal to 128 minus the number of hops between our probing machines and the filtering middlebox.

[1]Any TCP flag set to `PSH`, `FIN`, `URG` and/or `ACK` can trigger an injection from both HTTP and HTTPS filters. Since our probing packets encapsulate test domains in their application-layer payload, we opt to use `PSH+ACK` as the flags for our measurement system so that our traffic does not noticeably stick out from normal TCP packets that carry data in their application-layer payload, reducing the probability that our measurement machines will be blocked quickly.

3.2 Overall Architecture

The overall architecture of our measurement system is illustrated in Figure 2, which is comprised of four main tasks.

Confirming actively filtered IPs. As discussed in §2.3, Turkmenistan's filtering is applied differently at the granularity of each IP even when both filtered and non-filtered IPs belong to the same `/24` subnet announced via BGP. We are thus interested in examining the entire IP space of the country to have a comprehensive view of which IPs are actively being filtered. For this task, we obtain all IP prefixes allocated for ASes in Turkmenistan from CAIDA's *pfx2as* dataset [20]. For each IP, we send packets encapsulating opt-out, known blocked, and innocuous domains using probing mechanisms described in §3.1 to confirm which IPs are actively being filtered by Turkmenistan's firewalls.

Ethical Considerations. A primary goal of our system's design is to measure in an ethical and responsible manner. Unlike measurements conducted by volunteers [25] or machines that researchers can fully control [31, 45], this task involves sending probes destined for IP addresses not under our control. While wide network scanning activities are common on today's Internet [7, 14, 24], due to the sensitive nature of censorship measurement, we follow best practice for scanning at scale by providing an *opt-out* mechanism. Specifically, our probes are accompanied by packets encapsulating a non-censored domain under our control, from which our contact information and description of the study can be found to request opt-out. For more than two months running **TMC**, we did not receive any opt-out requests or complaints.

One may wonder whether our measurement system and evasion strategies will help the censor to enhance its filtering capability. The general consensus from the anti-censorship community over the years has been that work in this space helps the evaders more than the censors. The packet sequence used for our measurement system exploits a fundamental aspect of the middlebox, TCP noncompliance, allowing the censor to inject packets or block a connection even when they do not see all of the packets in a connection [14]. This fundamental aspect of the middlebox cannot be easily fixed. The same reasoning applies to the evasion strategies. Censors may patch trivial bugs, rendering a few evasion strategies ineffective. They, however, may not be able to easily fix the fundamental problems that enable the myriads of other strategies to succeed.

Detecting censored domains at scale. Recall from §2.3 that Turkmenistan's censorship infrastructure ignores traffic from our measurement machines after some time. To address this, we deploy

our measurement machines across different commercial virtual private servers (VPS) and frequently change their source IP addresses. After confirming actively filtered IP addresses in the previous task, we distribute our probes across these confirmed IP addresses while also scattering packets over different ports. Designing our measurement in this fashion helps us to avoid both (1) false negatives due to our measurement machines being banned and (2) false positives due to the residual censorship applied on the same TCP four-tuple as discussed in §2.3. We could use different port pairs for this task because Turkmenistan's firewalls filter on all network ports, not just standard ports (i.e., 53 for DNS, 80 for HTTP, and 443 for HTTPS).

As shown in Figure 2, the payload of our probes contains domains curated from the Citizen Lab lists [5], the full Tranco list [42], and Common Crawl Project [8]. Due to limited resources of our VPS, we opt to probe the first 10M FQDNs ranked by the Common Crawl Project instead of the full list of almost 400M FQDNs. The rationale behind selecting most popular domains is to shed light on the blocking status of sites that are often visited by most Internet users. Moreover, aggressively probing all 400M FQDNs is impractical since we do not own the IPs being probed and would likely cause our measurement traffic to be ignored more quickly. To that end, **we probed a total of 15.5M unique FQDNs** in September 2022.

Reverse-engineering blocking regular expressions. Some initial observations reported in [3] also indicate that Turkmenistan's filters employ regex-based blocking. To identify these rules, once a domain is detected to be censored by **TMC**, it is broken down into substrings with different length, prepended and appended with different random characters. These combinations of different substrings and random characters are then probed again to identify the shortest rule that could trigger the filtering middlebox. For instance, when **TMC** detected `account.trendmicro.com` to be censored, our system will carry out this task to reverse engineer the actual blocking regular expression of `.*\.trendmicro\.com.*`.

Identifying impacted domains on the Internet. Though we could not probe every single domain on the Internet, it is still our goal to assess the impact scale of Turkmenistan's regular-expression-based censorship. After we reverse engineer the regular expressions that the Turkmen censors use, we can test domains offline to see if they match any of the rules. We scanned all regular expressions that **TMC** discovered against all FQDNs that we could obtain from DNS zone files provided via ICANN's Centralized Zone Data Service [6] and the full host list from the Common Crawl Project [8], totaling 718M FQDNs.

4 WHO IS BEING CENSORED?

We begin our analysis by investigating which IP addresses within Turkmenistan are being subjected to censorship.

During August and September 2022, we used **TMC** to scan the entire IP address space of Turkmenistan to determine which addresses trigger censorship for DNS, HTTP, and HTTPS. Figure 3 shows the total number of IP addresses over time for each of these protocols. The low numbers in the first few days of our measurement window are a measurement artifact: this was before we learned Turkmenistan ignores our measurements after a certain amount of time. After those initial days, we switched to our distributed measurement approach (§3.2), which gave us consistent results.

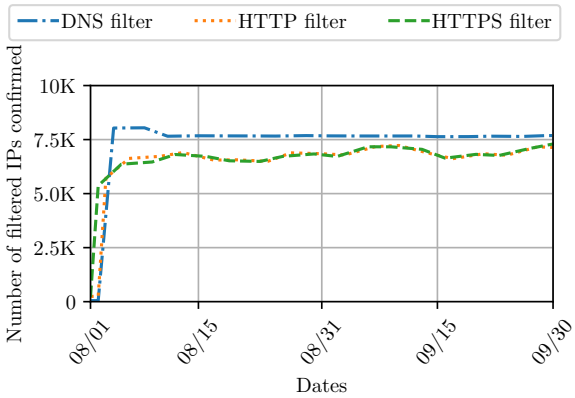


Figure 3: Number of filtered IPs confirmed over time.

Figure 3 in Appendix A shows that **TMC** could detect >7.5K IPs being actively filtered on a daily basis, occupying about 33% of all IPs allocated for ASes in the country. The IPs actively being filtered are similar across all three protocols. Although the purpose of this task is to confirm which IPs are actively filtered, we refrain from sending several probe packets to live hosts. IPs responding to probes containing the opt-out and innocuous domains are thus excluded from this plot and not used for later probing tasks (§5). For that reason, the number of probe-able IPs confirmed via probing HTTP and HTTPS filters are less than that of the DNS filter.

Actively filtered IP prefixes. At the time of conducting our measurement, there are six ASes allocated with a total 22.7K IPs. These IPs are announced via 24 prefixes as shown in Table 2 (Appendix A). Our probing results show that not all of these ASes are actively filtered. Even in ASes with IPs that **TMC** detects to be filtered, filtering is not applied across all addresses.

From Table 2, we can see that the vast majority of filtered IPs are allocated for AS20661 (State Company of Electro Communications Turkmentelecom). In this AS, `217.174.224.0/20` and `95.85.96.0/19` are the two subnets with the largest number of filtered IPs (more than 6.5K IPs). Two other ASes from which **TMC** detects network interferences are AS51495 (Telephone Network of Ashgabat CJSC) and AS201558 (State Bank for Foreign Economic Affairs of Turkmenistan).

These findings explain why we initially could not trigger censorship when probing `95.85.96.35`. This is because only 65.5% of IPs in `95.85.96.0/24` are actively being filtered by Turkmenistan's firewalls. Our findings underscore the importance of confirming actively filtered IPs to avoid false negatives when probing against non-filtered network locations.

AS topology. To better understand where censorship is taking place within Turkmenistan's network, we next look at its AS topology. We utilize CAIDA's AS Rank [4] to determine customer-provider relationships between the different ASes. We then conduct `traceroute` for every IP prefix to obtain the routes and routers' information via which our probing packets traverse. To determine where network filtering happens, we use the limited-TTL method to send multiple probe packets, encapsulating a known censored domain, to each IP prefix. More specifically, we incrementally increase the `IP.TTL` of our packets until we could trigger an injection from

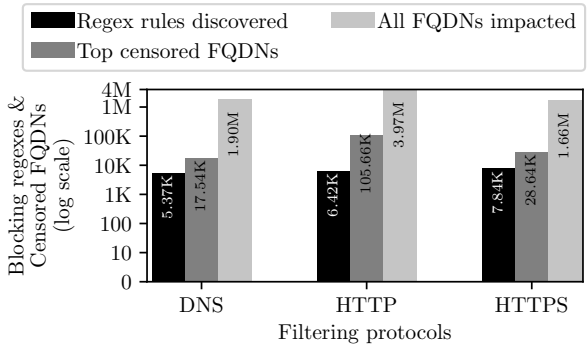


Figure 4: The number of censored FQDNs, filtering regular expressions, and impacted FQDNs across three protocols.

the filtering middleboxes. Ultimately, we were able to obtain the AS topology for Turkmenistan as shown in Figure 5 (Appendix A).

We could identify two national gateways through which all packets destined for any IP address in Turkmenistan will have to pass through. One is in AS20661 and the other one is in AS51495. Our limited-TTL experiments also indicate that the filtering middleboxes are posited behind these gateways since our probing packets will trigger an injection as soon as their TTL value is large enough to pass through the gateways. We could also confirm that these filtering boxes share the same blocking signature, i.e., the `IP.ID` field of an injected packet is always set to `30000` (§3.1).

5 WHAT IS BEING (OVER-)BLOCKED?

Throughout the entire month of September 2022, we used **TMC** to probe (via DNS, HTTP, and HTTPS) 15.5M test domains against all of the IP addresses we verified as being censored (§4). Here, we report on what content is being blocked (and over-blocked).

5.1 Censored Domains

TMC detects 122K censored FQDNs, with a different number of censored domains being detected for each protocol as shown in Figure 4. There are 17.54K domains filtered via DNS tampering, 105.66K domains filtered if detected in the `Host` field of an HTTP GET packet, and 28.64K domains filtered if detected in the SNI field of a TLS *ClientHello* packet.

Categorization of Blocked Domains. To better understand the primary motivation behind Turkmenistan's Internet censorship, we utilize Virus Total's classification service [9] to categorize the blocking regular expressions that **TMC** has discovered. To avoid over-counting, if there are two FQDNs that match the same inferred regular expression, we only count one of them. For example, the blocking of `m.twitter.com` and `www.twitter.com` should only be counted as one blocking under the rule `.*twitter\.com.*`.

Figure 6 shows the top categories to which censoring regular expressions belong. Adult Content (often pornography) alone occupies almost 25% of all blocking rules. The second most dominant group of blocking rules is classified as "unknown". Upon manual verifications, the vast majority of these domains are either (1) not hosting any Web content or (2) currently not actively online. These domains might have been blocked in the past prior to our work. Previous work has shown that once domains are added to blocklists of nation-state firewalls they often remain blocked for an extensive period of time regardless of their inactive status [31]. Other top categories including typical ones that have been observed in other countries include Business, News, and Proxy Avoidance (often used for circumventing censorship). Together, this group of top 20 categories makes up almost 75% of all blocking regular expressions.

A taxonomy of censored domains based on their language and popularity ranking may also be desirable. A categorization based on website language, however, is challenging as many websites serve localized content and have different language versions for different populations. In addition, measuring popularity is not a straightforward task either because a meaningful popularity ranking needs to consider several factors, including when, by who, and from where these websites are visited.

5.2 (Over-)Blocking Rules

By reverse-engineering the blocking rules of these FQDNs, we discover 16.5K unique regular expressions that define Turkmenistan's firewalls rules. Although the majority of these rules are "properly" implemented (i.e., blocked domains are actually subdomains of the blocking rule), there are more than about 6K rules that would cause over-blocking of unrelated domains (i.e., blocking of domains whose second-level domain is unrelated to the blocking regular expression). For instance, we discover the DNS tampering rule `.*\.cyou.*` impacts not only the entire `.cyou` top-level namespace with more than 770K second-level domains but also other 117K FQDNs that happen to contain that string.

Other extreme blocking rules include `.*wn\.com.*`, `.*u\.to.*`, and `.*w\.org.*`, causing over-blocking of hundreds of thousands unrelated domains. The log scale of Figure 4 shows the significant magnitude of the collateral damage that these over-blocking rules would cause, potentially affecting millions of FQDNs. A more detailed table listing top over-blocking rules can be found in Table 3 (Appendix B).

Unhealthy Over-blocking of DNS-over-HTTPS. Traditional DNS does not encrypt or authenticate its payloads, making it trivial for censors to observe and inject lemon responses [55]. To this, several domain name encryption technologies have been proposed, e.g., DNS over TLS (DoT) [39] and DNS over HTTPS (DoH) [38]. By encrypting DNS traffic, these protocols take away the visibility into plaintext DNS resolutions from the censors, effectively circumventing their DNS filtering [32, 35, 36]. Nevertheless, this has led to a new wave of blocking strategies targeting domain name encryption protocols [13, 21, 32, 37]. **TMC** also detects such blocking efforts in Turkmenistan. The DoH resolvers that **TMC** detected also align with those reported in the discussion of Net4People community [3].

What we found intriguing, however, is the DNS filtering rule `^doh\..*`, which we believe to be used for DoH blocking. Specifically, any DNS query containing a domain starting with `^doh\.` will trigger a fake DNS injection. While this rule is effective at blocking many publicly available DoH resolvers [23], it also causes over-blocking of totally unrelated websites, especially those used by Departments of Health in many jurisdictions (e.g., `doh.gov.ae`, `doh.gov.ph`, `doh.gov.uk`, `doh.pa.gov`, `doh.wa.gov`).

Blocking of educational domains. Although educational domains are not among top blocking categories, **TMC** found

numerous domains of higher-educational institutions being blocked, including `.*brookings\.edu.*`, `^liberty\.edu$`. `^hds\.harvard\.edu$`, `^scs\.illinois\.edu$`. Interestingly, many censored domains belong to institutions in Michigan, such as `.*cmich\.edu.*`, `.*wmich\.edu.*`, `.*\.kettering\.edu.*`, and `.*med\.umich\.edu.*`. Based on the regular expression of each blocking rule, while some intend to block particular departments' website many rules exhibit a *blanket* blocking behavior.

6 CENSORSHIP CIRCUMVENTION

We discovered novel censorship evasion strategies in Turkmenistan by using Geneva, an open source genetic algorithm that automatically learns by training against live censors [18]. For all of our testing, we used machines under our control outside of Turkmenistan, and issued requests to affected IP addresses inside the country. We started training Geneva to discover censorship evasion strategies in March 2022. For this training, we used domains known to be censored in Turkmenistan through anecdotal accounts [3]. After letting Geneva automatically discover new evasion strategies, we performed follow-up experiments to more fully understand *why* the various evasion strategies work.

Geneva can learn new evasion strategies by manipulating either TCP/IP headers or application-layer payloads (specifically HTTP and DNS). We describe these two classes of strategies in turn.

Altogether, we found five novel evasion strategies for Turkmenistan, and re-discovered several successful strategies that prior work found against other nation-state censors [18]. All the strategies we discover work 100% reliably. Following precedence from prior work, we provide Geneva's strategy syntax; they can be copy-pasted into Geneva's open-source engine and used to evade censorship today.

6.1 TCP-Based Evasion Strategies

We allowed Geneva to train by manipulating TCP/IP headers of censored HTTP traffic. For every such evasion strategy Geneva found, we also evaluated it on HTTPS traffic.

Strategy 1: TCP Segmentation Geneva discovered evasion strategies that segment the TCP payload at specific portions of the payload. For HTTP, the strategy succeeds by segmenting the HTTP version. For example, segmenting a `GET` request for Twitter into (1) `"GET / HTTP/1"`, (2) `".1\r\nUser-Agent:..."` evades censorship, but any other segmentation index that does not split the HTTP version fails to evade censorship. For HTTPS, segmentation that splits the first few bytes of the TLS Client Hello header: specifically, any segmenting at byte index 3-8 (the Record Header of the Client Hello) inclusive evades censorship. We also find that segmentation that splits the Server Name Indication (SNI) field evades HTTPS censorship.

Strategy 1: (HTTP) Segmentation (Similar for HTTPS)

```
[TCP:flags:PA]-fragment{tcp:8:True}-| \/
```

We believe this strategy works by interfering with the censor's internal parsing used to identify the request as HTTP or HTTPS. Since the censor fails open, the request is allowed to pass through.

Strategy 2: TCB Teardown One of the most famous packet manipulation layer strategies is the TCB Teardown [18, 41, 59, 60]. A client performs this strategy by injecting a `RST` or `FIN` packet in such a way that it is not processed by the destination server, often by limiting the `TTL` (time-to-live) field. The censor processes the teardown packet, incorrectly assumes that the connection has been torn down, and stops tracking the connection, enabling the client and server to continue communicating censorship free.

Strategy 2: (HTTP & HTTPS) TCB Teardown via RST

```
[TCP:flags:S]-duplicate(,
  duplicate(tamper{TCP:flags:replace:R}(
    tamper{TCP:chksum:corrupt},),))-| \/
```

Geneva discovered such strategies in Turkmenistan. Strategy 2 sends a `RST` packet with a broken checksum immediately after sending the `SYN` packet. Geneva identified other variants of this strategy, including tearing down with a `FIN` packet. Any combination of flags that includes a `RST` or a `FIN` is sufficient to evade censorship, including nonsensical flag combinations like `PSH+RST+FIN+SYN`. These evade censorship for HTTP and HTTPS.

We also tested the TCB Teardown strategies discovered by Bock et al. [18], and found that all of them are successful against Turkmenistan's HTTP and HTTPS censorship.

Strategy 3: Free Pass Geneva discovered a novel strategy against Turkmenistan's censor. This strategy sends a `RST` or `FIN` packet *before* sending the initial `SYN`. At first, this strategy appears to be a TCB Teardown attack, but it is not: since the `SYN` packet comes after the `RST`, the censor should not yet have any state to tear down. Further, we find that the injected `RST` or `FIN` packets only stave off censorship if the `SYN` packet is sent less than 5 seconds later. If the `SYN` is sent on or after the fifth second, the strategy no longer works. This strategy works for both HTTP and HTTPS.

Strategy 3: (HTTP & HTTPS) Free Pass (for < 5 seconds)

```
[TCP:flags:S]-duplicate(
  tamper{TCP:flags:replace:R},)-| \/
```

Frankly, we do not understand why this strategy works. We initially hypothesized that the `RST` and `SYN` packets might be arriving at the censor in the wrong order, causing a normal TCB teardown attack. But this is not the case: we can delay the `SYN` packet by up to 5 seconds after the `RST` packet, and we can still avoid censorship. These timing dynamics are also not present in the normal TCB teardown strategy. In the normal TCB Teardown case, the `RST` and `FIN` packets are effective for more than 5 seconds; we tested this by delaying the forbidden request after injecting the teardown packet.

The timing dynamics of this strategy mirror the dynamics we discovered with our measurement strategy: we do not receive censored responses from the server when we send an incomplete TCP handshake for 5 seconds from the first incomplete handshake, but get censored on the 5th second.

Amazingly, the client does not have to send a `RST` or `FIN` packet in order to evade censorship with this strategy: the client may simply elicit a `RST` from the server. This can be done by sending an innocuous `PSH+ACK` to the server *before* a TCP handshake has been established, causing the server to respond with a `RST`. This strategy suggests that there may be simple server-side evasion strategies [17] that are successful against Turkmenistan's censors.

6.2 Application-Layer Evasion Strategies

We also trained Geneva by manipulating DNS and HTTP payloads.

DNS Strategies: Elevated Count Above a Threshold When we set the `qdcount`, `ancount`, `nscount`, and/or `arcount` fields within the DNS query header to values above a certain threshold, we are able to bypass Turkmenistan's DNS injection. Through empirical testing, we determined that the threshold for all of these fields is 25. Although this elevated count is in violation of the RFC, many DNS servers still respond to such queries (as discovered previously [30]).

Strategy 4: (DNS) Elevated Count Above a Threshold

```
[DNS:*:*]-tamper{DNS:ancount:replace:32}-| \/
```

We note that the censor does not inspect DNS response packets, so if the DNS request itself is not censored, the request should succeed.

We also found that the DNS censor changed during the course of our study. Geneva initially found a simple strategy that created a second DNS question record without incrementing the query count (`qdcount`) field. This strategy evaded censorship: we would not get an injection with an A record pointing to 127.0.0.1, but instead a corrupted injection with an A record pointing to 0.0.1.44 (which a normal client would ignore). After further analysis, we realized that the response the censor had sent was malformed. The injected response had two answer records, but the response's `arcount` field was set to 1. This gave way to a parsing error, which made it seem as if the DNS injection's A record was pointing to another IP address, when in reality the injection had two A records, both pointing to 127.0.0.1.

Interestingly, Turkmenistan caught their mistake during the course of our study. Around May 26th, 2022, Turkmenistan fixed the censor so that their responses would have the correct number of answer records. Now, the censor responds to Geneva's request with one A record pointing to 127.0.0.1 because the request declared a `qdcount` of 1. As a result, this strategy no longer works.

Strategy 5: HTTP Host Header Whitespace Geneva discovered one novel HTTP strategy that bypasses censorship by inserting additional whitespace within the host header. More specifically, the strategy inserts a tab and a new line right before the host header value.

Strategy 5: (HTTP) Host Header Whitespace

```
[HTTP:host:*]-insert{%09%0A:start:value:1}-
| \/
```

Harrity et al. [30] discovered 77 application-layer HTTP strategies that evade censorship in other nation-states, and we tested each against Turkmenistan. We find only 5 of these were successful; see the detailed breakdown in the appendix.

7 RELATED WORK

Measurement of Turkmenistan's Censorship. Over the past decade, there have been several efforts to measure censorship in Turkmenistan. Some of these focused specifically on Turkmenistan [48, 53, 57] while others performed global-scale measurements [25, 45, 52]. Table 1 presents a detailed comparison between our study and these previous efforts, showing how **TMC**'s unique measurement method has allowed us to gain a more comprehensive view into Turkmenistan's Internet censorship landscape only after a short period of time.

Table 1: Comparison between different Turkmenistan censorship measurement studies/platforms.

Study	When	Censored/Tested	Method	Coverage
SecDev [53]	07/12–08/12	34/unknown	local	unknown
Qurium [48]	07/19	133/10K	unknown	unknown
ValdikSS [57]	12/18/21	60/1K	unknown	1 AS
OONI (DNS) [25]	02/17–09/22	254/2.2K	volunteers	5 ASes
Satellite [52]	08/18–05/22	267/4.7K	open resolvers	2 ASes
TMC	09/22	122K/15.5M	*No* vantage point	All ASes

Remotely Measuring Censorship. There have been myriad prior efforts to develop techniques that allow one to measure censorship of a country without requiring a vantage point from inside that country. The CensoredPlanet platform [49] incorporates multiple techniques, such as Quack [58], that can remotely measure Internet censorship without participating users in the country. However, these generally require some form of responsive server (typically echo servers) within the country. Such servers are unfortunately not widely available in countries with low Internet penetration like Turkmenistan. **TMC** employs a novel sequence of packets that trigger censorship without requiring any server-side participation within the country; while this borrows techniques from Bock et al. [14], we believe we are the first to apply them to wide-scale detection of censorship.

Evading Turkmenistan's Censorship. While there has been some earlier effort in the community to manually craft packets to sidestep Turkmenistan censorship [3], we are not aware of any prior studies that have systematically investigated censorship circumvention across different network layers in Turkmenistan. While other general mechanisms work to varying degrees of success, such as tunneling censored traffic over anonymity networks (e.g., Tor [22], I2P [34]), we believe we are the first to find circumvention strategies specific to their censorship infrastructure.

8 CONCLUSION

In this paper, we presented the most thorough evaluation of Turkmenistan's censorship of the Web (DNS, HTTP, and HTTPS). We found that blocking is not applied to all IP addresses equally, and that there are millions of domains that are very likely over-blocked due to inaccurate regular-expression rules. These results would not have been possible with traditional measurement techniques, which require some degree of user participation or server availability within the censored country. The design of **TMC** has enabled us to perform a large-scale measurement in a low-Internet-penetration country like Turkmenistan. While the specific packet sequences designed for **TMC** may not work elsewhere, the high-level approach can. Our study is a first step towards country-wide measurements from the outside without access to responsive vantage points or volunteers. We hope that our paper will lead to more work in this direction. To assist in such future efforts, we make our measurement code and the evasion strategies discovered by Geneva to aid in its integration with any existing evasion software, publicly available at https://doi.org/10.5281/zenodo.7631411.

ACKNOWLEDGMENTS

We would like to thank all the anonymous reviewers for their thorough feedback. We also thank Sudhamshu Hosamane, Martin Lutta Putra, and others who preferred to remain anonymous for helpful comments and suggestions to improve this paper.

This research was supported in part by the National Science Foundation (NSF) through award CNS-1943240, and the Defense Advanced Research Projects Agency (DARPA) through award HR00112190126. The opinions in this paper are those of the authors and do not necessarily reflect the opinions of the sponsors.

REFERENCES

[1] 2010. Tor Metrics. https://metrics.torproject.org/userstats-relay-country.html?start=2021-01-01&end=2022-09-28&country=tm&events=off Accessed September 2022.
[2] 2019. Turkmenistan Blocks DNS-over-HTTPS Resolvers. https://ntc.party/t/turkmenistan-blocks-dns-over-https-resolvers/244
[3] 2021. Bidirectional DNS, HTTPS, HTTP injection in Turkmenistan. https://github.com/net4people/bbs/issues/80
[4] 2022. CAIDA AS Rank. https://asrank.caida.org/
[5] 2022. Citizen Lab Lists. https://github.com/citizenlab/test-lists
[6] 2022. ICANN Centralized Zone Data Service. https://czds.icann.org
[7] 2022. Shodan: The search engine for Security. https://shodan.io/
[8] 2022. The Common Crawl Project. https://commoncrawl.org
[9] 2022. Virus Total. https://www.virustotal.com/gui/home/url
[10] Collin Anderson. 2013. Dimming the Internet: Detecting Throttling as a Mechanism of Censorship in Iran. (2013). arXiv:1306.4361 [cs.NI]
[11] Anonymous, Arian Akhavan Niaki, Nguyen Phong Hoang, Phillipa Gill, and Amir Houmansadr. 2020. Triplet Censors: Demystifying Great Firewall's DNS Censorship Behavior. In *USENIX FOCI*.
[12] Simurgh Aryan, Homa Aryan, and J. Alex Halderman. 2013. Internet Censorship in Iran: A First Look. In *USENIX FOCI*.
[13] Simonetta Basso. 2021. Measuring DoT/DoH Blocking Using OONI Probe: a Preliminary Study.
[14] Kevin Bock, Abdulrahman Alaraj, Yair Fax, K. S. Hurley, Eric Wustrow, and Dave Levin. 2021. Weaponizing Middleboxes for TCP Reflected Amplification. In *USENIX Security Symposium*.
[15] Kevin Bock, Pranav Bharadwaj, Jasraj Singh, and Dave Levin. 2021. Your Censor is My Censor: Weaponizing Censorship Infrastructure for Availability Attacks. In *USENIX Workshop on Offensive Technologies (WOOT)*.
[16] Kevin Bock, Yair Fax, Kyle Reese, Jasraj Singh, and Dave Levin. 2020. Detecting and Evading Censorship-in-Depth: A Case Study of Iran's Protocol Filter. In *USENIX FOCI*.
[17] Kevin Bock, George Hughey, Louis-Henri Merino, Tania Arya, Daniel Liscinsky, Regina Pogosian, and Dave Levin. 2020. Come as You Are: Helping Unmodified Clients Bypass Censorship with Server-side Evasion. In *ACM SIGCOMM*.
[18] Kevin Bock, George Hughey, Xiao Qiang, and Dave Levin. 2019. Geneva: Evolving Censorship Evasion Strategies. In *ACM CCS*.
[19] Kevin Bock, Gabriel Naval, Kyle Reese, and Dave Levin. 2021. Even Censors Have a Backup: Examining China's Double HTTPS Censorship Middleboxes. *ACM FOCI* (2021).
[20] CAIDA. 2022. Routeviews Prefix to AS mappings Dataset for IPv4 and IPv6. https://www.caida.org/data/routing/routeviews-prefix2as.xml
[21] Zimo Chai, Amirhossein Ghafari, and Amir Houmansadr. 2019. On the Importance of Encrypted-SNI (ESNI) to Censorship Circumvention. In *USENIX FOCI*.
[22] R. Dingledine, N. Mathewson, and P. Syverson. 2004. Tor: The Second-Generation Onion Router. In *Proceedings of the 13th USENIX Security Symposium*. 303–319.
[23] DNS over HTTPS Public Resolvers. 2020. DOH. https://github.com/curl/curl/wiki/DNS-over-HTTPS
[24] Z. Durumeric, E. Wustrow, and J. A. Halderman. [n. d.]. ZMap: Fast Internet-wide Scanning and Its Security Applications. In *USENIX Security '13*.
[25] Arturo Filasto and Jacob Appelbaum. 2012. OONI: Open Observatory of Network Interference. In *FOCI*.
[26] Freedom House. 2022. Global Freedom Scores 2022. https://freedomhouse.org/countries/freedom-world/scores
[27] Freedom House. 2022. Turkmenistan Freedom Score 2022. https://freedomhouse.org/country/turkmenistan/freedom-world/2022
[28] Geoff Huston. 2022-09-02. DoH, DoT, and plain old DNS. APNIC. https://blog.apnic.net/2022/09/02/doh-dot-and-plain-old-dns/
[29] Geremie R. Barme And Sang Ye. 1997. The Great Firewall of China. https://www.wired.com/1997/06/china-3/ Accessed May 2022.
[30] Michael Harrity, Kevin Bock, Frederick Sell, and Dave Levin. 2022. GET /out: Automated Discovery of Application-Layer Censorship Evasion Strategies. In *USENIX Security Symposium*.
[31] NP. Hoang, AA. Niaki, J. Dalek, J. Knockel, P. Lin, B. Marczak, M. Crete-Nishihata, P. Gill, and M. Polychronakis. 2021. How Great is the Great Firewall? Measuring China's DNS Censorship. In *USENIX Security '21*.
[32] NP. Hoang, M. Polychronakis, and P. Gill. 2022. Measuring the Accessibility of Domain Name Encryption and Its Impact on Internet Filtering. In *PAM*.
[33] Nguyen Phong Hoang, Sadie Doreen, and Michalis Polychronakis. 2019. Measuring I2P Censorship at a Global Scale. In *Proceedings of the 9th USENIX Workshop on Free and Open Communications on the Internet*.
[34] Nguyen Phong Hoang, Panagiotis Kintis, Manos Antonakakis, and Michalis Polychronakis. 2018. An Empirical Study of the I2P Anonymity Network and Its Censorship Resistance. In *Proceedings of the 18th Internet Measurement Conference (IMC '18)*.
[35] Nguyen Phong Hoang, Ivan Lin, Seyedhamed Ghavamnia, and Michalis Polychronakis. 2020. K-resolver: Towards Decentralizing Encrypted DNS Resolution. In *Proceedings of The NDSS Workshop on Measurements, Attacks, and Defenses for the Web 2020 (MADWeb '20)*.
[36] Nguyen Phong Hoang, Arian Akhavan Niaki, Nikita Borisov, Phillipa Gill, and Michalis Polychronakis. 2020. Assessing the Privacy Benefits of Domain Name Encryption. In *Proceedings of the 15th ACM ASIA Conference on Computer and Communications Security (AsiaCCS '20)*.
[37] Nguyen Phong Hoang, Arian Akhavan Niaki, Phillipa Gill, and Michalis Polychronakis. 2021. Domain Name Encryption Is Not Enough: Privacy Leakage via IP-based Website Fingerprinting. In *Proceedings of the 21st Privacy Enhancing Technologies Symposium (PoPETs '21)*.
[38] P. Hoffman and P. McManus. 2018. *DNS Queries over HTTPS (DoH)*. RFC 8484. IETF. https://tools.ietf.org/html/rfc8484
[39] Z. Hu, L. Zhu, J. Heidemann, A. Mankin, D. Wessels, and P. Hoffman. 2016. *Specification for DNS over Transport Layer Security (TLS)*. RFC 7858. IETF.
[40] Human Right Watch. 2022. Turkmenistan Events of 2021. https://www.hrw.org/world-report/2022/country-chapters/turkmenistan
[41] S. Khattak, M. Javed, PD. Anderson, and V. Paxson. 2013. Towards Illuminating a Censorship Monitor's Model to Facilitate Evasion. In *USENIX FOCI*.
[42] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. 2019. Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. In *NDSS*.
[43] M. Xynou and A. Filastò. 2022. New Blocks Emerge In Russia Amid War In Ukraine: An OONI Network Measurement Analysis. Open Observatory of Network Interference. https://ooni.org/post/2022-russia-blocks-amid-ru-ua-conflict/
[44] P. Mockapetris. 1987. *Domain Names - Concepts And Facilities*. RFC 1034. IETF. https://datatracker.ietf.org/doc/html/rfc1034
[45] AA. Niaki, S. Cho, Z. Weinberg, NP. Hoang, A. Razaghpanah, N. Christin, and P. Gill. 2020. ICLab: A Global, Longitudinal Internet Censorship Measurement Platform. In *IEEE S&P '20*.
[46] NP. Hoang. 2018. I2P Metrics Portal. https://i2p-metrics.np-tokumei.net/router-distribution Accessed July 2022.
[47] OpenNet Initiative. 2010. Turkmenistan Report. https://opennet.net/research/profiles/turkmenistan
[48] Qurium. 2019. Turkmenistan And Their Golden DPI. https://www.qurium.org/alerts/turkmenistan-and-their-golden-dpi/
[49] RS. Raman, P. Shenoy, K. Kohls, and R. Ensafi. 2020. Censored Planet: An Internet-wide, Longitudinal Censorship Observatory. In *ACM CCS*.
[50] RFE/RL'S Turkmen Service. 2021-08-10. VPNs Are Not A-OK: Turkmen Internet Users Forced To Swear On Koran They Won't Use Them. https://www.rferl.org/a/turkmenistan-vpn-koran-ban/31402718.html
[51] RFE/RL'S Turkmen Service. 2022-01-13. Internet In Turkmenistan, Already The World's Slowest, Faces Further Restrictions. https://rferl.org/a/turkmenistan-internet-slowest-restrictions/31652467.html
[52] W. Scott, T. Anderson, T. Kohno, and A. Krishnamurthy. 2016. Satellite: Joint Analysis of CDNs and Network-Level Interference. In *USENIX ATC*.
[53] SecDev. 2012. Neither Here Nor There: Turkmenistan's Digital Doldrums. https://www.opensocietyfoundations.org/publications/neither-here-nor-there-turkmenistan-s-digital-doldrums
[54] Simone Basso and Maria Xynou and Arturo Filastò. 2022-09-25. Iran Blocks Social Media, App Stores And Encrypted DNS Amid Mahsa Amini Protests. Open Observatory of Network Interference. https://ooni.org/post/2022-iran-blocks-social-media-mahsa-amini-protests/
[55] Sparks, Neo, Tank, Smith, and Dozer. 2012. The Collateral Damage of Internet Censorship by DNS Injection. In *ACM CCR*.
[56] Thomas Erdbrink. 2018-05-01. Iran, Like Russia Before It, Tries to Block Telegram App. The New York Times. https://www.nytimes.com/2018/05/01/world/middleeast/iran-telegram-app-russia.html
[57] ValdikSS. 2021. Turkmenistan AGTS reachability test 18 Dec 2021. https://ntc.party/t/turkmenistan-agts-reachability-test-18-dec-2021
[58] B. VanderSloot, A. McDonald, W. Scott, J. Alex Halderman, and R. Ensafi. 2018. Quack: Scalable Remote Measurement of Application-Layer Censorship. In *USENIX Security Symposium*.

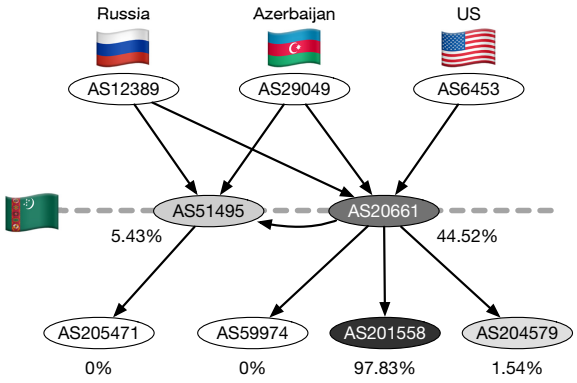


Figure 5: Turkmenistan's AS topology (edges are provider → customer). Percentages and gray-scale denote how many of the Turkmen AS's IP addresses are subjected to censorship.

B TOP CENSORED DOMAIN CATEGORIES, BLOCKING RULES AND SAMPLE CENSORED FQDNS

Figure 6 shows the top 20 domain categories censored in Turkmenistan. Adult Content alone occupies almost 25% of all blocking rules. This category, together with the Unknown (often inactive or parking domains), Business, and News, make up more than 50% of blocking regular expressions.

Table 3 shows the top ten overblocking rules with some sample impacted FQDNs. We can see that some blocking regular expressions such as `.*\.cyou.*`, `.*porn.*`, and `.*\.rocks.*` are all active generic top-level domains (gTLDs). These blocking rules, thus, would block access to not only all second-level domains (SLDs) registered under these gTLDs but also other SLDs happen to contain such strings. In addition, extremely short blocking rules like `.*w\.org.*` (a WordPress domain) tends to cause large collateral

Table 2: IPv4 prefixes allocated to Turkmenistan ASes and the average percentage of filtered IPs observed over time.

A	IP prefix	Filtered (%)	Organization
	103.220.0.0/22	0	
	119.235.112.0/20	0.34	
	177.93.143.0/24	0.41	
	185.69.184.0/24	0	State Company of Electro Communications Turkmentelecom
	216.250.8.0/21	0	
	217.174.224.0/20	**54.02**	
	95.85.96.0/19	**53.47**	
AS20661	95.85.100.0/22	9.58	
	95.85.100.0/24	9.45	
	95.85.101.0/24	9.52	
	95.85.104.0/22	11.13	
	95.85.104.0/24	11.33	Leased line customers
	95.85.96.0/24	**65.55**	
	95.85.98.0/24	0.065	
	95.85.99.0/24	0	
	93.171.220.0/22	5.43	
	93.171.220.0/24	0	
AS51495	93.171.221.0/24	0	Telephone Network of Ashgabat CJSC
	93.171.222.0/24	0	
	93.171.223.0/24	**21.72**	
AS205471	185.69.185.0/24	0	State Company of Electro Communications Turkmentelecom
AS59974	185.69.186.0/24	0	Mobile Customers Inet Access
AS201558	185.69.187.0/24	**97.83**	State Bank for Foreign Economic Affairs of Turkmenistan
AS204579	185.246.72.0/22	1.54	Turkmen hemrasi CJSC

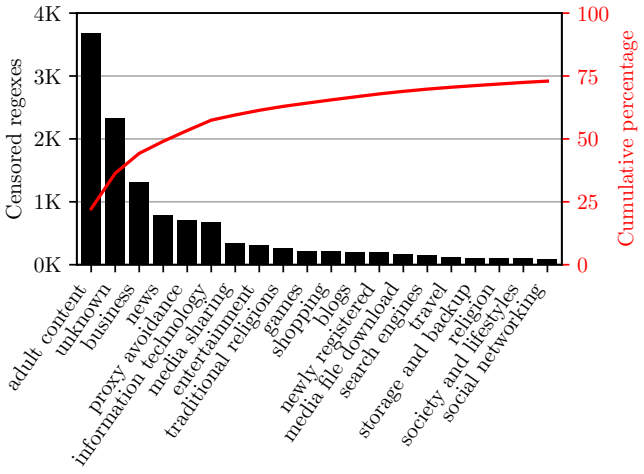


Figure 6: Top 20 domain categories of filtering regexes.

damage as they end up blocking totally unrelated yet valuable domains (e.g., `tensorflow.org`). Even internationalized domains, e.g., xn–vl2b99byzlcpd9yy.com, which is a Korean website unrelated to `.*yy\.com.*`, are impacted by these short blocking rules.

Table 3: Top 10 blocking regular expressions with the highest number of impacted FQDNs.

# impacted FQDNs	Regular expressions (Blocking protocols)	Sample domains
887K	.*\.cyou.* DNS, HTTPS	cyou-TLD zone, movizland.cyou, starlink.cyou committee.cyou, gomovies.cyou
568K	.*vpn.* HTTP	nordvpn.com, avira-vpn.com, expressvpn.com openvpn.net, vpnoverview.com, vpnmentor.com
480K	.*porn.* DNS	porn-TLD zone, pornhub.com, youporn.com, nopornnorthampton.org, pornphiphit.co.th antipornography.org
300K	.*wn\.com.* HTTP	wn.com, 423down.com, dawn.com uptodown.com, respawn.com, bandsintown.com
267K	.*u\.to.* HTTP	u.to, shahed4u.town, u.today hindilinks4u.to, dsu.toscana.it
217K	.*w\.org.* HTTP	cookielaw.org, w.org, hrw.org jw.org, democracynow.org, tensorflow.org
208K	.*xx\.com.* HTTP	code-boxx.com, uproxx.com, mixx.com idexx.com, tkmaxx.com, speexx.com
192K	.*\.rocks.* DNS, HTTPS	rocks-TLD zone, vox.rocks, say.rocks kavin.rocks, yolk.rocks
175K	.*twitter\.com.* DNS, HTTP, HTTPS	bretwitter.com, notrealtwitter.com spotwitter.com, johnnys-twitter.com financetwitter.com
173K	.*yy\.com.* HTTP	ammyy.com, abbyy.com, haliyy.com playdayy.com, xn--vl2b99byzlcpd9yy.com

C APPLICATION-LAYER HTTP STRATEGIES

Harrity et al. [30] recently discovered 77 application-layer HTTP using open-source modifications to Geneva. We tested all of those strategies, and report the successful 5 strategies here. Coincidentally, all five of these strategies were discovered against Kazakhstan's HTTP censorship. Note that although the same strategies work here as in Kazakhstan, they may not succeed for the same reason.

HTTP Strategy 1: Request Line Whitespace After Version

```
[HTTP:version:*]-insert{%20%0A%09:end:value:1}-|
\/
```

This strategy inserts a space, newline, and tab right after the HTTP version. Other spaces, newlines, and/or tabs may be inserted after the version as well, as long as these three characters are in the correct sequence. Any omission or swapping of the three characters causes the strategy to fail. However, a variant of this strategy, where one or more spaces are inserted after the version, succeeds.

HTTP Strategy 2: Request Line Whitespace After Method

```
[HTTP:method:*]-insert{%0A:start:value:1}-| \/
```

This strategy inserts a new line right before the HTTP method. The strategy succeeds with any number of new lines inserted before the method as long as there is at least one.

We believe these strategies work by breaking how the Turkmenistan censor identifies the HTTP version, which we know the censor relies on to parse.

HTTP Strategy 3: Sandwich Strategy Ver. 1

```
[HTTP:host:*]-insert{%20:end:value:3391}(
      duplicate(duplicate(,
            replace{a:name:1}),
                  insert{%09:start:name:1}),)-|
\/
```

The most complex strategy discovered by Harrity et al. [30] is the *sandwich strategy*. This strategy works by inserting specially crafted HTTP headers before and after the forbidden HTTP header to prevent the censor from reading the HTTP host successfully. This specific strategy has two components; first, it inserts 3,391 (or more) spaces after the host header value. Second, it creates new HTTP headers before and after the Host header (but with a tab character before the trailing header). We modified this strategy from Harrity's original; through manual experimentation, we discovered that 3,391 spaces is the minimum successful number to evade the Turkmenistan censor. If less than 3,391 spaces are inserted, the strategy does not evade censorship. We initially hypothesized this strategy was inducing TCP segmentation by increasing the packet size; but we find that when TCP segmentation occurs, it does not occur at indices that evade censorship when the spaces are omitted. Instead, we hypothesize this strategy is working by exceeding the maximum number of bytes that the Turkmenistan censor can process.

Here is a second variant of the strategy, which works the same way.

HTTP Strategy 4: Sandwich Strategy Ver. 2

```
[HTTP:host:*]-insert{%20:end:value:3391}(
      duplicate(duplicate(
            insert{%09:start:name:1},),
                  replace{a:name:1}),)-|\/
```