Understanding the Practices of Global Censorship through Accurate, End-to-End Measurements

LIN JIN, University of Delaware, USA
SHUAI HAO, Old Dominion University, USA
HAINING WANG, Virginia Tech, USA
CHASE COTTON, University of Delaware, USA

It is challenging to conduct a large scale Internet censorship measurement, as it involves triggering censors through artificial requests and identifying abnormalities from corresponding responses. Due to the lack of ground truth on the expected responses from legitimate services, previous studies typically require a heavy, unscalable manual inspection to identify false positives while still leaving false negatives undetected. In this paper, we propose Disguiser, a novel framework that enables end-to-end measurement to accurately detect the censorship activities and reveal the censor deployment without manual efforts. The core of Disguiser is a control server that replies with a static payload to provide the ground truth of server responses. As such, we send requests from various types of vantage points across the world to our control server, and the censorship activities can be recognized if a vantage point receives a different response. In particular, we design and conduct a cache test to pre-exclude the vantage points that could be interfered by cache proxies along the network path. Then we perform application traceroute towards our control server to explore censors' behaviors and their deployment. With Disguiser, we conduct 58 million measurements from vantage points in 177 countries. We observe 292 thousand censorship activities that block DNS, HTTP, or HTTPS requests inside 122 countries, achieving a 10^{-6} false positive rate and zero false negative rate. Furthermore, Disguiser reveals the censor deployment in 13 countries.

CCS Concepts: • **Networks** → *Network measurement*; *Public Internet*; • **Security and privacy** → *Network security*; • **Social and professional topics** → **Censorship**.

ACM Reference Format:
Lin Jin, Shuai Hao, Haining Wang, and Chase Cotton. 2021. Understanding the Practices of Global Censorship through Accurate, End-to-End Measurements. *Proc. ACM Meas. Anal. Comput. Syst.* 5, 3, Article XXX (December 2021), 25 pages. https://doi.org/10.1145/XXXXXXX

1 INTRODUCTION

Internet censorship controls what can be viewed by a certain group of Internet users. Such information control, typically placed by authority entities such as governments, ISPs, or organizations, can be successfully achieved by various techniques ranging from IP-layer censorship (*e.g.*, blocking IP addresses) to application-layer censorship (*e.g.*, DNS manipulation). Internet censorship has been widely witnessed and its severity varies from country to country.

Significant research efforts [8, 13, 18, 32, 34, 36, 39, 44, 45, 49, 51, 56] have been devoted to measuring Internet censorship. Specifically, to detect censorship activities in a region, the basic idea is to send a request from a vantage point within the region and then compare the response with a valid response from a legitimate server. However, the dilemma here is that if the request is

Authors' addresses: Lin Jin, linjin@udel.edu, University of Delaware, 210 South College Ave., Newark, Delaware, USA, 19716; Shuai Hao, shao@odu.edu, Old Dominion University, 1 Old Dominion University, Norfolk, Virginia, USA, 23529; Haining Wang, hnw@vt.edu, Virginia Tech, 900 N Glebe Rd, Arlington, Virginia, USA, 22203; Chase Cotton, ccotton@udel.edu, University of Delaware, 210 South College Ave., Newark, Delaware, USA, 19716.

© 2021
2476-1249/2021/12-ARTXXX $15.00
https://doi.org/10.1145/XXXXXXX

blocked, the vantage point has no ground truth to identify the valid response. To tackle this issue, existing studies [18, 32, 36, 39, 45] collect the valid responses from nodes deployed in multiple countries. However, this approach inevitably reduces the detection reliability due to the diversity and flexibility of Internet services. For example, clients at diverse locations may obtain different but valid IP addresses for the same domain, and websites may intentionally restrict their services on certain locations or offer different content to the clients from different locations. Thus, manual inspection is usually needed, causing the analysis to be unscalable and inefficient. More importantly, manual analysis can only identify false positives (*i.e.*, misclassified censorship) but false negatives (*i.e.*, undetected censorship) remain uncountable due to the lack of ground truth on what should have been received as mentioned above. As a result, the accuracy and reliability of the detection are still questionable after manual analysis.

One recent technique, Quack [51], addresses such a dilemma with servers running the Echo service that reflects back any bytes sent to it. Thus, each request sent to an echo server inside the censored region is reflected and then the outgoing traffic will encounter the censor. In the meantime, the request itself would also be the expected response if no censorship presents. However, the requests sent to and received from the echo servers are not on standard HTTP/HTTPS ports, and such requests cannot trigger a censor if it only examines requests on standard ports. We observe that many censors in 32 countries, including those enforce severe censorship policies such as Saudi Arabia and UAE, only block the requests sent to standard HTTP/HTTPS ports. To further improve the effectiveness, another work [44] replaces echo servers with genuine web servers by which an error page can be served as the ground truth of server responses since the genuine web servers do not host the test domains exposed in the requests. Unfortunately, such inbound requests may not be inspected by censors [36, 51] as their main goal is typically to control the content being accessed by users within the censored regions, resulting in the censorship undetected. To this end, it is imperative to explore an accurate and efficient methodology for understanding the censorship practices on a global scale.

In this paper, we propose *Disguiser*, a novel framework that accurately detects the censorship and explores the deployment of censors. Disguiser introduces a control server as the destination of all probing requests to provide the ground truth of server responses. In particular, we send requests from vantage points, located in the tested regions, to our control server placed outside the tested region, which will return *static* responses we crafted. As such, by comparing the response obtained from the vantage point with the static response, we can accurately identify the censorship activities. Furthermore, we investigate the censor deployment using application traceroute, by which each vantage point makes a three-way handshake with our control server and then sends the requests with incremented Time-to-Live (TTL) values to identify censor behaviors and determine the location of censors.

Based on Disguiser, we conduct comprehensive and large-scale measurements on the censorship with three fundamental protocols: DNS, HTTP, and HTTPS. Specifically, we acquire vantage points from the SOCKS proxy network and RIPE Atlas to send our probing requests. The experiments are performed in two six-weeks periods in two years (2020 and 2021). During our study, we conduct 58 million measurements from 177 countries and observe 292 thousand censorship activities from vantage points in 122 countries. We find that HTTP-based blocking is the most prevalent censorship activities. Also, by comparing the censorship activities in two years, we observe a significant increase on the number of censored domains in China in 2021, and Russia adopts HTTPS interception in 2021. Furthermore, we use commercial VPN servers to explore the deployment of censors by application traceroute, since only the VPN servers can set the TTL values of the probing requests and observe ICMP packets. We collect available VPN servers and then reveal censor deployment in 13 countries.

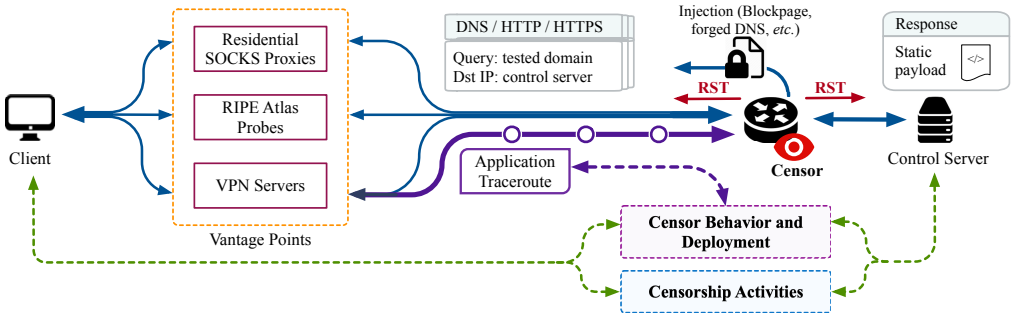

Fig. 1. Architecture of Disguiser Framework. Disguiser conducts censorship measurements through three types of vantage points. The vantage points issue requests with embedded domain names in the test list to our control server that replies with a static response. In particular, VPN servers conduct application traceroute to detect the censor deployment.

Ideally, Disguiser does not generate false positives and false negatives. However, a cache proxy placed between a vantage point and our control server may intercept the connections and inject responses from its local cache or retrieve responses from legitimate servers rather than our control server, resulting in false positives. To minimize such an impact, we first design and conduct a cache test to exclude those cache related vantage points in an injecting-and-probing manner. Also, we apply heuristics to efficiently clean our datasets. Without manual efforts, Disguiser achieves a 10^{-6} false positive rate and zero false negative rate in detecting censorship activities.

The remainder of this paper is organized as follows. Section 2 introduces the background of Internet censorship. In Section 3, we describe our system design. We present detailed observations on censorship policies and behaviors in Section 4, and then we reveal the deployment of censors in Section 5. We present extensive discussions on Disguiser in Section 6. We survey related work in Section 7, and finally, we conclude the paper in Section 8.

2 BACKGROUND

Domain names and IP addresses are the most straightforward and useful information for censors to monitor. As CDNs have been widely adopted [26] where the IP addresses are typically shared with many legitimate services [27], the IP-based blocking may cause significant collateral damage [17, 21, 35, 57]. On the other hand, the domain name based blocking enables the censors to accurately block their undesired Internet services. Here, we briefly describe the application-level censorship that blocks domain names. Then, we present how a censor could be deployed and monitor the traffic.

2.1 Application-level Censorship

Domain names very often are sent in plaintext. This allows a censor to learn the destination resource a client intends to access and block the traffic if the accessed information is prohibited by authorities.

2.1.1 DNS Blocking. When visiting a website, a client first resolves its domain name obtain the network address by using DNS. Since DNS was originally designed as an unencrypted protocol, the censors on the network paths are able to manipulate the DNS responses or drop the DNS queries. Although UDP-based DNS is mostly adopted, TCP-based DNS is also inherently supported. For a TCP-based DNS request, a censor may tear down the connection with RST/FIN packets.

2.1.2 Domain Names Blocking in HTTP. The HTTP `Host` header presents the domain name a client is visiting, specifying the target service since a web server may host multiple domains. Still, the HTTP protocol is unencrypted and the censors can know exactly the requested domain. To block an HTTP connection if needed, a censor may inject a *blockpage* indicating that the domain is prohibited, tear down the connection with RST/FIN packets, or directly drop the request without any notification.

2.1.3 Domain Names Blocking in HTTPS. HTTPS encrypts all the HTTP packets after a TLS handshake so that the `Host` header is no longer visible to the censors. However, it is common that a web server hosts multiple domains, and each domain is associated with an independent certificate. Therefore, in order to present the correct certificate to a client before the ephemeral keys are exchanged, an SNI extension is required to be included in the `Client Hello` message to indicate which domain the client intends to visit. As a result, the domain name in the SNI extension is sent in plaintext and is visible to the censors. As such, to block an HTTPS connection, a censor may tear down the connection with RST/FIN packets, directly drop the packet, or inject an incorrect, forged certificate to intercept the connection.

2.2 On-path and In-path Censors

In order to examine network traffic, censorship devices can be deployed in two different ways. An on-path censor is the device attached to a router and can obtain a copy of all the packets passing through the router. Since it cannot operate on the original packets, it is unable to prevent the packets from reaching their destinations. Correspondingly, it needs to inject packets to interfere or terminate a connection.

An in-path censor acts as a Man-In-The-Middle to examine the actual packets. Therefore, it can directly manipulate or drop the packets associated with the prohibited services. The in-path device is usually hard to be identified; however, due to the capacity of operating on the actual packets, it can be efficiently detected by the Disguiser's control server.

3 SYSTEM DESIGN

Disguiser is an end-to-end measurement framework for *accurately* investigating the practices of global Internet censorship that is based on either DNS or HTTP/HTTPS. Figure 1 illustrates the system design of Disguiser. The objective of Disguiser is to detect censorship activities and explore the censor deployment, while effectively eliminating false negatives[1] and minimizing false positives without manual inspection. The high-level idea is that a client instructs the vantage points to (1) craft DNS/HTTP/HTTPS requests with the test domain names embedded, (2) send the packets to our control server to trigger censorship, and (3) collect the response back for later analysis. On the other side, our control server replies to arbitrary requests with a static payload for each type of protocol. To identify the location of censors and examine their deployment, application traceroute is performed in which the packets with increased TTL values are repeatedly sent for encountering the censors. Importantly, to eliminate the noise data, we carefully design tests to exclude the vantage points which could be potentially affected by the proxies/middleboxes placed in the network path. A detailed comparison of detection capabilities and accuracy between Disguiser and existing censorship measurement systems will be presented in Section 6.2.

[1]Note that the elimination of false negatives means that, for each request we sent, we will not falsely classify it as not censored. However, for the cases that we do not test (*e.g.*, requests for domains that are not in our test list), we cannot tell if censorship would occur or not. Therefore, even if all the requests we sent from a vantage point are classified as negative, it does not mean that the vantage point does not suffer censorship.

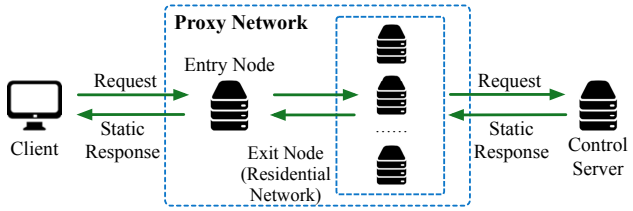

Fig. 2. Residential SOCKS Proxy Network.

Table 1. Coverage of Vantage Points. (U-/T-DNS: UDP-based/TCP-based DNS)

	SOCKS Proxies			RIPE Atlas
	T-DNS	HTTP	HTTPS	U-DNS
Number of VPs	9,470	9,383	8,599	1,400
Number of Countries	176	177	172	136
Median of Countries	39	31	28.5	4
Number of ASes	1,418	1,402	1,339	726

3.1 Vantage Points

In order to carry out extensive experiments, we need to use a large number of vantage points distributed across the world to issue different types of queries, *i.e.*, TCP- and UDP-based DNS, HTTP, and HTTPS. To satisfy those requirements, we leverage multiple types of platforms, including the residential SOCKS proxies, RIPE Atlas, and VPNs, to acquire vantage points and complete our experiments.

3.1.1 SOCKS Proxies. SOCKS proxies allow us to proxy TCP-based queries to any IP address. Figure 2 illustrates a typical residential proxy network [33] and how it fits our design. The entry node receives our test requests and forwards the requests to exit nodes distributed across the world. The exit nodes then serve as our vantage points which will be responsible to send requests to our control server and relay the response back to our client through the entry node. In our study, we issue TCP-based DNS queries and HTTP/HTTPS queries through the SOCKS proxies. For the ethical considerations, instead of using the open SOCKS proxies, we subscribe to managed and paid proxy services from ProxyRack [40] with a cost of $120 per month. ProxyRack provides widely distributed SOCKS proxies by recruiting a large number of hosts that join the platform at their will.

Note that since a SOCKS proxy works above the transport layer, it would not return the information of IP layer back to our client. Thus, the SOCKS proxies are not suitable for application traceroute that requires to recognize ICMP packets at client side. Also, although the SOCKS protocol itself can support UDP packets, we are not able to find a managed service that implements it on its proxies. Therefore, we do not conduct UDP-based DNS tests on SOCKS proxies.

3.1.2 RIPE Atlas. RIPE Atlas [46] is a global Internet measurement platform built by RIPE NCC. RIPE Atlas probes are mostly hosted by volunteers who willingly join the platform and earn credits for running their own experiments. In particular, RIPE Atlas enables many different types of measurements from the probes in over 150 countries, including ping, traceroute, DNS, NTP, SSL, and HTTP, with the support of parameter control. However, due to the limited control of HTTP measurement, we were unable to tune the HTTP packets with increased TTL values and the destination of our control server. In our study, we use RIPE Atlas to conduct UDP-based DNS tests to complement the results of TCP-based measurement from SOCKS proxies.

Table 1 summarizes the protocols and corresponding platforms of vantage points. The details on their coverage are described in Section 4.1.

3.1.3 VPNs. Virtual Private Network servers (VPNs) allow us to gather richer network information from IP layer for enabling the application traceroute. However, in comparison to the widespread SOCKS proxies and RIPE Atlas probes, the availability of reliable VPN vantage points in certain countries is relatively limited. In this study, we use VPN vantage points to conduct the application traceroute to investigate the deployment of censors.

3.2 Censorship Detection

The key component of our framework is a Disguiser server under our control that would be specified as the destination for all the requests. As a result, we do not send any requests to legitimate servers, and the accessed domains in the requests (if being censored) would still trigger the censorship since the censor devices will see the requested domains but have no knowledge whether the destination IP address is associated with a legitimate server of the censored domain. In the end, this provides us a baseline by controlling what should be expected at the client side when no censorship is involved so as to accurately recognize the censorship activities.

With regard to the DNS experiments, our control server will reply to the requests with a static and reserved IP address that has never been used in manipulated DNS responses by any censors in different countries.[2] For the HTTP test, the control server will reply to the requests with a customized, static webpage that would be unique to any other webpages. Finally, for the HTTPS test, our control server will present a self-signed certificate. As such, if our client receives a different certificate, it will further fetch the webpage of a corresponding domain to verify whether such an interception is caused by censorship. Otherwise, the client will terminate the connection and conclude that there is no HTTPS-based censorship. Likewise, our control server will reply with the same static webpage described in the HTTP test.

For each query we sent, we wait 15 seconds before we terminate the connection. Taking network congestion into consideration, we then retry the queries at most four times. If all retries timeout, we consider that the tested domain is being blocked. In particular, due to the high churn of residential SOCKS proxies, between each retry, we conduct a proxy-alive check by sending a request with an uncensored domain to our control server to make sure the proxy is online.

Due to the design above, the way that we detect the censorship is self-evident, by which we can directly mark the DNS and HTTPS responses as being censored if they are inconsistent with our static payload. For each HTTP response, we search our static payload from the response and mark it as being censored if it does not contain our payload, which also eliminates the potential impact of ad injections by ISPs.[3] In UDP-based DNS experiments, the expected censorship behavior is either the injection of a forged DNS record or no response. In other TCP-based experiments, the expected censorship behavior is an injection, connection teardown, or no response.

3.3 Minimizing Cache Proxy Impact

As stated above, our framework includes a control server to provide a ground truth of responses to significantly facilitate the censorship analysis. However, ISPs may deploy cache proxies as a part of network infrastructures so that resources can be reused, reducing network traffic and latency. Nevertheless, those cache proxies may introduce interference for our measurements. Specifically, when a cache proxy is deployed, it may directly respond to us if it already has a copy of the content of the requested domains. Also, the cache proxy may intercept the connection between one of our vantage points and the control server, perform its own DNS resolution, and issue the request to

[2]We collect and exclude the private IP blocks containing the addresses used in DNS manipulation, which have been identified in prior study [39].

[3]An ad injector may inject ads into the responses, but typically it would not alter our original payload. We actually have not observed such ad injections in our dataset.

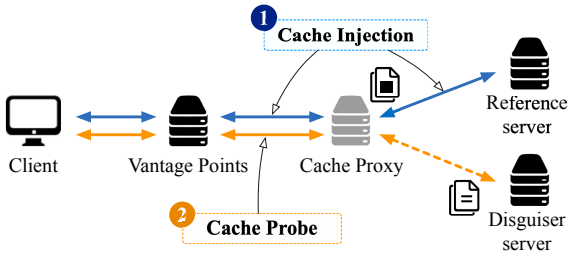

Fig. 3. Workflow of Cache Test (DNS & HTTP).

the legitimate server of the corresponding domain. As a result, our client will obtain a response different from the static payload provided by our control server, and our system may incorrectly determine that such a response is injected by censors, resulting in false positives.

3.3.1 Cache Detection and Elimination. In order to minimize the impacts of cache proxies, we conduct a cache test for each vantage point before we conduct the measurement, as depicted in Figure 3. First, we set up a reference system consisting of an authoritative DNS server for a domain under our control and a web server that hosts the domain. The response of the authoritative DNS server specifies the web server and the web server hosts a landing page different from the static webpage provided by the Disguiser's control server.

Next, we detect the presence of DNS/HTTP proxies by an injecting-and-probing manner as follows. (1) The vantage point resolves our domain name from the authoritative nameserver. (2) With the nameserver's response, the vantage point fetches the landing page of our domain from our legitimate web server. (3) If any type of cache is present, the requests in (1) and (2) would encounter a DNS or HTTP cache proxy, and corresponding responses would be cached in those invisible proxies. (4) After that, the vantage point resolves our domain and fetches the landing page from the Disguiser's control servers that provide different responses. As such, if a vantage point receives any responses associated with the reference system when querying Disguiser, we consider that a cache proxy is in effect, and hence we drop the vantage point without conducting the measurement.

3.3.2 Data Cleaning. Although the cache test is efficient to identify most cache proxies, some unusual behaviors may cause the cache proxies to remain undetected. First, the proxies may only cache popular content but ignore the resources of our domain. Also, cache proxies may selectively intercept some connections so that our cache test does not trigger such an interception. Therefore, in order to improve the detection reliability, we further examine the collected data and exclude the vantage points that are impacted by the cache proxies with unusual behaviors above. Note that the vantage points we need to examine are a small portion of the ones that observe (1) public IP addresses of measured domains or (2) webpages different from the static payload of any domain.

DNS Dataset Cleaning. To identify the vantage points impacted by DNS caches, we apply the following heuristics to the domains that Disguiser can obtain public IP addresses.

First, we resolve those domain names locally to obtain their valid IP addresses. If an IP address matches the address obtained from a vantage point, we exclude the vantage point in the later analysis since it indicates that the vantage point received a valid address through DNS cache proxy. Second, for the rest of the domains, we retrieve their landing pages using the public IP addresses obtained by Disguiser. In the meantime, we retrieve their landing pages locally. If, for any domain, the two landing pages present the same, non-empty `<title>` tag, we conclude that the vantage

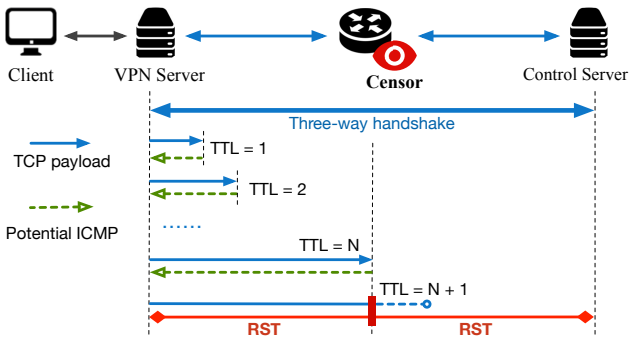

Fig. 4. Application Traceroute.

point has been impacted by a cache and contributes false positives, and we exclude such vantage points in the later analysis.

HTTP Dataset Cleaning. Similar to the cases in DNS, for the domain names with a webpage different from our static payload, we retrieve their landing pages locally and compare the `<title>` tags. We exclude the associated vantage points if the `<title>` tags are identical and non-empty.

3.4 Censor Deployment Detection

In addition to accurately detecting the censorship activities, Disguiser further explores the deployment of censors using application traceroute, which is illustrated in Figure 4. As we mentioned earlier, we use VPN servers as our vantage points to conduct application traceroute. Particularly, the vantage point first completes a TCP three-way handshake with our control server to establish a connection. Then, it increases the TTL of the request that contains a censored domain name. As the TTL increases, we should receive Time Exceeded ICMP packets from routers on the path. We stop sending the requests when we receive a sign of censorship that could be an injected packet or the TTL reaching its maximum number of 64.

Assuming that the censor's router is N hops away from vantage points and the censor uses its own default TTL value in its injected packet, we then should observe a sign of censorship only when TTL is set to $N+1$ or larger; otherwise, the packet will be dropped before reaching the censor router (TTL$< N$) or without being processed when it reaches the censor router (TTL$= N$).

However, we discover that the censor may directly copy the TTL value of our original packets to its injected packets. As a result, the injected packets may expire on their way to our vantage point or control server if the TTL value of our original packet is not large enough. Only when the TTL of original packets is set to $2N$ or larger, should the vantage point observe the injected packets. The discussion of the censor's TTL behaviors is detailed in Section 5.

In the case that the censor router simply drops the original packets without injecting any additional packets such as RST, application traceroute would stop at its maximum TTL value. If this happens, we can see that, after certain hops, we no longer receive any ICMP responses. Then, we run a normal traceroute to our control server to find out if we can observe the IP address of the router being one hop further downstream. If so, we conclude that the packet dropping is caused by the active censorship at the last router we encountered in application traceroute.

3.5 Test Domain Names

To study Internet censorship on a global scale, we need a list of domains that are potentially being censored in different countries. In doing so, we leverage the standard approach used in prior studies

[36, 39, 51], *i.e.*, compiling a test domain list that consists of popular domains and sensitive domains. We collect Alexa's top 1,000 domains [3] as the popular domain list. For sensitive domains, we resort to the widely used test lists provided by Citizen Lab [14]. The Citizen Lab offers two types of test lists, a global test list and a country-specific test list for certain counties. Correspondingly, we compile the country-specific test list with the popular list and global test list to form the domain list for each country. Note that some domains in the lists may be expired or may not be in service. Therefore, we retrieve the landing page of the domains and exclude the ones if the response status code is not `200 OK`. As such, the number of domain names we prepared for vantage points in different countries ranges from 1,908 to 3,953. In addition, we determine the content type of each test domain using the classification services provided by FortiGuard [19].

3.6 Ethical Considerations

Since our study does not involve the collection of personal information or human participation, it falls outside the purview of IRBs [29]. On the other hand, censorship studies still pose ethical concerns due to active, large-scale experiments. We here discuss our experiment design for reducing the potential risk and real-world impact.

Our system does not send queries to an actual server of legitimate websites, and thus there will be no direct connections between vantage points and sensitive domains, which significantly reduces the risk of participants. Moreover, we explicitly state the purpose of our requests and leave our contact information in the static payload of the response. During the entire period of our study, we did not receive any complaints regarding the experiments.

Also, as mentioned above, ISPs may deploy cache proxies as a part of network infrastructures to serve their users so that popular resources can be reused, reducing network traffic and latency. However, we perform comprehensive cache tests to actively exclude those cache related cases from our measurement. Further, to minimize the accidental cache impacts that we are not aware of, our DNS responses have the TTL set to 1 for an arbitrary query.[4] For HTTP responses, the Disguiser's control server sets the `Cache-Control` header to `no-cache, no-store, private, max-age=0` so that the cache proxy cannot store our responses in their cache. Even if a cache proxy violates our configuration, the cached content (*i.e.*, our static payload) can be considered harmless and the cache proxy will still be able to re-validate the content from the corresponding legitimate website.

Finally, in our study, we utilize different types of vantage points to conduct our experiments. ProxyRack is a paid residential proxy service that requires opt-in from their participants who join the business at will for profits [41], and they can opt-out anytime. Such a proxy network has a large number of exit nodes across the world, and an exit node is randomly assigned for us. During our experiments, we do not repeatedly use one node to further reduce the potential risk of each individual participant. Also, we carefully examine and follow the RIPE Atlas's ethical guidelines [47]. In particular, DNS requests have been considered as innocuous for censorship measurement [10]. An extensive censorship measurement [5] conducted over the RIPE Atlas has also not yet detected any authority that is attracted by the measurement traffic. Lastly, VPN services have been widely used to bypass the Internet censorship, and hence the VPN operators understand the risks of deploying servers in a country [36].

4 CENSORSHIP CHARACTERIZATION

4.1 Measurement and Dataset

[4]Note that the behavior of resolvers for the DNS records with the TTL of 0 is not well defined (*e.g.*, it could be rejected or interpreted as a maximum value [1, 2]), and hence we set the value to 1.

Table 2. DNS Censorship Statistics. For TCP-based DNS, we show the top-5 countries that block the tested domains. Categories column lists the top three categories of censored domains within a country. The abbreviations are inspired from ICLab [36] and we made some adjustments. The abbreviations of domain categories are specified in Appendix B.

Protocol	Country	Perc.	Categories
U-DNS	China	19.2%	NEWS, SRCH, PROX
	Iran	12.6%	SHOP, PROX, PORN
T-DNS	China	20.1%	NEWS, SRCH, PROX
	Iran	16.5%	BLOG, NEWS, PORN
	Egypt	7.3%	NEWS, PORN, ILLE
	Turkey	5.7%	PORN, GAMB, PROX
	Colombia	5.1%	INFO, ORGA, SHOP

Table 3. DNS Censorship Techniques.

Protocol	Censorship Techniques	Percentage
U-DNS	Forged Record	100%
T-DNS	Forged Record	0.87%
	SERVFAIL	0.009%
	REFUSED	0.004%
	Connection Teardown	96.19%
	Timeout	2.93%

We conducted a global scale measurement study with Disguiser over two six-weeks periods, one from April 2020 to May 2020, and another from June 2021 to July 2021. In total, we conduct 58 million measurements from vantage points in 177 countries. In particular, we conducted the experiment once a week through SOCKS proxies. For each country, we selected up to 15 vantage points to conduct measurement for each protocol. Also, we conducted a one-time measurement with RIPE Atlas probes. We first randomly select and test 1,000 prepared domains for each country and the probes are randomly selected by RIPE Atlas within the country. If we recognize any censorship activities in a country, we then test all the domain names in the list prepared for that country. The SOCKS and RIPE Atlas vantage points used in Disguiser is listed in Table 1. For each TCP-based protocol, we conducted measurements from roughly 9,000 SOCKS proxies distributed in more than 170 countries and issued UDP-based DNS queries from 1,400 RIPE Atlas probes located in 136 countries. During our experiments, we found that censors in China block the address pair of the vantage point and our control server for roughly 90 seconds when an HTTP/HTTPS request triggers a censor. As a result, we need to add a delay when the censor is triggered. Since we cannot hold the SOCKS proxies for a very long time, we instead used the VPN servers (Section 5) for the HTTP/HTTPS tests in China.

In total, we identify 292,852 censorship activities in 122 countries and achieve a 10^{-6} false positive rate and zero false negative rate (detailed in Section 6). Based on our observations, we present a censorship map in Appendix A, visualizing the severity of censorship in each tested country. Note that the authorities that enforce the censorship policies could be organizations, ISPs, or governments. For easy presentation, we present our analysis results at the country level. Table 14 in Appendix C lists the fractions of vantage points that observe censorship in different countries.

4.2 DNS (UDP and TCP)

After excluding the cache-impacted vantage points, our DNS dataset consists of 19,422,558 DNS responses, among which 54,691 are being manipulated. Here we present the details of our observations for the DNS-based censorship.

4.2.1 Censorship Detection. Table 2 compares the censorship detection results between UDP-based and TCP-based DNS protocol. China and Iran block both UDP- and TCP-based DNS queries and the numbers of censored domains are all ranked top-2. Further, China's DNS censorship policy is mostly consistent, where the censored domains in UDP- and TCP-based queries are almost the same. One interesting observation is that only 1 out of 65 vantage point from Iran observes TCP-based DNS censorship, indicating that Iran does not enforce a nation-wide TCP-based DNS censorship.

Table 4. HTTP and HTTPS Censorship Statistics. We show the top-5 countries that block the tested domains.

Protocol	Country	Percentage	Categories
HTTP	Iran	41.0%	NEWS, BLOG, PORN
	China	22.1%	NEWS, SRCH, PROX
	UAE	19.1%	NEWS, PROX, PORN
	Saudi Arabia	15.5%	NEWS, PORN, PROX
	France	15.2%	PROX, GAMB, PORN
HTTPS	Iran	37.2%	NEWS, BLOG, PORN
	China	24.1%	NEWS, SRCH, INFO
	UAE	17.9%	NEWS, PROX, PORN
	Saudi Arabia	16.4%	NEWS, PORN, PROX
	Israel	14.1%	PROX, GAMB, PORN

However, its UDP-based DNS censorship are observed by all vantage points. As a result, TCP-based DNS can be leveraged to circumvent its DNS-based censorship.

China and Iran are the only two countries, we observed, that block UDP-based DNS queries. However, when it comes to TCP-based DNS, the censorship activities we recognized are slightly more prevalent, even though we only observe 5 countries that block more than 5% of the tested domain names. We infer that the countries that only block TCP-based DNS may not intentionally block the DNS queries; instead, the censors inspect the TCP traffic and then block the connections with undesired domain names presented. Overall, we conclude that the DNS-based censorship is not prevalent across the world.

4.2.2 Censorship Techniques. Table 3 shows the identified censorship techniques used to block DNS queries, and we observed a significant difference between blocking UDP- and TCP-based DNS. In particular, censors in Iran and China all block UDP-based DNS by injecting a forged DNS response. The difference is that Iran's censors inject private IP addresses,[5] while China's censors inject public IP addresses. Those public IP addresses we collected are distributed in 17 ASes, and 68% of them belong to a Facebook's AS (AS32934). By contrast, only 0.87% of censored TCP-based DNS queries are blocked using forged responses, and the forged responses contain loopback addresses from 127.0.0.0/8 and empty records. Instead, 96.19% of censored queries are blocked by a TCP connection teardown. Also, 2.93% of censored queries are timeouts, indicating that the DNS queries are dropped by censors without any responses, and the majority of the timeouts occur in United Arab Emirates (UAE). Additionally, we observe a handful of SERVFAIL and REFUSED error messages but do not see the ones with NXDOMAIN.

Although directly dropping the packets may be the most straightforward way to block the DNS queries, it requires the censors to be in-path by which they operate on the actual packets. Moreover, it may trigger the retransmissions from clients, increasing the traffic load of censors. As such, to block TCP-based DNS, most censors prefer to tear down the TCP connection; while to block UDP-based DNS, censors intentionally inject a forged DNS response.

4.3 HTTP and HTTPS

After excluding the cache-impacted vantage points, our dataset consists of 18,702,111 HTTP responses and 17,988,634 HTTPS responses, among which 157,731 and 80,430 of the requests are being censored, respectively. In comparison to the DNS-based censorship, countries are significantly more aggressive in blocking the domain names with HTTP/HTTPS requests.

[5]Note that the private IP addresses are domestically routable in Iran's national network [4], and those injected addresses usually point to web servers that present blockpages.

Table 5. Common Censorship Techniques used by HTTP and HTTPS.

Protocol	Censorship Techniques	Percentage
HTTP	Blockpage	52.03%
	Connection Teardown	38.47%
	Timeout	9.50%
HTTPS	HTTPS Interception	1.76%
	Connection Teardown	82.13%
	Timeout	16.11%

Table 6. HTTPS Interception.

Certificate Issuer	Country	Number of VPs	Number of Domains
Megafon	Russia	5	182
COMODO	Singapore	2	1
DigiCert	Singapore	2	1
Fortinet	Israel	1	268
-	Russia	1	145
-	Nepal	1	142
Everythink	Canada	1	42

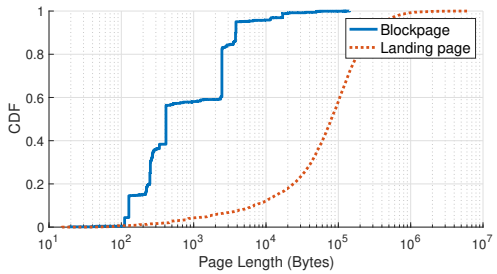

Fig. 5. CDF of Page Lengths.

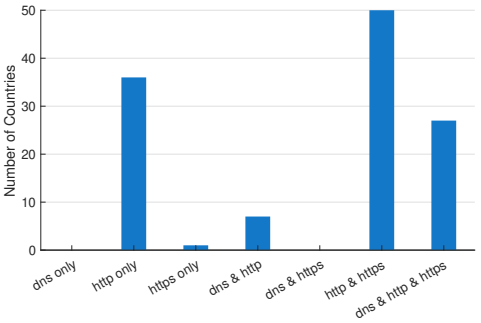

Fig. 6. Censorship by Protocols.

4.3.1 Censorship Detection. Table 4 lists the top countries that fulfill the censorship based on the domain names in HTTP or HTTPS, sorted by the percentage of the censored domains. Iran leads the first place for the two protocols, and Saudi Arabia, UAE, and China all block more than 15% of the tested domains. Also, the top three categories of the censored domains are almost the same for the two protocols, and the percentages are close as well. We found that France, a country that is well known for having an open Internet policy, blocks 15.2% of the tested domains. While we inspected our data, we found that only 2 out of 159 vantage points observed abnormally high censorship activities. Therefore, we infer that the censorship observation of France may be the result of triggering local or organizational censorship products.

Although the top-ranked countries treat HTTP and HTTPS roughly the same, we notice that for the long-tail countries, the HTTP censorship is much more prevalent (see Section 4.5) and the total number of censored HTTP requests is almost twice that of HTTPS requests.

4.3.2 Censorship Techniques. Table 5 presents the observed censorship techniques used to block HTTP and HTTPS requests. For the HTTP protocol, 52.03% of censored requests receive a blockpage, which has been widely used for the indication of the accessed domain being prohibited. Connection teardown and timeout are responsible for 38.47% and 9.50% of censored requests, respectively. We observe blockpages in all the countries listed in Table 4 except China that blocks all HTTP requests by tearing down the TCP connections. For the HTTPS protocol, connection teardown dominates in censorship techniques and is responsible for 82.13% of censored requests. Timeout accounts for 16.11% of censored requests, while 1.76% of censored requests are blocked by HTTPS interception.

4.3.3 Blockpage. Based on the ground truth of static payload in the control server, we can accurately and efficiently identify the blockpages used by censors. In total, we observe 82,060 blockpages during our experiments. Figure 5 plots the CDF of the lengths of blockpages, as well as that of landing pages of the tested domains that we retrieved separately. We can see that the lengths of

Table 7. Censorship Changes for 5 Countries that Censor Domains the Most. Difference is the number of increased(+) and decreased(-) domains divided by the number of tested domains.

Country	Censored Domains in 2020	Censored Domains in 2021	Difference
Iran	39.2%	40.6%	+0.7% / -0.4%
China	23.0%	34.9%	+13.0% / -1.0%
UAE	21.8%	13.8%	+0.6% / -8.5%
Saudi Arabia	15.7%	14.8%	+0.9% / -1.7%
Ukraine	11.8%	10.2%	+4.5% / -6.1%

Table 8. Policy Consistency.

DNS - HTTP		DNS - HTTPS		HTTP - HTTPS		DNS - HTTP - HTTPS	
Country	Consistency	Country	Consistency	Country	Consistency	Country	Consistency
Egypt	84.9%	Egypt	90.3 %	Afghanistan	100%	Egypt	84.5%
Turkey	83.6%	Turkey	86.9%	UK	98.4%	Turkey	79.7%
China	50.7%	Iran	40.1%	Uzbekistan	98.3%	Iran	37.5%
Iran	40.2%	China	39.7%	Nepal	97.9%	China	26.7%
UAE	9.5%	UAE	9.9%	Israel	95.7%	UAE	8.1%

blockpages are saliently shorter than those of legitimate landing pages and are concentrated in certain values. Specifically, the majority of the blockpages are shorter than 10^4 bytes, and roughly 60% of the blockpages are less than 10^3 bytes. We observe that the shortest blockpage is from Iran with the length of 19 bytes and the longest blockpage is from Vietnam with the length of 142,667 bytes. By contrast, the smooth curve for the landing page's length CDF shows that the lengths of landing pages are more scattered, and they are much longer than those of the blockpages (*e.g.*, about 60% of the landing pages are longer than the 99^{th} blockpage). Note that, although the distribution of page lengths presents the salient difference between blockpages and landing pages, they are still overlapped to some extent; without the ground truth enabled in Disguiser, it would still fall short of being used as a dominant feature for accurately identifying the blockpages.

4.3.4 HTTPS Interception. In our study, we observe 1,415 HTTPS interceptions from 27 vantage points in 14 countries and collect 30 certificates. Table 6 lists the information of 7 certificates that intercept at least 2 requests. We can see that 5 of the vantage points observe a same certificate issued by Megafon, and those vantage points are located in 3 ISPs. In addition, for the two vantage points that observe the COMODO certificate, they also observe a different certificate from DigiCert when visiting another domain. Furthermore, we observe another four certificates that are used to block a significant number of domains from Israel, Russia, Nepal, and Canada. Note that all of the certificates listed in Table 6 are observed in our 2021 experiment periods. Overall, HTTPS interception is still rarely used to block domains, but Russia starts to adopt this technique to enforce its censorship policies.

4.4 Longitudinal Analysis

Collecting data from two periods in two years allow us to observe changes on the censorship landscape. Table 7 shows the observed changes on censored domains from 5 countries that censor domains the most. The most significant change is that China increases 13% of the tested domains in 2021. It now censors roughly 50% more domains than it did in 2020. The top 3 categories of the newly blocked domains are news and media, information technology, and general organizations. In addition, the majority of the newly blocked domains are blocked when HTTP/HTTPS are used. By contrast, the most significant drop (8.5%) of censored domains is observed in UAE. Iran and Saudi

Arabia do not modify their censorship list too much. Although vantage points in Ukraine observe a non-trivial increase and decrease on the censored domains, we are not able to draw any conclusion from the change because the fraction of vantage points that observe censorship is relative low (roughly 10%), and so the change could be the result of triggering different censors.

4.5 Censorship Policy Consistency

Figure 6 plots the number of countries that are observed to monitor different types of protocols. In particular, we identify that 36 countries block HTTP requests only. Syria is the sole country that blocks HTTPS requests only, and only one specific domain is blocked. We confirm that this is not a false positive because we observe multiple vantage points in Syria block the HTTPS traffic of this domain at different times. We do not observe any country that blocks DNS queries only. Furthermore, blocking both HTTP and HTTPS requests is the most prevalent scenario (50 countries), and we observe 27 countries that block all three protocols.

In addition, censors may not consistently enforce their policies across different protocols. As a result, this can be leveraged to circumvent the censorship. For example, if a DNS request for a domain is blocked but the HTTP/HTTPS requests are not blocked, then an encrypted DNS resolver may help circumvent the censorship. Therefore, we explore whether the censorship policy in each country is consistent across different protocols. Here, we define a policy consistency C_{PROT} for the three protocols such that:

$$C_{PROT} = \frac{|D_{dns} \cap D_{http} \cap D_{https}|}{|D_{dns} \cup D_{http} \cup D_{https}|}, \tag{1}$$

where $D_{[protocol]}$ is the set of domains that experience censorship when the corresponding protocol is used, and $|\cdot|$ is the cardinality of the set. Similarly, when calculating the policy consistency for two protocols, we remove a domain set in both numerator and denominator of Equation 1. Also, we determine that a domain experiences DNS censorship in a country if the domain is blocked in that country with either UDP- or TCP-based DNS, *i.e.*, $D_{dns} = D_{u\_dns} \cup D_{t\_dns}$, where D_{u_dns} and D_{t_dns} are the sets of domains being censored when UDP- and TCP-based DNS are used, respectively.

Table 8 lists the top 5 countries in each category of policy consistency. Overall, Egypt and Turkey exhibit a high policy consistency when all protocols are examined. Many countries, such as Afghanistan, UK, and Uzbekistan, show a significantly high policy consistency between HTTP and HTTPS protocols. Interestingly, although Russia is the country that has been considered of having a decentralized censorship control [45], it also shows a relatively high policy consistency (81.6%) in blocking HTTP and HTTPS protocols, and Russia does not block our DNS queries. Nevertheless, other than Egypt and Turkey, countries show a relatively lower policy consistency between DNS and HTTP(S) than that between HTTP and HTTPS protocols. This is mainly because most countries do not fulfill a comprehensive DNS-based censorship policy.

4.6 Commonly Censored Domains

The commonly censored domains, ordered by the number of countries that censor some domains, are listed in Table 9. The most commonly censored domains are `www.xvideos.com`, a pornography website that being censored by 51 countries. As we can see, the top 7 censored websites all belong to pornography category, and they are followed by two proxy avoidance domains and 1 domain that hosts malicious content.

Table 10 highlights the categories that are mostly censored among countries. Not surprisingly, pornography and proxy avoidance are the top two categories, and the related domains are censored by 76 and 71 countries, respectively. Moreover, comparing with the results in Table 9 where

Table 9. Top 10 Commonly Censored Domains.

Domain	Category	Number of Countries
www.xvideos.com	Pornography	51
www.pornhub.com	Pornography	47
www.xnxx.com	Pornography	47
www.youporn.com	Pornography	45
www.redtube.com	Pornography	43
www.gotgayporn.com	Pornography	40
www.beeg.com	Pornography	40
secure.proxpn.com	Proxy Avoidance	37
anonymouse.org	Proxy Avoidance	37
www.fadskis.com	Malicious Websites	34

Table 10. Top 10 Commonly Censored Categories.

Category	Number of Countries	Censored Domains†
Pornography	76	166 / 390 (42.6%)
Proxy Avoidance	71	78 / 274 (28.5%)
Information Technology	65	198 / 857 (23.1%)
News and Media	60	778 / 4,227 (18.4%)
Illegal or Unethical	50	56 / 110 (50.9%)
Gambling	50	72 / 235 (30.6%)
Business	48	208 / 1,352 (15.4%)
Streaming Media/Download	43	76 / 268 (28.4%)
Search Engines and Portals	42	117 / 398 (29.4%)
Other Adult Materials	39	35 / 93 (37.6%)

† The column shows statistics as the format of "number of censored domains / number of total domains (percentage)".

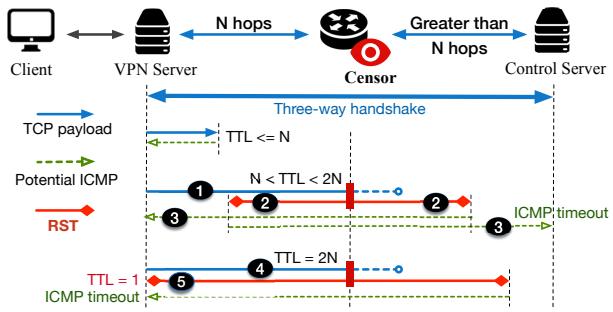

Fig. 7. Application Traceroute When Censors Copy TTL.

at most there are 51 countries censoring one single domain, we observe that countries usually selectively block domains within a category, instead of compiling a comprehensive list covering as many domains as possible from a specific category. Also, among the top 10 censored categories, the category of illegal or unethical exhibits the greatest fraction of censored domains, while the category of business shows the smallest.

5 CENSOR DEPLOYMENT

In this section, we describe the experiments exploring censor behavior and deployment. To do so, we first sign up VPN services and conduct censorship measurement to identify whether a VPN server experiences any censorship. If identified, we then perform the application traceroute towards our control server to discover the censor's location.

5.1 Measurement

There are two additional requirements for a VPN server to carry out such an experiment. First, the VPN server and its default gateway should not alter the TTL values of our packets so that the intermediate routers can process the packets properly according to the TTL values we set. Second, the VPN server must be physically located in the country as advertised. The VPN providers have been recognized to often lie about their servers' locations [53]. Therefore, we need to validate the location of the VPN server when performing the application traceroute. In doing so, we collect (1) the nationality of intermediate routers based on both IP geolocation and AS information, and (2) the

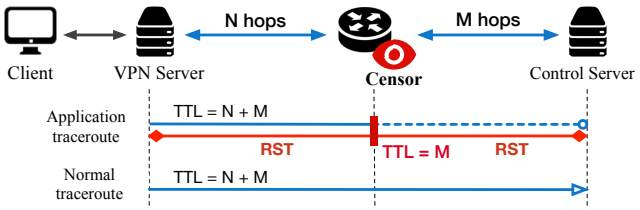

Fig. 8. Validation of Copying TTL.

hostnames of routers by reverse DNS lookup to search for any geolocation hints (*e.g.*, country code, city, or airport code) from one hostname. If any of the geolocation information confirms the country of the VPN server as advertised, we consider its location being validated. We discard the VPN servers that we cannot confirm their correct locations. Consequently, we sign up 18 popular VPN service providers and in total identify 36 VPN servers in 13 countries that experience censorship and satisfy the above two requirements.

As we mentioned in Section 3.4, we discover that some censors may copy the TTL values of the censored packets to their injected packets. Here, we describe the network trace of application traceroute to detect such a behavior, as illustrated in Figure 7. First, when the initial TTL of packets is less than or equal to N, the VPN server receives ICMP packets as expected. Then, as we keep increasing the TTL, our requests start to trigger the censor. Specifically, if the initial TTL of our packets is in the range of $(N, 2N)$, the TTL of the packets will be in the range of $(0, N)$ when they reach and trigger the censor (❶). At that time, if the censor copies the TTL values of the received packets to its injected packets (*e.g.*, RST packets), the injected packets will expire on their way to our VPN server and control server (❷), assuming the distance between the censor and control server is greater than N hops. As a result, the intermediate routers between the VPN server and censor, as well as those between the censor and control server, should receive expired packets and send ICMP packets to our control server and the VPN server, respectively (❸). Next, when the TTL of packets increases to $2N$ (❹), the TTL of packets will be decremented to N when triggering the censor. Thus, the TTL value of injected packets sent back to the VPN server will also be N as the censor copies the TTL. In this scenario, the VPN server will observe injected packets and their TTL has been decremented to 1 (❺).

As such, if we observe that the TTL of the first injected packet is 1 in application traceroute, it strongly implies that the censor copies the TTL values of the censored packets to its injected packets. However, there are still other cases that may produce a coincidence: (1) a censor may set a small TTL value that happens to be N in its injected packets, or (2) a censor may intentionally add a number to the TTL values of received packets as the new TTL of its injected packets. Both cases may lead to an injected packet with the TTL of 1 observed at the VPN server.

To further validate the behavior of copying TTL, we design and conduct two additional experiments. First, after we observe the first injected packet whose TTL equals to 1, we keep increasing the TTL values of the packets in the traceroute; if the TTL of the injected packets being received afterwards also increases, then we can exclude the case (1). Second, to exclude the case (2), we increase the TTL of the packets until our control server receives an injected packet, as shown in Figure 8. Assuming that the censor directly copies the TTL without adding a number, the TTL that we obtained in application traceroute should be the same as that in a normal traceroute towards our control server. Otherwise, these two TTLs should be different. These additional experiments further verify the existence of copying TTL behaviors when some censors intend to terminate the connections by injected packets.

Table 11. Censor Information

Country	Censor Router[◊]	ASN	ISP	AS Rank	Hops to Border	In-path	Copy TTL
Belarus	178.124.134.*	AS6697	Beltelecom	3	3	✓	
China[†]	202.97.84.*	AS4134	Chinanet	2	2		
	219.158.4.*	AS4837	China Unicom	3	4		
	61.152.24.*	AS4812	China Telecom	7	4		
	220.181.177.*	AS23724	China Telecom	14	4		
Egypt[‡]	172.17.50.*	-	-	-	0	✓	
India	125.19.50.*	AS9498	Bharti Airtel	1	1		
	116.119.44.*						
	125.18.125.*						
Iran[ϟ]	10.199.250.*	-	-	-	3	✓	✓
Kazakhstan	195.93.153.*	AS48716	PS Internet	17	1		
	91.185.5.*	AS41798	Transtelecom	2	0	✓	
	92.47.151.*	AS9198	Kazakhtelecom	4	1		
Oman	134.0.217.*	AS8529	Omantel	1	-	✓	
Pakistan	110.93.252.*	AS38193	Transworld Associates	1	0	✓	
Russia	195.239.20.*	AS3216	PJSC Vimpelcom	3	1	✓	
	31.192.111.*	AS49335	Server v arendy	111	-		
Saudi Arabia	84.235.94.*	AS39386	Saudi Telecom	1	1	✓	✓
	84.235.12.*	AS25019		5			
South Korea	112.174.83.*	AS4766	Korea Telecom	1	1		
	112.174.84.*						
Turkey	81.212.201.*	AS9121	Türk Telekom	1	2	✓	
Vietnam	113.171.45.*	AS45899	VNPT	2	3	✓	
	113.171.59.*				2		

◊ For the ethical consideration, we mask the last octet of their IP addresses.
† We observe dozens of censor routers in multiple /24 subnets in China, here we only show four of them.
‡ Not like the case of Iran, although the censor's router in Egypt also responses private addresses, it is likely to be the one which hides publicly reachable interfaces.
ϟ As mentioned earlier, the censor routers in Iran respond with private IP addresses which are domestically routable [4].

Finally, copying the TTLs helps censors unnoticeable at the server side, since legitimate packets and injected packets have the same TTLs (similar to the case in Figure 8), which avoids producing any abnormality in TTL values.

5.2 Censor Information

The application traceroute is performed as illustrated in Figure 4. Table 11 lists the censor deployment information that we can obtain through VPN servers. We identify that the censors in Iran and Saudi Arabia copy the TTL of our requests to their injected packets.

As we presented in Section 2, dropping or modifying the actual packets can only be fulfilled by the in-path censors. Using Disguiser, we can identify an in-path censor if our control server cannot receive the packets even with sufficiently large TTLs or receives the modified packets. Note that an in-path censor may also act like an on-path censor, without dropping or modifying any packets. We here consider such censors as on-path according to their actual behaviors. In total, we discover that 10 out of 13 countries deploy in-path censors, and we find that none of the in-path censors we detected modify the request. Instead, they all directly drop the requests. Also, we identify that the censors in Vietnam and Kazakhstan do not inject any packets, including RST packets, after dropping the requests.

The "Hops to Border" column in Table 11 shows the number of hops between the censor router and the nation's border router we manually identified. We can see that the censors do tend to be deployed close to the nation's border routers. In some cases (*i.e.*, one censor in Russia and one in Oman), we cannot identify the nation's border routers because the routers close to the border do not respond to ICMP requests. Furthermore, other than AS4812 and AS23724 in China, all the ASes in the table are the nation's border ASes.

The "AS Rank" column shows the country-level ranks of censors' ASes, retrieved from CAIDA's ASRank [12]. Although ASRank presents a global ranking of ASes based on the resource of an AS's all direct and indirect customers, the relative ranks within the country show that the censors are often deployed in the ASes with a high rank, indicating that they can monitor a large number of Internet users in that country. One exception is a censor we detected in Russia, and we further identify that this censor is deployed in the ISP of the VPN's hosting provider. We infer that such an observation is mainly due to the decentralized control enforced by Russia [45].

6 DISCUSSION

6.1 Validation

Our system design eliminates the false negatives since we label a case as negative only when we observe our static payload. However, as mentioned earlier, the cache proxies may introduce false positives. Although our cache tests and heuristics are able to effectively exclude most cache-impacted vantage points, there is no theoretical method to calculate how many false positives are left in our dataset. Therefore, in order to prove the detection reliability of our system, we manually search for false positives in our dataset that consists of 58 million measurements.

First, after applying the DNS heuristics (Section 3.3.2) that exclude 324 cache-impacted vantage points, our DNS dataset only includes 329 suspicious positive cases (*i.e.*, domains that have public IP addresses). We manually identify that 20 of them are false positives. Also, after applying the HTTP heuristics that exclude 446 cache-impacted vantage points and removing duplicate webpages, we leave 9,611 unique webpages for further review. In total, we identify 38 false positives in our HTTP dataset. In addition, we manually review all HTTPS interceptions and do not identify any false positives in the HTTPS dataset. As such, in total, we identify 58 false positives in our dataset, achieving a 10^{-6} false positive rate during our study.

Note that the manual review is solely used to identify the false positives generated by Disguiser that operates as an automated system. As a result, the false positive rate and false negative rate is achieved without any manual effort. In addition, the manual review can further improve the detection reliability of Disguiser, and the manual effort needed to review the dataset of Disguiser is not significant as discussed above.

6.2 Existing Censorship Measurement Systems

Table 12 presents a comparison of Disguiser with existing censorship measurement systems. Disguiser adopts remote measurement techniques and detects censorship in all three protocols, as well as identifies the deployment of the censor's equipment, while other platforms are in short of at least one of those capabilities. In addition, we evaluate Disguiser by analyzing its false positive rate and false negative rate, while other platforms do not disclose the metrics of evaluating their detection reliability, except ICLab that reports a false positive rate for DNS-based censorship only.

6.2.1 OONI. OONI [18] recruits volunteers to install software and manually run pre-defined censorship measurements. In comparison, Disguiser adopts a remote measurement technique, *i.e.*, conducting measurements remotely using the existing network protocols and infrastructures. As such, our framework can be easily adopted by other researchers to run their studies, while

Table 12. Comparison of Disguiser with Existing Censorship Measurement Systems.

System	Remote Technique†	Detection Capabilities DNS	HTTP	HTTPS	Deployment	Countries	FPR	FNR
OONI[18]		✓	✓			239	N/A♮	N/A
Iris[39]	✓	✓				151	N/A	N/A
Quack[51]	✓		✓	✓		167	N/A	N/A
ICLab[36]	✓	✓	✓			62	10^{-4} ‡	N/A
Disguiser	✓	✓	✓	✓	✓	177	10^{-6}	0

† Remote measurement leverages existing protocols and infrastructure to detect censorship activities [51].
‡ ICLab reports a 10^{-4} false positive rate for DNS-based censorship detection, while the false positive rate for HTTP-based censorship detection is not reported.
♮ N/A here means that it cannot be recognized or is not officially reported.

OONI's vantage points are not open to the public for customized experiments. Also, previous work has reported the inaccuracy of OONI's detection results. In particular, Yadav *et al.* [56] manually checked OONI's results and reported that the false negative rate could be as high as 89%. Also, Niaki *et al.* [36] reported that OONI's blockpage detector could have a 74% false positive rate.

6.2.2 Quack. As mentioned earlier, Quack [51] leverages the Echo service that reflects back the packets sent to it to trigger censors. However, as Echo servers operate on port 7 and the Quack client runs on ephemeral ports, Quack does not generate HTTP/HTTPS traffic on their standard ports, so that it may generate false negatives when a censor only monitors traffic on standard ports.

In order to compare with Quack, we operated HTTP/HTTPS on non-standard ports (including port 7 and other random ports) at our control server and issued HTTP/HTTPS requests through vantage points to our control server on these non-standard ports. Meanwhile, from the same vantage point, we conducted our standard tests, *i.e.*, sending HTTP/HTTPS on standard ports to our control server. In the comparison experiment lasting for one week, in total we identified censors in 32 countries that ignore the requests sent to non-standard ports but block the requests to standard ports. More importantly, many of those censors enforce severe censorship policies, including the ones in Saudi Arabia, UAE, Ukraine, Russia, India, South Korea, Pakistan, Kuwait, and Thailand. As a result, our observation shows that issuing legitimate traffic is critical to accurately detect censorship activities.

6.2.3 ICLab. ICLab [36] relies on VPNs to measure censorship activities. However, VPNs are usually deployed at countries that do not enforce strict censorship policies. As a result, ICLab has a limited coverage (62 countries, less than half of our covered countries), and according to the presentation in their paper, ICLab misses very important observations, such as severe DNS manipulation in China, which are detected by us and previous studies [6, 39, 49]. Moreover, ICLab does not have a ground truth on the server responses, requiring considerable manual review to identify and reduce false positives. Nevertheless, its false negatives are still unidentifiable. Therefore, its detection reliability remains unclear.

6.3 Limitation

Our censorship detection relies on the IP geolocation database to associate the censorship policies with countries. Although the IP geolocation databases are known to be unreliable, the country-level results are mostly accurate [23]. We also double check our IP geolocation results of vantage points with multiple services to improve the reliability [24, 25]. In addition, we use VPN servers experiencing censorship to detect the censor deployment, which highly relies on the availability of those servers that we can acquire.

We notice that there are some cases, in which a country/region enforces censorship but we cannot detect it in our experiment. First, in order to minimize false positives, we exclude the vantage points that are impacted by cache proxies (Section 3.3). However, it may result in the miss of censorship activities on those vantage points. Second, although our test domain list contains both popular and sensitive domains, its number is still limited and we cannot detect any censored domains that are not in our list. Third, as we only focus on the censored domains in our experiment, we will miss the censorship that examines the full URL in an HTTP request. Fourth, besides the in-network censors that examine the traffic, some DNS resolvers may also enforce their own blocking policies. Since our design does not involve with any DNS resolvers, we are unable to detect such blocking policies. Last, Disguiser cannot detect IP-based censorship activities because we specify the control servers, instead of legitimate servers, as the destination of all probing requests.

7 RELATED WORK

7.1 Censorship Measurement

Other than existing censorship measurement platforms we discussed in Section 6.2, there are plenty of significant studies that focus on censorship measurement. In recent years, researchers have investigated Internet censorship in particular countries and protocols [6–8, 13, 16, 20, 28, 32, 34, 42, 45, 56] and demonstrated that censorship activities are pervasive and persistent.

An emerging line of work adopts the remote measurement techniques to study the Internet censorship. Scott *et al.* [49] collected information on DNS resolution and resource availability to develop a tool-chain for measuring and detecting ISP-level DNS hijacking in the wild. Pearce *et al.* [38] introduced Angur, a system that infers Internet censorship practices by leveraging TCP/IP side channels to measure reachability between two arbitrary Internet locations without direct control of a vantage point at either location. Raman *et al.* [44] presented FilterMap, which leverages an enhanced Quack (*i.e.*, HyperQuack) to detects the vendors of censors based on the blockpages. To identify and collect blockpages, HyperQuack creates the templates web servers' error pages as the expected responses. However, with such a design, HyperQuack can only send inbound requests from places outside a censored region to the servers located in censored regions, and such requests may also be ignored by the censors. Furthermore, Raman *et al.* [43] further integrated Augur, Satellite/Iris, Quack, and Hyperquack into CensoredPlanet for censorship observation.

Researchers have also made significant efforts in identifying the censor deployment. Crandall *et al.* [15] first introduced the idea of application traceroute to study the location of censors. They collected and established TCP connections with web servers in China from nodes outside China, and increased the TTLs of payload to identify the location of censors. Xu *et al.* [55] then slightly modified the method by increasing the TTLs of ACK packets.However, these methods rely on the inbound requests that are less interested by censors. Yadav *et al.* [56] further utilized the application traceroute to study India's censor location with outbound requests. Instead, in our study, we leverage the VPN vantage points to successfully detect the censor deployment in 13 countries. Furthermore, by carefully examining the network trace of application traceroute, we identify the censor's behavior of copying TTL that impacts the analysis of the censor's deployment.

7.2 Censorship Circumvention

Tschantz *et al.* [50] presented a survey on the approaches for censorship circumvention by collecting real-world attack data and assessing the difficulty of blocking an approach. Nisar *et al.* [37] implemented C-Saw, a system that provides adaptive censorship circumvention for crowdsourced users while collecting censorship measurement data. Burnett *et al.* [11] developed Collage, which allows users to exchange messages through hidden channels in sites that host user-generated

content. Domain fronting has been proposed in [17]. It runs circumvention proxies on the web services that share IP addresses with other uncensored services. Many studies [9, 31, 52] utilized the implementation discrepancies of TCP state machines between censors and general-purpose servers to circumvent censorship.

CDNBrowsing systems have been proposed in [21, 35, 57] to bypass the censorship by leveraging the collateral damage of IP blocking. Another line of work [22, 30, 54] proposed Decoy routing that proxies the connection to the censored destination by routers. However, Schuchard *et al.* [48] showed that censors can effectively block the participating routers to nullify Decoy routing.

8 CONCLUSION

Internet censorship has been witnessed and studied for a long time. However, currently we still fall short of achieving an accurate censorship detection in a large scale as we have no ground truth on what are the expected responses without censor intervention. To address this issue, in this paper, we proposed Disguiser, a novel framework that enables end-to-end measurement to accurately detect the censorship activities and reveal the censor deployment. Using Disguiser, we conducted 58 million measurements from 177 countries in two time periods in two years and observed 292 thousand censorship activities in blocking DNS, HTTP, and HTTPS requests from 122 countries. Disguiser can achieve a 10^{-6} false positive rate and zero false negative rate in detecting censorship activities. We found that DNS-based censorship is not as prevalent as the HTTP/HTTPS-based censorship, and we presented the different techniques used to block domains in different protocols. In addition, we observed that China significantly added more censored domains in 2021, and Russia started to adopt HTTPS interception to enforce its censorship policies. By comparing the censored domains from different protocols, we explored the consistency of the censorship policies in each country. Furthermore, we revealed the deployment of censors in 13 countries and identified that most censors are deployed near national border gateways.

ACKNOWLEDGMENT

REFERENCES

[1] [n.d.]. Never use DNS TTL of zero (0). https://mark.lindsey.name/2009/03/09/never-use-dns-ttl-of-zero-0/.

[2] 1&1 IONOS INC. [n.d.]. Understanding and Configuring DNS TTL. https://www.ionos.com/community/server-cloud-infrastructure/dns/understanding-and-configuring-dns-ttl/.

[3] Alexa. [n.d.]. https://www.alexa.com/topsites.

[4] Collin Anderson. 2012. *The Hidden Internet of Iran: Private Address Allocations on a National Network.* Technical Report. https://arxiv.org/pdf/1209.6398v1.pdf

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

[6] Anonymous. 2012. The Collateral Damage of Internet Censorship by DNS Injection. *ACM SIGCOMM Computer Communication Review* (2012).

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

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

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

[10] Stéphane Bortzmeyer. [n.d.]. DNS Censorship (DNS Lies) As Seen By RIPE Atlas. https://labs.ripe.net/Members/stephane_bortzmeyer/dns-censorship-dns-lies-seen-by-atlas-probes.

[11] Sam Burnett, Nick Feamster, and Santosh Vempala. 2010. Chipping Away at Censorship Firewalls with User-Generated Content. In *USENIX Security Symposium.*

[12] CAIDA AS Rank. [n.d.]. http://as-rank.caida.org/.

[13] Abdelberi Chaabane, Terence Chen, Mathieu Cunche, Emiliano De Cristofaro, Arik Friedman, and Mohamed Ali Kaafar. 2014. Censorship in the Wild: Analyzing Internet Filtering in Syria. In *ACM Internet Measurement Conference (IMC)*.
[14] Citizen Lab. 2019. URL Testing Lists Intended for Discovering Website Censorship. https://github.com/citizenlab/test-lists/.
[15] Jedidiah R. Crandall, Daniel Zinn, Michael Byrd, Earl Barr, and Rich East. 2007. ConceptDoppler: A Weather Tracker for Internet Censorship. In *ACM Conference on Computer and Communications Security (CCS)*.
[16] Roya Ensaf, Philipp Winter, Abdullah Mueen, and Jedidiah R. Crandall. 2015. Analyzing the Great Firewall of China Over Space and Time. In *Privacy Enhancing Technologies Symposium (PETS)*.
[17] David Fifield, Chang Lan, Rod Hynes, Percy Wegmann, and Vern Paxson. 2015. Blocking-resistant Communication through Domain Fronting. In *Privacy Enhancing Technologies Symposium (PETS)*.
[18] Arturo Filastò and Jacob Appelbaum. 2012. OONI: Open Observatory of Network Interference. In *USENIX Workshop on Free and Open Communications the Internet (FOCI)*.
[19] FortiGuard Labs. [n.d.]. https://fortiguard.com/webfilter.
[20] Nguyen Phong Hoang, Arian Akhavan Niaki, Jakub Dalek, Jeffrey Knockel, Pellaeon Lin, Bill Marczak, Masashi Crete-Nishihata, Phillipa Gill, and Michalis Polychronakis. 2021. How Great is the Great Firewall? Measuring China's DNS Censorship. In *USENIX Security Symposium*.
[21] John Holowczak and Amir Houmansadr. 2015. CacheBrowser: Bypassing Chinese Censorship Without Proxies Using Cached Content. In *ACM Conference on Computer and Communications Security (CCS)*.
[22] Amir Houmansadr, Giang T. K. Nguyen, Matthew Caesar, and Nikita Borisov. 2011. Cirripede: Circumvention Infrastructure using Router Redirection with Plausible Deniability. In *ACM Conference on Computer and Communications Security (CCS)*.
[23] Bradley Huffaker, Marina Fomenkov, and kc claffy. 2011. *Geocompare: a comparison of public and commercial geolocation databases.* Technical Report.
[24] IP-API. [n.d.]. http://ip-api.com/.
[25] IPinfo.io. [n.d.]. http://ipinfo.io/.
[26] Lin Jin, Shuai Hao, Haining Wang, and Chase Cotton. 2018. Your Remnant Tells Secret: Residual Resolution in DDoS Protection Services. In *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*.
[27] Lin Jin, Shuai Hao, Haining Wang, and Chase Cotton. 2019. Unveil the Hidden Presence: Characterizing the Backend Interface of Content Delivery Networks. In *IEEE International Conference on Network Protocols (ICNP)*.
[28] Lin Jin, Shuai Hao, Haining Wang, and Chase Cotton. 2021. Understanding the Impact of Encrypted DNS on Internet Censorship. In *The Web Conference (WWW)*.
[29] Ben Jones, Roya Ensafi, Nick Feamster, Vern Paxson, and Nick Weaver. 2015. Ethical Concerns for Censorship Measurement. In *ACM SIGCOMM Workshop on Ethics in Networked Systems Research (NS Ethics)*.
[30] Josh Karlin, Daniel Ellard, Alden W. Jackson, Christine E. Jones, Greg Lauer, David P. Mankins, and W. Timothy Strayer. 2011. Decoy Routing: Toward Unblockable Internet Communication. In *USENIX Workshop on Free and Open Communications the Internet (FOCI)*.
[31] Sheharbano Khattak, Mobin Javed, Philip D. Anderson, and Vern Paxson. 2013. Towards Illuminating a Censorship Monitor's Model to Facilitate Evasion. In *USENIX Workshop on Free and Open Communications the Internet (FOCI)*.
[32] Sheharbano Khattak, Mobin Javed, Syed Ali Khayam, Zartash Afzal Uzmi, and Vern Paxson. 2014. A Look at the Consequences of Internet Censorship Through an ISP Lens. In *ACM Internet Measurement Conference (IMC)*.
[33] Xianghang Mi, Xuan Feng, Xiaojing Liao, Baojun Liu, XiaoFeng Wang, Feng Qian, Zhou Li, Sumayah Alrwais, Limin Sun, and Ying Liu. 2019. Resident Evil: Understanding Residential IP Proxy as a Dark Service. In *IEEE Symposium on Security and Privacy (S&P)*.
[34] Zubair Nabi. 2013. The Anatomy of Web Censorship in Pakistan. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*.
[35] Milad Nasr, Hadi Zolfaghari, Amir Houmansadr, and Amirhossein Ghafari. 2020. MassBrowser: Unblocking the Censored Web for the Masses, by the Masses. In *Network and Distributed System Security Symposium (NDSS)*.
[36] Arian Akhavan Niaki, Shinyoung Cho, Zachary Weinberg, Nguyen Phong Hoang, Abbas Razaghpanah, Nicolas Christin, and Phillipa Gill. 2020. ICLab: A Global, Longitudinal Internet Censorship Measurement Platform. In *IEEE Symposium on Security and Privacy (S&P)*.
[37] Aqib Nisar, Aqsa Kashaf, Ihsan Ayyub Qazi, and Zartash Afzal Uzmi. 2018. Incentivizing Censorship Measurements via Circumvention. In *ACM SIGCOMM*.
[38] Paul Pearce, Roya Ensafi, Frank Li, Nick Feamster, and Vern Paxson. 2017. Augur: Internet-Wide Detection of Connectivity Disruptions. In *IEEE Symposium on Security and Privacy (S&P)*.
[39] Paul Pearce, Ben Jones, Frank Li, Roya Ensafi, Nick Feamster, Nick Weaver, and Vern Paxson. 2017. Global Measurement of DNS Manipulation. In *USENIX Security Symposium*.
[40] ProxyRack. [n.d.]. https://www.proxyrack.com/.

[41] ProxyRack. [n.d.]. https://www.proxyrack.com/become-a-peer/.
[42] Ram Sundara Raman, Leonid Evdokimov, Eric Wurstrow, J. Alex Halderman, and Roya Ensaf. 2020. Investigating Large Scale HTTPS Interception in Kazakhstan. In *ACM Internet Measurement Conference (IMC)*.
[43] Ram Sundara Raman, Prerana Shenoy, Katharina Kohls, and Roya Ensaf. 2020. Censored Planet: An Internet-wide, Longitudinal Censorship Observatory. In *ACM Conference on Computer and Communications Security (CCS)*.
[44] Ram Sundara Raman, Adrian Stoll, Jakub Dalek, Armin Sarabi, Reethika Ramesh, Will Scott, and Roya Ensafi. 2020. Measuring the Deployment of Network Censorship Filters at Global Scale. In *Network and Distributed System Security Symposium (NDSS)*.
[45] Reethika Ramesh, Ram Sundara Raman, Matthew Bernhard, Victor Ongkowijaya, Leonid Evdokimov, Anne Edmundson, Steven Sprecher, Muhammad Ikram, and Roya Ensafi. 2020. Decentralized Control: A Case Study of Russia. In *Network and Distributed System Security Symposium (NDSS)*.
[46] RIPE Atlas. [n.d.]. https://atlas.ripe.net/.
[47] Robert Kisteleki. [n.d.]. Ethics of RIPE Atlas Measurements. https://labs.ripe.net/Members/kistel/ethics-of-ripe-atlas-measurements.
[48] Max Schuchard, John Geddes, Christopher Thompson, and Nicholas Hopper. 2012. Routing Around Decoys. In *ACM Conference on Computer and Communications Security (CCS)*.
[49] Will Scott, Thomas Anderson, Tadayoshi Kohno, and Arvind Krishnamurthy. 2016. Satellite: Joint Analysis of CDNs and Network-level Interference. In *USENIX Annual Technical Conference (ATC)*.
[50] Michael Carl Tschantz, Sadia Afroz, Anonymous, and Vern Paxson. 2016. SoK: Towards Grounding Censorship Circumvention in Empiricism. In *IEEE Symposium on Security and Privacy (S&P)*.
[51] Benjamin VanderSloot, Allison McDonald, Will Scott, J. Alex Halderman, and Roya Ensafi. 2018. Quack: Scalable Remote Measurement of Application-Layer Censorship. In *USENIX Security Symposium*.
[52] Zhongjie Wang, Shitong Zhu, Yue Cao, Zhiyun Qian, Chengyu Song, Srikanth V. Krishnamurthy, Kevin S. Chan, and Tracy D. Braun. 2020. SymTCP: Eluding Stateful Deep Packet Inspection with Automated Discrepancy Discovery. In *Network and Distributed System Security Symposium (NDSS)*.
[53] Zachary Weinberg, Shinyoung Cho, Nicolas Christin, Vyas Sekar, and Phillipa Gill. 2018. How to Catch When Proxies Lie: Verifying the Physical Locations of Network Proxies with Active Geolocation. In *ACM Internet Measurement Conference (IMC)*.
[54] Eric Wustrow, Scott Wolchok, Ian Goldberg, and J. Alex Halderman. 2011. Telex: Anticensorship in the Network Infrastructure. In *USENIX Security Symposium*.
[55] Xueyang Xu, Z. Morley Mao, and J. Alex Halderman. 2011. Internet Censorship in China: Where Does the Filtering Occur?. In *Passive and Active Network Measurement (PAM)*.
[56] Tarun Kumar Yadav, Akshat Sinha, Devashish Gosain, Piyush Kumar Sharma, and Sambuddho Chakravarty. 2018. Where The Light Gets In: Analyzing Web Censorship Mechanisms in India. In *ACM Internet Measurement Conference (IMC)*.
[57] Hadi Zolfaghari and Amir Houmansadr. 2016. Practical Censorship Evasion Leveraging Content Delivery Networks. In *ACM Conference on Computer and Communications Security (CCS)*.

A CENSORSHIP MAP

Figure 9 presents a censorship map that visualizes the severity of censorship in each tested country.

B ABBREVIATION OF DOMAIN CATEGORY

Table 13 lists the abbreviations of domain categories used in the paper.

C FRACTION OF VPS

Table 14 shows the fraction of vantage points that observe censorship activities in different countries. We only list the countries that we observe censorship activities on more than 5% of test domains, and they are sorted by the percentage of censored domains.

Received August 2021; revised October 2021; accepted November 2021

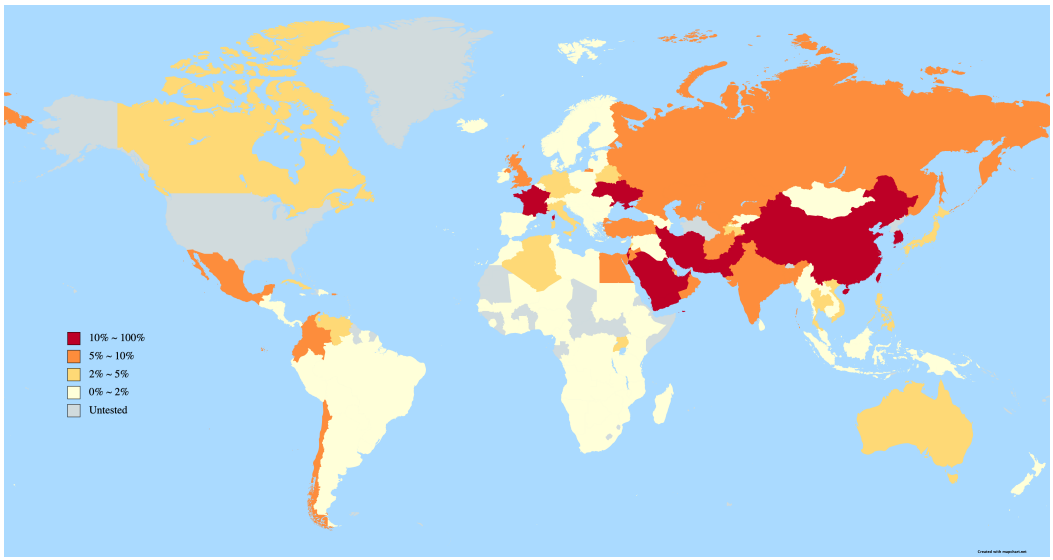

Fig. 9. Censorship Map. The percentage in the label is the ratio of the number of censored domain names to the number of tested domain names. A domain name is identified as being censored in a country if we observe the censorship behavior in any of the DNS, HTTP, and HTTPS protocol. We do not test the censorship in United States because (1) we place our control server in the United States for the convenience and (2) we do not expect pervasive censorship activities in the United States.

Table 13. Domain Category Abbreviation.

Abbreviation	Category
ADUL	Other Adult Materials
BLOG	Personal Websites and Blogs
GAMB	Gambling
ILLE	Illegal or Unethical
INFO	Information Technology
NEWS	News and Media
ORGA	General Organizations
PORN	Pornography
PROX	Proxy Avoidance
SHOP	Shopping
SRCH	Search Engines and Portals

Table 14. Fraction of VPs that Observe Censorship Activities in Countries.

Country	DNS	HTTP	HTTPS
Iran	100%† / 1.5%	100%	66.7%
China	100%† / 98.1%	100%	100%
UAE	97.9%	100%	100%
Saudi Arabia	0%	100%	100%
Ukraine	0.6%	12.1%	10.8%
Israel	0%	5.7%	4.7%
France	0.6%	2.5%	0%
Kuwait	50%	83.3%	66.7%
Pakistan	1.6%	44.5%	82.6%
Yemen	0%	100%	1%
South Korea	0%	98.2%	97.4%
Russia	0%	40.7%	30.4%
United Kingdom	0.6%	3.2%	4.4%
India	0.6%	70.0%	45.7%
Oman	1.3%	100%	100%
Egypt	21.4%	100%	100%
Nepal	0%	3.7%	1.2%
Colombia	0.8%	4.5%	3.1%
Palestinian	0%	23.1%	0%
Bahrain	0%	100%	100%
Turkey	66.7%	94.5%	93.5%
Afghanistan	0%	25.0%	33.3%
Bangladesh	4.3%	94.2%	78.4%
Ecuador	0%	12.5%	50.6%
Mexico	0%	21.0%	29.0%
Uzbekistan	0%	92.3%	100%
Kazakhstan	0%	100%	100%
Puerto Rico	0%	11.1%	0%
Jordan	15%	63.2%	72.2%

† The vantage points in Iran and China observe censorship activities on UDP-based DNS. All others are with TCP-based DNS censorship.