usenix
THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

Quack: Scalable Remote Measurement of Application-Layer Censorship

Benjamin VanderSloot, Allison McDonald, Will Scott, J. Alex Halderman, and Roya Ensafi, *University of Michigan*

https://www.usenix.org/conference/usenixsecurity18/presentation/vandersloot

This paper is included in the Proceedings of the 27th USENIX Security Symposium.

August 15–17, 2018 • Baltimore, MD, USA

ISBN 978-1-939133-04-5

Quack: Scalable Remote Measurement of Application-Layer Censorship

Benjamin VanderSloot, Allison McDonald, Will Scott, J. Alex Halderman, and Roya Ensafi
University of Michigan
{benvds, amcdon, willscott, jhalderm, ensafi}@umich.edu

Abstract

Remote censorship measurement tools can now detect DNS- and IP-based blocking at global scale. However, a major *unmonitored* form of interference is blocking triggered by deep packet inspection of application-layer data. We close this gap by introducing Quack, a scalable, remote measurement system that can efficiently detect application-layer interference.

We show that Quack can effectively detect application-layer blocking triggered on HTTP and TLS headers, and it is flexible enough to support many other diverse protocols. In experiments, we test for blocking across 4458 autonomous systems, an order of magnitude larger than provided by country probes used by OONI. We also test a corpus of 100,000 keywords from vantage points in 40 countries to produce detailed national blocklists. Finally, we analyze the keywords we find blocked to provide insight into the application-layer blocking ecosystem and compare countries' behavior. We find that the most consistently blocked services are related to circumvention tools, pornography, and gambling, but that there is significant country-to-country variation.

1 Introduction

Governments often keep specific targets of censorship secret, in order to avoid public accountability or to increase fear and uncertainty [24]. We must measure censorship to gain insights into the deployment of network interference technologies, policy changes in censoring nations, and the targets of interference. Making opaque censorship more transparent illuminates this emerging practice.

Implementing global censorship measurement continues to be a challenging problem. Today, the most common way to characterize censorship uses in-country volunteers to host network probes, such as OONI [19], or to provide anecdotes about what seems to be blocked to monitoring organizations. Both are challenging to scale. Moreover, both rely on human volunteers. For individuals living under repressive or secretive government controls, cooperating with security researchers has substantial risks.

An emerging body of work addresses these problems by using existing protocols and infrastructure to remotely measure network interference. Such approaches have been effective in measuring DNS poisoning [35,41] and for detecting interference in TCP/IP-connectivity between remote machines [17,34]. There has not yet been a global, remote method for detecting another broadly deployed censorship technique: *application-layer* censorship.

Application-layer censorship has become increasingly important with the rise of content delivery networks (CDNs). CDNs use a small number of network entry-points for a large number of customers, resulting in sizable collateral damage to IP-based blocking techniques. When an adversary wishes to block some, but not all, of these sites, they must look into the content of requests and understand the HTTP or HTTPS headers to determine which site is being requested. This form of blocking is prevalent and effective, but it is not captured by measurements of either DNS or IP connectivity.

In this paper, we introduce Quack, the first remote censorship measurement technique that efficiently detects application-layer blocking. Like other remote measurement approaches, we make use of existing internet infrastructure. We rely on servers running protocols that allow the client to send and reflect arbitrary data. This behavior is present in several common protocols, such as in the TLS Heartbeat Extension [42], Telnet servers supporting the "echo" option [38], FTP servers allowing anonymous read and write [43], and the Echo protocol [37]. After identifying compatible servers with scanning, we reflect packets that are crafted to trigger DPI policies. We aggregate instances of reliably detected disruption to identify what and where blocking occurs.

The bulk of our measurements use the RFC 862 Echo Protocol [37]. Echo was introduced in the early 1980s as a network testing tool. Servers accept connections on TCP port 7 and send back the data they receive, making the protocol easy to scan for and to validate expected responses. We find that the public IPv4 address space contains over 50,000 distinct echo servers, providing measurement vantage points in 196 countries and territories. We design and evaluate an echo-based measurement system to test over 500 domain-server pairs per second. The echo protocol also allows us to understand the importance of directionality, cases where blocking is only triggered by messages leaving a region.

The efficiency of our technique allows us to measure application-layer blocking in new detail. We first test 1,000 sensitive domains from our 50,000 vantage points around the world — taking just 28 hours. We find anomalously elevated rates of interference in 11 countries. Each of these countries is reported as restricting web freedoms by Freedom House [21]. We then consider a larger set of keywords in the 40 countries with more than 100 vantage points. We test 100,000 domains, a significantly larger corpus than can be efficiently enumerated by previous techniques. From these experiments, we observe elevated rates of interference for specific domains in 7 countries. These experiments demonstrate the effectiveness of this technique for gaining a fine-grained view of application-level blocking policy across time, space, and content.

Application-layer blocking and deep packet inspection is meant to limit access to targeted content. However, our measurements show evidence of implementation bugs introducing collateral damage. For instance, a health and wellness website is blocked in Iran because it shares part of its name with a circumvention tool. Other websites with similar content remain available.

By dynamically and continuously test application-layer blocking at global scale, Quack can reveal both deliberately and incidentally blocked websites that have not previously been enumerated. The source code is available online at https://censoredplanet.org/projects/quack.html.

2 Related Work

The phenomenon of network censorship first gained notoriety in 2002, when Zittrain et al. [49] investigated keyword-based filtering in China. This initial investigation focused on understanding policy, based off of a single snapshot of content blocking by a single entity.

Both detection and circumvention of censorship remain active problems. Many studies are based on in-country vantage points such as volunteer machines or VPNs, or are one-time and country-specific measurement projects such as studies on Thailand [23], China [9], Iran [4], or Syria [7]. These direct measurements have shown how different countries use different censorship mechanisms such as the injection of fake DNS replies [3], the blocking of TCP/IP connections [46], and HTTP-level blocking [12, 26]. Our measurements are also one-time; however our technique considerably reduces the cost of longitudinal measurement of censorship.

Application-layer Blocking Many measurement systems measure lists of keywords to test for censorship. In the context of the web, domain names are commonly used as a proxy for services, and are typically drawn either from lists of popular global domains [2], or from curated lists of potentially sensitive domains [8]. Our system uses both of these sources to maximize our comparability, and to test over a sufficiently large corpus.

Detection of keywords more broadly has made use of corpora extracted from observed content deletion, along with NLP and active probing to refine accuracy [11,22,48]. Previous systems determining such keywords have largely focused on individual countries and services, especially related to Chinese social media such as Weibo and TOM-Skype [10, 27, 28].

Direct Measurement Systems Since censorship policies change over time, researchers have focused on developing platforms to run continuous censorship measurements. One notable platform is Tor project's Open Observatory of Network Interference (OONI) [44], which performs an ongoing set of censorship measurement tests from the vantage points of volunteer participants [19]. By running direct measurements, OONI tests are harder for an adversarial network to specifically target. However, these platforms cannot easily certify that it was not the adversary themselves that contributed measurements in an effort to confound results. Moreover, OONI has a smaller number of vantage points, compared to our technique.

Remote Measurement Systems Academic measurement projects have recently renewed their focus on remote measurement of DNS poisoning [35, 41] and TCP/IP connectivity disruptions [34]. Our system extends this broad strategy to detect application-layer disruption. Our approach provides a uniquely detailed view of the trigger and implementation of interference. We can answer which direction of which packet or keyword was the trigger, and whether interference is implemented through packet injection or dropping. This level of detail is not possible in existing DNS or IP-level side channels.

Investigations of DPI Policies Deep packet inspection (DPI) and application-level disruption have become standard practice online [14]. Asghari et al. [5] find support for their hypothesis that nations pursing censorship are likely to push deployment of DPI technology. OONI reports on DPI-based censorship in 12 countries with identified vendors, and the Tor project has faced DPI-based blocking in at least 7 countries [1].

3 Design and Implementation

Quack is designed to track the use and behavior of deep packet inspection. We focus on four goals:

Detection: Since the specific triggers and behavior of DPI systems are varied and opaque, Quack focuses on detecting when keywords are blocked and the what technical methods are employed. It does not focus on uncovering application-specific grammars.

Safety: Quack is designed to run from a single vantage point, with a goal of worldwide coverage without the need to engage end users to help measure their networks. Instead, our design focuses on the use of existing network infrastructure, in this case echo servers, where the existing protocol reflects network interference information while minimizing risk to end-users.

Robustness: Our system must distinguish unrelated network activity such as sporadic packet loss or other systematic errors that only become apparent at scale from network interference. This goal is achieved by retrying upon indication of failed tests.

Scalability: We aim to accurately measure the phenomenon of keyword blocking on a global scale with minimal cost. This objective is achieved by daily scans for active echo servers, which provide us with coverage of an average of 3,716 autonomous systems daily.

In this section, we discuss our approach to detecting network interference, describe the specifics of the system we designed and built, define the datasets we acquired through our five experiments, and examine the ethical questions that arise in this work.

3.1 System Design

The Echo Protocol We chose to focus initial measurements on the Echo Protocol. The Echo Protocol, as defined in RFC862 in 1983 by J. Postel, is a network debugging service, predating ICMP Ping. The RFC states, in its entirety:

> A very useful debugging and measurement tool is an echo service. An echo service simply sends back to the originating source any data it receives.
>
> TCP Based Echo Service: One echo service is defined as a connection based application on TCP. A server listens for TCP connections on TCP port 7. Once a connection is established any data received is sent back. This continues until the calling user terminates the connection.
>
> UDP Based Echo Service: Another echo service is defined as a datagram based application on UDP. A server listens for UDP datagrams on UDP port 7. When a datagram is received, the data from it is sent back in an answering datagram.

There are many active echo servers around the world, including countries known to use DPI. Our vantage points are detailed in Section 5.

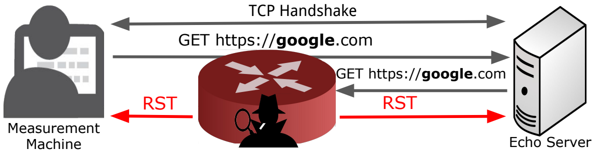

Figure 1: **Echo Protocol**—The Echo Protocol, when properly performed, is a simple exchange between the client and server where the server's response is identical to the client's request. In the example above, the censoring middlebox ignores the client's inbound request, but reacts to the the echo server's response, injecting RST packets and terminating the TCP connection.

We use echo servers for their defined purpose: measuring transport reliability. We gain additional information about the nature of any unreliability by varying the transport-layer data and observing differences in the network's behavior. This affords us insight into the nuanced network perspectives of remote hosts, contributing to the exposure of national censorship policies.

We take advantage of three features of echo that lend themselves to our purposes. First, the protocol has a well defined response to every request, which makes the classification of abnormal responses trivial. Second, due to the to send arbitrary binary data, we can test censorship of any application-layer protocol that utilizes TCP or UDP as its transport protocol. In this paper, we focus on HTTP and HTTPS. Finally, because echo servers reflect content back to our measurement machine, we are able to also detect censorship in the outbound direction, and differentiate it from censorship triggered by our inbound request. Direction-sensitive interference is a known capability of modern DPI boxes. Figure 1 illustrates the Echo Protocol in the absence of noise.

If, unlike in Figure 1, the middlebox injects a non-RST response to the echo server, we are still able to observe the interference. In fact, we are able to see the injected message because the echo server will echo the content it observes back to our measurement machine.

We note that echo is not the only protocol that can be used for this technique. We focus on it here because it provides a clear signal, but more scale can be achieved by extending measurements to any other protocol where an expected response will occur when client probes are sent.

Defining A Trial We call an individual transaction with a remote server a trial. A trial is conducted with a single server, using a single keyword, and with a single application protocol containing that keyword. For example, consider `example.com` as a keyword wrapped within the format of an HTTP/1.1 request.

During a trial, we initialize a connection to the server and send it the formatted keyword. We read the response, and pause for a short period. Finally, we send a short, innocuous payload to verify that the connection remains

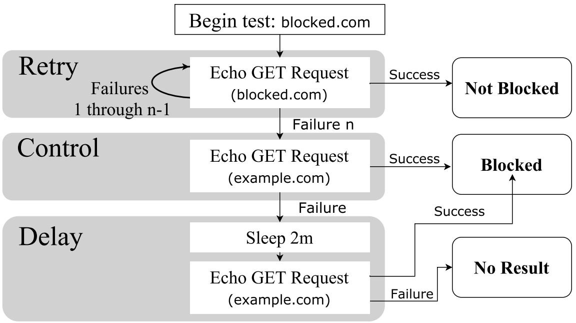

Figure 2: **Test Control Flow**—A single test using an echo server is performed by following this diagram. The most common path is also the fastest, in which an echo server responds correctly to the first request and the test is marked as Not Blocked. If the server never responds correctly, the experiment is considered a failure and we do not use the test in our evaluation.

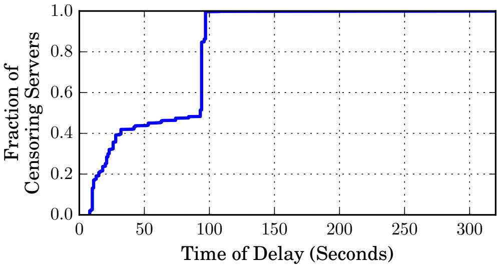

Figure 3: **Persistent Interference Duration**—We use echo servers in all countries we observe censorship to empirically measure the length of time interference occurs after a censorship event has been triggered. Roughly half of the servers responded correctly to our request within 60 seconds. By 100 seconds, 99.9% responded correctly. We therefore choose two minutes as a safe delay in the Delay Phase.

active. If the server responds the connection is closed successfully, we consider the trial a success.

The pause is necessary to allow injected RSTs by interference technology to reach either host in the connection. This gives us the ability to directly identify that an interfering network is attempting to exploit a race condition via a Man-on-the-Side deployment. By verifying that the connection is still open after the keyword is sent, we ensure that there is not asymmetric interference occurring, in which the interfering network closes the connection or begins dropping packets to our measurement machine.

Test Phases The Echo Protocol enables trivial disambiguation between correct and incorrect responses, but distinguishing noise from network interference requires additional effort. The Internet is by definition best-effort, and therefore even in the face of no interference, there will be failed connections with echo servers. Additionally, interference technologies are themselves imperfect, meaning that some trials will be successful even when the data is typically disallowed, for example when the DPI boxes are overloaded [18].

Quack is designed to extract meaningful signal from the noisiness of the network. We think about this as validating signs of failure through additional measurements, but there is a trade-off: Not retrying would lead to many false positives, resulting in an inflated rate of interference. On the other hand, many retries increase false negatives as sensitive connections slip past interference technology and are categorized as successful. We choose to be conservative in our designation of interference, designing our system to minimize false positives by retrying failures several times.

Our implementation designates a “test” as the repeated trial of a particular server and keyword. A test proceeds in three phases, as shown in Figure 2:

Retry Phase First, we run a trial with the keyword and retry if it fails. We end the test as soon as we have a successful trial, and declare the test a success. We expect interference to be sparse. For example, the highest failure rate we observed when testing sensitive keywords in a country known to implement interference was 2.2% of tests not ending in success after the first trial. We allowed up to 5 retries in our experiments.

Control Phase After five trials have failed, we progress to the Control Phase. In the Control Phase, we trial an innocuous keyword. If the server successfully completes this trial, we conclude that the five previous failures were due to network interference. If the control keyword fails, we proceed to the final phase. In our experiments, we use `example.com` as our control keyword.

Delayed Phase Finally, we account for stateful disruption. This is observed, for example, in China [47]. We test for this behavior by performing another innocuous trial after a delay. If this trial succeeds, we classify the keyword as sensitive. If it fails, we mark the test as No Result. This may occur if the echo server becomes unresponsive during our test.

We use a two minute delay determined empirically. Knowing that some middleboxes perform stateful blocking, we test every server in censoring countries with an HTTP request for the most commonly censored domain in that country. Then, we attempt to reconnect every 10 seconds with an innocuous payload until we succeed. The resulting distribution in Figure 3 shows 120 seconds is a sufficient delay.

These steps ensure that Quack is robust and can distinguish unrelated network activity, such as sporadic packet loss and other systematic errors, from deliberate forms of network interference.

Classifying Interference Although we conduct multiple trials within a test, false positive tests can still occur. We do not categorize a single failed test as interference, since it could be due to temporary routing issues or other transient failure. Even if the test is representative of policy, we wish to differentiate interference that is occurring at a local level, such as a corporate firewall, from that implemented at a national or regional level. To address both of these, we consider all tests in a country, comparing keywords by the rate of tests yielding a Blocked result. This allows us to observe the phenomenon of blocking at a country level.

This last layer of aggregation is formed by calculating a "blocking rate" for each keyword-country pair, equal to the number of tests classified as Blocked divided by the number classified as either Blocked or Not Blocked. Effectively, this removes No Results from our analysis. Prior work that has looked at failure rates aggregated across servers has required a minimum number of trials in an aggregated group to report on the blocking rate for that group [41]. We follow this convention, as it is consistent with our design goal of *Robustness*. Selecting a threshold for the number of experiments that is too low reduces our confidence, while selecting a threshold that is too high excludes more countries. Upon manual inspection of the number of servers in countries reported to perform blocking, we determine 15 as threshold that balances *Robustness* and the inclusion of anecdotally blocking countries. In Section 6.1, we validate the countries in which we observe widespread censorship using external evidence.

Due to No Result tests and echo servers churning out of our test set, the keyword blocking rates in a given country have many possible values. To approximate the probability density function of the keyword blocking rates in a country, we count the number of blocking rates in n even intervals over $[0,1]$, where n is configurable. Having this approximated distribution in each country of keyword blocking rates lets us consider each keyword's failures in the context of the country's noise. We can also categorize each country based on its distribution.

When there is no blocking, we assume Blocking events due to noise are independent and only occur with very small probability. We confirm this in Section 6.1. Since the probability of failure due to noise is so small, given our redundancy in each test, we would expect that our approximated distribution of the blocking rates be monotonic in the case that there is no blocking. In our control experiments with no expected interference in Section 6.1, we find all distributions to be monotonic, and we empirically find the blocking rate to be 0.01%.

We mark interference in countries whose distribution of keyword blocking rate is not monotonic. More precisely, we say that the keywords whose blocking rates are in the interval that breaks the monotonic trend and those keywords with higher blocking rates experience interference in that country.

We considered several trade-offs when choosing the number of intervals, n. We do not want an n larger than the minimum number of tests per keyword, 15, because this could cause consecutive numbers of blocking results to be in the same interval, creating an artifact in the distribution. However, we want as many buckets as possible, so that our smoothing does not remove too much of the detail of the distribution. To balance these concerns, we use $n = 15$ buckets consistently for the rest of our analysis.

We implement a system in Go 1.6, utilizing lightweight threads for parallelism. We restrict ourselves to one concurrent request per echo server, to restrict load on the echo server, and at most 2000 total concurrent requests. Our test server was able to process 550 requests per second and has a quad-core Intel E3–1230 v5 CPU, 16 GB of RAM, and a gigabit Ethernet uplink.

While we initially ran tests with our measurement machine source port set to 80, in order to appear more similar to real HTTP connections, we found no difference in our results while using an ephemeral source port. Using an ephemeral source port also allowed us to follow standard conventions and to host an abuse website on the standard HTTP port of our measurement machine.

3.2 Ethical Issues

Active network measurement [33], and active measurement of censorship in particular [25], raise important ethical considerations. Due to the sensitive nature of such research, we approached our institution's IRB for guidance. The IRB determined that the study fell outside its purview, as it did not involve human subjects or their personally identifiable data. Nevertheless, we attempted to carefully consider ethical questions raised in our work, guided by the principles in the Belmont [30] and Menlo [13] reports and other sources. We discussed the study's design and potential risks with colleagues at our institution and externally, and we attempted to follow or exceed prevailing norms for risk reduction in censorship measurement research.

Like most existing censorship measurement techniques, ours involves causing hosts within censored countries to transmit data in an attempt to trigger observable side-effects from the censorship infrastructure. This creates a potential risk that users who control these hosts could suffer retribution from local authorities. There is no documented case of such a user being implicated in a crime due to any remote Internet measurement research, but we nonetheless designed our technique and experiments so as to reduce this hypothetical risk.

Existing techniques [6, 34, 35, 41] in censorship measurement cause oblivious hosts in censored countries to make requests for or exchange packets with prohibited sites. In contrast, our measurements only involve connections between a machine we control and echo servers, so the echo servers never send or receive data from a censored destination.

Still, our interactions with the echo servers are designed to trigger the censorship system, as if a request for a prohibited site had been made. We cannot entirely exclude the possibility that authorities will interpret our connections as user-originated web requests, either mistakenly or by malicious intent. However, we believe that the actual risk is extremely small, for several reasons.

First, even upon casual inspection, the network traffic looks very different from a real connection from the host running the echo server to a prohibited web server. The TCP connection is initiated by us, not from the echo server. Our source port is in the ephemeral range, and the echo server's is the well known port 7. The first data is an HTTP request from us, followed by the same data echoed by the server, and there is never any HTTP response. The request itself is minimal, with no optional headers, unlike requests from any popular browser. Any of these factors would be enough to distinguish a packet capture of our probes from real web browsing.

Second, the network infrastructure from which we source our probes looks very different from prohibited web servers. We tried to make it easy for anyone investigating our IP addresses to determine that they were part of a measurement research experiment. We set up reverse DNS records, WHOIS records, and a web page served from port 80 on each IP address, all indicating that the hosts were part of an Internet measurement research project based at our university.

Third, most echo servers look very different from end-user devices. We find (see Section 5.3) that the vast majority of public echo servers appear to be servers, routers, or other embedded devices. In the unlikely event that authorities decided to track down these hosts, it would be obvious that users were not running browsers on them.

There are additional steps that we did not take for this initial study that could further reduce the risk of misidentification. We recommend that anyone applying our techniques for longitudinal data collection incorporate them. Although we established that few echo servers are end-user devices by random sampling, in a long-term study, each server should be individually profiled, using tools such as Nmap, to exclude all those that are not clearly servers, routes, or embedded devices. In addition, the requests sent to echo servers could include an HTTP header that explains they are part of a global measurement study. This would provide one more way for authorities to conclude that the traffic did not originate from an end user.

Given these factors, we believe that the risks of our work to echo server operators are extremely small. We considered seeking informed consent from them anyway, but we rejected this route for several reasons.[1] First, the risk to these users is low, but if we were to contact them to seek consent, this interaction with foreign censorship researchers would *in and of itself* carry a small risk of drawing negative attention from the authorities. Second, if we only used servers for which the operators granted consent, these operators would face a much higher risk of reprisal, since their participation would be easy to observe and would imply knowing complicity. Third, obtaining consent would be infeasible in most cases, due to the difficulty of identifying and contacting the server operators; if we limited our study to echo servers for which we could find owner contact information, this would lead to far fewer usable servers, thus severely reducing the benefit of the study. The communities that stand to benefit most from our results are those living in regions that practice aggressive censorship, and thus those who will likely benefit include the echo server operators in these regions, conforming with Menlo's Principle of Justice [13].

Beyond these risks, we also sought to minimize the potential financial and performance burden on echo server operators. We rate-limited our measurements to one concurrent connection per server, and each connection sent an average of only two packets per second. Our ZMap scans were conducted following the ethical guidelines proposed by Durumeric et al. [15], such as respecting an IP blacklist shared with other scanning research conducted at our institution and including simple ways for packet recipients to opt out of future probes.

We contrast our work with Encore [6], a censorship measurement system that has been widely criticized on ethical grounds. Websites install Encore by embedding a sequence of JavaScript. When users visit these sites, their browsers make background HTTP requests to censored domains, possibly without notice or consent. While we too make oblivious use of existing hosts without obtaining consent, the network traffic and endpoints differ dramatically from normal requests for censored content. We believe this substantially reduces the risk of harm.

4 Experimental Setup and Data

In our study, we examine URLs as the source of content that may be disrupted. In our experiments, unless specified otherwise, we send the domain name in the context of a valid HTTP/1.1 GET request. This allows us to observe a particular subset of application-layer interference, and one that is well documented [11].

[1] As discussed by others [33,40], informed consent is not an absolute requirement for ethical research, so long as the research abides by other principles, e.g. those in the Belmont and Menlo reports or those steps proposed by Partridge and Allman [33], as we have strived to do.

Control We first perform a control study. To do so, we test a number of innocuous domains as our keywords, which are expected not to be censored, and repeat them against every echo server. The domains we choose are of the form `testN.example.com` with incrementing values of N. We perform this experiment 1109 times per server. Since there should be no artificially induced network interference, we can validate our technique using the results of this study. This test was performed July 20–21, 2017 from our measurement machine inside of an academic network.

Citizen Lab We use the the global Citizen Lab Block List (CLBL) [8] from July 1, 2017 as a list of keywords to run against all echo servers. This list has 1109 entries. It is curated by Citizen Lab to provide a set of URLs for researchers to use when they are conducting censorship research. Significant difference between this test and the previous test indicates that our system is capable of detecting application-layer interference of the domains in this list. This test ran on July 21–22, 2017, from our measurement machine.

Discard We then repeat the Citizen Lab study using a closely related protocol, the Discard Protocol [36]. The Discard Protocol is designed similarly to the Echo Protocol, but instead of echoing back any received data, it is simply discarded. By repeating our experiment with discard, we can determine if existing middleboxes detect keywords that are seen inbound to its network. If this were the case we would see the same interference in the Discard Protocol as the Echo Protocol. Otherwise, we will be able to determine that interference technologies do notice the direction of sensitive content. This test run on July 19–20, 2017, from our measurement machine.

TLS This study demonstrates the application-layer flexibility of our technique. We perform the Citizen Lab experiment again, but instead of embedding the Citizen Lab domain list in valid HTTP request, we place the domain in the SNI extension of a valid TLS `ClientHello` message. This will allow us to discern what difference exists between interference of HTTP and HTTPS. This test ran on July 23–24, 2017, from our measurement machine.

Alexa Top 100k Finally, we use our system to test the top 100,000 domains from Alexa [2] downloaded on July 12, 2017. This is a set of domains orders of magnitude larger than that of prior works studying application-layer censorship. To achieve full measurement of such a large set of domains, for each domain we select 20 servers in each country. Additionally, we restrict our test to the 40 countries with more than 100 echo servers. This test demonstrates most of all that our tool can be used at scale for significant research into application-layer blocking at a country granularity. This test ran on July 25–28, 2017, from our measurement machine.

Server Set	IP Addresses	/24s	ASNs	Countries
SYNACK	5,260,118	109,729	6,932	198
Echo	57,890	38,977	3,766	172
Stable (24 hr)	47,276	31,802	3,463	167

Figure 4: **Discovery of Echo Servers**—Server discovery is a staged process. A ZMap scan discovers servers that SYNACK on port 7, but we find that most of these servers will fail to ACK or will RST when receiving any data. To remove these misbehaving echo servers, we attempt to send and receive a random string to all SYNACK servers, giving us the set of functioning echo servers. Of these, 47,276 remained Stable over 24 hours, making them useful for long running experiments.

5 Characterization

In order to better understand any biases inherent in our data, we first characterize the population of echo servers we make use of in our study.

5.1 Discovery

To discover echo servers in diverse subnets and geographic locations, we perform Internet-wide scans with the ZMap toolchain [15] on the IPv4 address space. We ran daily scans for two months, between June 1st to July 31st, discovering more than 50,000 echo serves each day.

Upon discovering hosts that respond to our SYN packets on port 7, we initiate connections to the potential echo servers. We send a randomly generated string and verify that they reply with an identical string. During our first trial, we find that $57,890$ servers reply with the correct string, over $3,766$ ASNs. Many of our experiments take place over the course of a day, so we measure the coverage of echo servers that reply 24 hours later. We find 92% of ASNs have an echo server that is online during this second test.

In Figure 4, we show the number of servers still online after 24 hours, which is significant because our experiments run over the course of a day. Only those servers that are stable for at least 24 hours will test all keywords in the experiment. We observe that this reduces the diversity of our coverage, but not significantly, and note that this biases our results towards stable echo servers.

5.2 Churn

We looked at our daily scans in order to understand how stable echo server IP addresses are over time. While an average of 17% of echo servers churn away from their IP address within 24 hours, we observed that 18% were stable and responsive throughout the entire duration of our measurement. Additionally, the rate at which echo servers churn decelerates, so the first day reports the largest churn rate across the study.

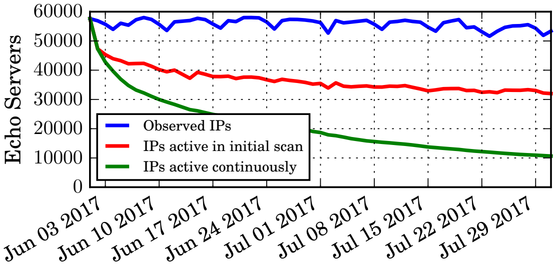

Figure 5: **Echo Server Churn**—Only 18% of tested servers were reachable in every observation over 2 months of daily scans. However, 56% were present in both our first and final scans.

Echo servers not only churn out of the set of IP addresses from a given day—they also churn back in, as shown in Figure 5. While we only observed 18% of echo servers from our first discovery scan in every daily scan, 56% of echo servers from our first discovery scan were also in our final scan 61 days later.

5.3 Identification

To understand the composition of machines running echo servers, we randomly selected 1% of responding echo servers on July 17, 2017. For this sample, we performed OS detection on each IP address using Nmap. The most common system families as defined by Nmap are shown in Figure 6. There were 56,228 working echo servers on this date. Of the 562 we tested, Nmap identified 463 (82.4%) of the operating systems. Nmap reported a median accuracy of 99% for the identifications. This test covered 54 countries.

Of the echo servers we scanned with Nmap, 251 (44.7%) had full device labels containing the words “server”, “router”, or “switch”. Of the remaining echo servers, 70 (12.5%) were Linux, and 26 (4.6%) were Windows. The rest were identified as various other systems such as firewalls, controllers, and embedded systems. In total, 4% of echo servers were given device labels that left doubt as to whether they were infrastructure machines, because they were identified as non-server Windows machines, and 2 devices were identified as running Android. It would be infeasible to run Nmap’s OS detection service against all echo machines, but we do not believe that to be necessary to safely use all functioning echo servers, as we discuss in Section 3.2.

5.4 Coverage

Echo servers provide us diverse vantage points in a majority of countries. We associate IPs with autonomous systems using the publicly available Route Views dataset [39], and locate each server to a country using the MaxMind GeoIP2 service [29].

OS Family	Echo Servers
Windows	180 (32.0%)
Embedded	139 (24.7%)
Linux	71 (12.6%)
Cisco IOS	38 (6.8%)
Unsuccessful identification	99 (17.6%)
Other	35 (6.2%)

Figure 6: **Identification of Echo Servers**—We scanned 562 (1%) echo servers with Nmap’s operating system detector on July 17, 2017 and found that the most of the echo servers were either Windows machines or embedded devices, as identified by Nmap. This scan yielded a median accuracy of 99%.

On average, we observed echo servers in 177 countries. Of these countries, we observe an average 39 countries with more than one hundred echo servers and 82 countries with more than fifteen echo servers. This provides insight into a large number of countries.

We compare our method’s coverage with that of the OONI project [19], which enlists volunteers worldwide to run scans from local devices to measure network disruption. OONI makes this data public with the consent of the volunteers, but probes do not have unique identifiers; therefore, we use the of number of distinct autonomous systems per country to estimate coverage.

We compared the number of unique ASes observed for both tests during the week of July 8–15, 2017. As shown in Figure 7, echo servers have a much more diverse set of vantage points and over a larger number of countries. During the week of our comparison, OONI data was available for 113 countries, while echo servers were responsive in 184. Furthermore, the total number of ASes seen in the echo measurements was nearly an order of magnitude larger than that of OONI: we observed echo servers in 4458 unique ASes; OONI measures 678. While OONI probes provide rich measurement for the locations they have access to, our technique providers broader and more consistent measurements.

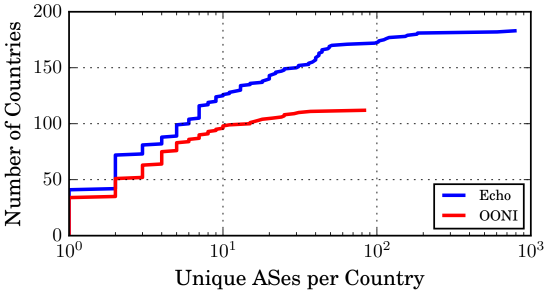

Figure 7: **Coverage of Autonomous Systems per Country**—Echo servers were present in 184 countries with 4458 unique ASes, while OONI probes were in 113 countries with 678 ASes.

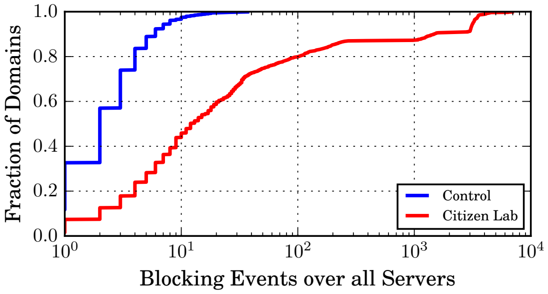

Figure 8: **Keyword Reliability**—Each of 1109 domains were sent to 54,515 echo servers for the Control and 54,802 for the Citizen Lab experiment. We count the blocking events per keyword, observing that the largest blocking rate for a given keyword was 8.5% in CLBL and 0.08% in the Control. This supports our hypothesis that these domains are sensitive.

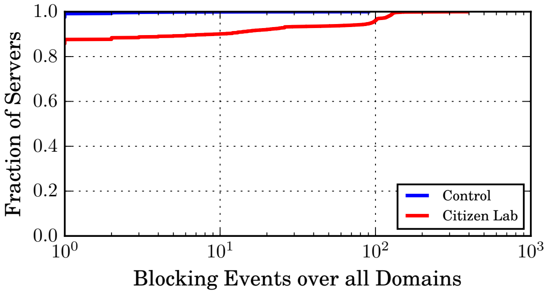

Figure 9: **Server Reliability**—For both the Control and Citizen Lab experiments, we send 1109 mock HTTP requests to all echo servers. We find that 98% of servers never resulted in a blocking event in the Control experiment. We observe significantly more blocking among a small set of servers in the CLBL test. This demonstrates that interference occurs with very few hosts.

6 Evaluation

In this section, we provide the results of the studies described in Section 3. Our evaluation provides support for the Quack's practicality as an application-layer measurement tool in two ways. First, we describe what behavior our measurements detected given a set of URLs known to be censored, in order to verify that our results correlate with previously observed phenomena. Then, we support our claim that our system works at scale, and present the results of an experiment that measured a larger corpus of domains across a greater number of countries than any previous study.

6.1 Validation

We control for noise, non-protocol-compliant servers, and other anomalous behaviors by measuring echo server behavior using innocuous domains of the form test$N$.example.com. Mock queries to these domains are used to demonstrate behavior in the absence of disruption, since these domains are unlikely to be blocked. This allows us to identify a baseline for ordinary network and echo server failure when interacting with each remote network, and understand our subsequent test results in light of a baseline model of expected behavior.

The first assumption we make in designing our control tests is that the class of domains test$N$.example.com will face no blocking by the network between our server and the echo server. To validate this assumption, we perform a set of measurements to all echo servers using only this control class of domains, and consider the failure percentages we observed. We show the distribution of failures per domain tested in Figure 8.

We observe a median domain failure rate of less than 0.01%, and a maximum failure rate across 1109 domains of 0.08%. Additionally, the domains in the upper quartile of disruption rates are evenly distributed over the class of innocuous domains, independent of the value of N.

Using the technique described in Section 3.1, we classify no country as interfering with any of our control domains. We also confirmed these results using another control domain: `echotest.[redacted].edu`, validating our control.

We assume failures in the absence of network interference are independent of which server is used. This allows us to present a distribution for the null hypothesis that is independent of either variable, and therefore constant. A few factors could cause a given server to fail many innocuous domains: network unreliability, echo server unreliability, or actual blocking occurring for our innocuous domains. Despite this, in Figure 9 we see that 98% of servers see no blocking events.

We observe that during the duration of our experiment, 17% of echo servers appear to churn away, which is indicated by their yielding two No Result tests sequentially. This is roughly as many as we observe churning away in a day for our discovery scans. This confirms that our results will be biased toward networks with stable echo servers.

Finally, we empirically determine how long measurements should wait when a blocking event is detected in order to allow stateful DPI disruption to disengage. Shorter timeouts will allow us to test more domains against a given server in a shorter time, while longer timeouts are less likely to incorrectly classify a domain as a failure due to a previous sensitive domain having triggered stateful blocking. Our system as implemented is not fundamentally limited by a longer timeout, because there are more servers to test at any given time than there are servers waiting for that timeout to expire. As such, the two-minute delay we empirically determined as shown in Figure 3 is

Country	HTTP	Discard	TLS	Top Categories
China	126	126	0	NEWS, ANON
Egypt	6	5	2	ANON, NEWS
Iran	25	0	374	PORN, LGBT
Jordan	8	1	4	ANON, NEWS
Kazakhstan	4	0	0	MMED, FILE
Saudi Arabia	2	0	0	NEWS, ANON
South Korea	14	0	0	PORN, GMB
Thailand	11	0	0	PORN, NEWS
Turkey	12	14	14	ANON, NEWS
UAE	8	0	17	NEWS, COMT
Uzbekistan	1	—	1	MISC
Union	220	146	435	NEWS, ANON

Figure 10: **Interference of CLBL**—We perform multiple experiments to measure interference of domains in the Citizen Lab global block list. Quack detected keyword blocking in 13 countries, with 220 unique domains blocked in our simple HTTP experiment. There is little intersection between different countries, and only 20% of tested domains exhibited interference anywhere. Category abbreviations are defined in the Appendix.

a minimum, and the system may take longer to schedule the subsequent trial in a test against a disrupted server. We observe that all delays were less than five minutes in practice.

6.2 Detection of Disruption

Next we test each of the domains on the Citizen Lab global list against all echo servers by formatting them as valid HTTP GET requests. We expect to see interruption in this test because the Citizen Lab domains are known to be blocked in countries around the world. This is confirmed by the difference to the control in Figure 8 and Figure 9.

Using our method of classifying interference as described in Section 3.1, only 12 countries of 74 tested against all domains demonstrate evidence of keyword blocking in this test. The interfering countries, number of domains for which we observe interference, and what categories those domains are contained in are given in Figure 10.

For each country we list in Figure 10, we look for external evidence to support the conclusion that we observe government-sanctioned censorship. One source of external evidence is the Freedom on the Net report by Freedom House [21]. Of the countries in the table, nine are rated as "Not Free" and two are rated as "Partially Free."

South Korea and Jordan are those listed as Partially Free by Freedom House; however, both are indicated in ONI's most recent country profiles as performing filtering [31, 32]. In the case of South Korea, blocking based on HTTP request content is specifically identified. In further support of the observed phenomenon being action at a national level, the echo responses in South Korea that did not match the echo requests were HTTP redirects to a government-run website outlining the reason the requested domain was blocked. This is another advantage of the Echo Protocol — we are able to see the responses injected to the echo server, because they are then echoed back to us.

Two countries were identified by our system as having a significant proportion of blocking, but had no evidence from other sources that there would be restrictions on the Internet: Ghana and New Zealand. Ghana is not evaluated by Freedom Net, but the Department of State stated in its 2016 Human Rights report [45] that there were no governmental restrictions to the Internet. Upon inspecting the scope of blocking, in both cases, it is restricted to a single academic network in the country, and all echo servers in that AS reported interference. In all other countries identified by our system as performing blocking, we observe interference in more than one AS. Our technique is not fine-grained enough to detect censorship across all networks, and in these cases we have visibility into only a few locations that have close proximity. For these reasons, we exclude Ghana and New Zealand from Figure 10.

While this presents a case that the interference we identify is genuine, we do not claim that we identify all genuine interference. The list of all countries with at least 15 echo servers is presented in the Appendix. This list has multiple other countries that are listed as "Not Free" in the Freedom of the Net report, including Belarus, Russia, Pakistan, and Vietnam.

Pakistan, as an example, is identified by prior work [41] as practicing DNS poisoning. DNS poisoning is one potential implementation of Internet censorship, and would render application-layer blocking unnecessary. The technique presented in this paper does not consider any other possible implementations of Internet censorship, and will therefore miss countries who do not rely heavily on application-layer censorship. Furthermore, many non-technical factors are included in the Freedom of the Net rating; not all "Not Free" countries block content using technical means.

We have validated our classifications with anecdotal reports, but we also want to ensure there is consistency in our classification. To do so, we look at what percent of ASes, /24s, and echo servers in a given country observe any Blocked result in this experiment. The countries that we observe widespread blocking in are represented in the shaded region in Figure 11. While some countries have interference in almost all instances, e.g. China, there are several countries with interference not performed across the entire country. This potentially reflects heterogeneous deployments of interference. We observe in Figure 11 that some countries that we do not classify as blocking any domains have comparable numbers of servers experiencing at least one Blocked result as countries we do classify

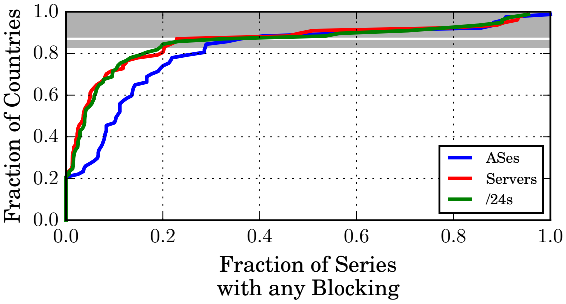

Figure 11: **Blocking Rates Per Country**—We examine the CLBL results, looking at what fraction of ASes, Servers, and /24s in each country observe any Blocked result. The shaded regions are countries we identify as having widespread interference. While some countries face near ubiquitous interference across tested servers, more countries display large variation.

Domain	Blocking Countries	Category
www.hotspotshield.com	5	ANON
www.xvideos.com	4	PORN
www.pornhub.com	4	PORN
www.gotgayporn.com	4	PORN
bridges.torproject.org	4	ANON
www.pokerstars.com	3	GMB
adultfriendfinder.com	3	DATE
www.torproject.org	3	ANON
www.wetplace.com	3	PORN
ooni.torproject.org	3	ANON

Figure 12: **Top Interfered CLBL Domains**—We compared the list of domains interfered with in each country to find those most broadly blocked. The top 10 are presented above. Pornographic websites are overrepresented in the table, but the single most broadly blocked domain is the homepage of a free circumvention technology. China blocks every domain in the table.

as blocking. These countries, Mexico and Zambia, have blocking events that are disperse and inconsistent in the set of domains being blocked, reflecting either unreliable echo servers or echo servers with highly-local blocking. Additionally, these countries had “no reports of blocking” in the Freedom of the Net 2016 report [21].

The most commonly blocked domains we observe in the Citizen Lab block list are shown in Figure 12. The most commonly blocked domain is `www.hotspotshield.com`, the homepage for a free VPN service. VPNs are common circumvention tools. Surprisingly, it is only blocked in five of the 13 countries where we detected censorship: China, Iran, Jordan, Turkey, and UAE. We see that the most consistently blocked domains are for circumvention tools, pornography, and gambling. Political content tends to be region-specific, and is less often blocked by multiple countries.

6.3 Disruption Mechanisms

By using echo servers, we ensure that the potentially sensitive payload is on both the inbound and outbound halves of the connection. This means that our system will detect interference regardless of directionality of the censor. In order to test whether the direction of the request matters, we perform the Citizen Lab test using the Discard Protocol [36]. This protocol is similar to the Echo Protocol, but instead of echoing the request, the server only ACKs the data. Blocks that occur in our test of echo servers, but not discard servers, could be instances of blocking on only outbound data. This test provides additional valuable insight into the mechanisms used for blocking.

We test the subset of echo servers that are also discard servers, sending identical payloads as in Section 6.2. Echo servers are also often discard servers, so this requirement reduced the number of testable servers from 57,309 to 27,966. Of the 11 interfering countries, we are able to maintain enough servers to classify disruption in all but Uzbekistan.

In the 10 remaining countries we observed blocking when using echo servers, we continue to observe disruption in only 4 when using the Discard Protocol: China, Egypt, Jordan, and Turkey. This implies the other countries we observe performing HTTP blocking are doing so only on data outbound from their network. This evidence is not necessarily conclusive, as the reduced set of echo servers may be reducing our visibility into these countries. For example, we observe reliable disruption in a few Iranian ASes for the Discard Protocol. However, because the vast majority of Iranian ASes do not interfere in this test, we do not classify the interference as widespread across the country.

6.4 HTTP vs. HTTPS

The Echo Protocol allows arbitrary data to be sent to and returned by the echo server. This flexibility is a strength of our technique, and is an advantage over other protocols with more constraints on sending and receiving arbitrary byte streams. To demonstrate why this capability is important, as well as illuminate practices in network interference, we repeat our test of the Citizen Lab Block List, but send requests formatted as valid TLS ClientHello messages with the Server Name Indication (SNI) Extension.

The Server Name Indication Extension [16] was developed to allow a TLS client to inform the server what domain it is attempting to connect to before the server must send a certificate. Since certificates are used for authentication and linked to domain names, a server hosting many websites would need this information to connect to a client securely. Unfortunately, SNI places the domain name in clear-text in the first message sent by the client to the server, making it easy to detect when a client is connecting to a particular site from only the first message in a TLS handshake. We find that networks do interfere based on this first message alone.

Of the 12 interfering countries we detect in the Citizen Lab experiment, we were able to conduct enough tests to confidently classify all of them. We continue to observe disruption in only 5 when using TLS: Egypt, Iran, Jordan, Turkey and UAE. For the other countries in Figure 10, TLS may aid in circumventing interference of HTTP requests based on the application-layer.

The only instance of interference occurring in a country that was not detected with just HTTP requests from the Citizen Lab list occurs in New Zealand. The domains blocked are identical across two servers in the same /24 routing prefix, which is allocated to an academic institution in the country. We conclude that the blocking is being performed by the institution, and not a national policy decision to only block HTTPS.

While the domains we observe interference with are similar in four of the five countries, in Iran the set of disrupted domains grows drastically when testing with TLS ClientHellos: the number of blocked domains in Iran increases from 25 to 374. The list of blocked URLs also changes composition to include significantly more domains classified by Citizen Lab as News, Human Rights, and Anonymization tools.

There are several possible reasons a country would implement a policy blocking a domain through HTTPS but not HTTP. As the domain name is the only visibility into the nature of the content in a HTTPS connection, a country could be aggressive in blocking domains where only a single page on the domain is undesired. In the case of HTTP they could simply block the specific page or given keywords, since all of the content will be visible to the censor. Alternatively, a country could wish to have visibility into the resources accessed at a given site, which forcing a downgrade to HTTP would allow.

6.5 Disruption Breadth

We have established to this point that we have a tool that allows us to test for application-layer censorship across 74 countries for roughly a thousand domains. While this is useful, we explore a different capability of our tool in this section. We perform a search for disruption across 40 countries for the 100,000 top domains as ranked by Alexa [2].

In order to perform tests across this many domains, we restrict ourselves to at most 20 requests per domain per country; this reduces the total number of requests dramatically. Several countries contain thousands of echo servers. Additionally, because we only make serial requests to any particular server, we test only in countries with at least 100 servers. This means the most requests a server must process sequentially is 20,000.

Country	Domains Blocked	Unique	Citizen Lab
China	787	712	146
Egypt	27	20	1
Iran	1002	795	10
Saudi Arabia	3	2	1
South Korea	1572	1139	15
Thailand	38	16	0
Turkey	291	120	7
Union	3293	—	180

Figure 13: **Interference of Alexa 100k**—We test the Alexa Top 100k domains across the 40 countries with the most echo servers and observe censorship in 7. The number of censored domains in the Alexa list does not necessarily correlate with the number blocked in the CLBL, but every country seen blocking in the Citizen Lab experiment also interferes in the Alexa 100k.

This experiment reveals interference in 7 countries, presented in Figure 13. Of the countries with enough echo servers to be tested, the countries we observe blocking the top domains are the same countries who were blocking domains in the Citizen Lab experiment.

Of the domains that are similar in both the Citizen Lab list and the Alexa Top 100k, we see large overlap in blocked domains. We define similar domains as those with the same domain name, not including sub domains.

One interesting behavior this heuristic shows is in Egypt. Several `torproject.org` subdomains are tested in the CLBL, but only the root domain was tested in Alexa. We observe that the interference in Egypt is dependent on subdomain: the root domain `torproject.org` is not blocked, and the subdomain `www.torproject.org` is blocked on one echo server in Egypt when tested only seconds apart.

Another interesting blocking behavior we observe is that Iran blocks an innocuous health and lifestyle site, `psiphonhealthyliving.com`. This site is likely collateral damage, as Iran also blocks the domain `psiphon.ca`, the homepage for a censorship circumvention technology. Additionally, we can observe that in Iran, all domains belonging to the Israeli TLD (`.il`) are blocked.

Testing the Alexa 100k provides insight into what is being blocked in each country, without introducing the biases of the people manually curating lists, such as the CLBL. In Figure 14, we analyze the domains blocked in our Alexa experiment that were not included in the Citizen Lab experiment. Our domain categorization is performed by FortiGuard Labs, a common DPI tool provider, using their web interface [20].

Many of the domains we discover as blocked in our test of domains from Alexa are pornography. Interestingly, some domain classifications were not at all present in the

Category	Blocked Domains Not in CLBL
Pornography	2085 (99%)
News and Media	114 (92%)
Search Engines and Portals	100 (98%)
Information Technology	85 (97%)
Personal Websites and Blogs	85 (50%)
Proxy Avoidance	59 (87%)
Shopping	36 (100%)
Other Adult Materials	35 (90%)
Entertainment	33 (97%)
Streaming Media and Download	31 (86%)
Uncategorized	89 (96%)
Other	378 (94%)

Figure 14: **Alexa Domain Discovery**—We categorize the domains blocked in each country in our Alexa 100k experiment, excluding those with a similar entry in the Citizen Lab experiment, and present the top 10 categories. As in other experiments, the largest censored category is pornography. However, other categories show the breadth that can be uncovered by testing the entire Alexa 100k. For example, none of the blocked shopping domains in the Alexa dataset were in the CLBL.

Citizen Lab experiment, such as Shopping. Other categories, such as Personal Websites and Blogs and News and Media, can be extremely informative when considering what content is deliberately blocked by countries. Overall, we see that $3,130$ of the domains we observe as blocked are not in the CLBL. This is a significant improvement in coverage of blocked URLs, as we only see 220 URLs blocked from the Citizen Lab list.

Using the large set of domains tested, we can compare what domains are blocked in multiple countries, despite the sparseness of block list intersections. Many categories have domains that are not blocked in multiple countries, e.g. News and Media, meaning that the particular news sites blocked by each country are not the same as in other countries that also censor News and Media sites. In contrast, the set of blocked domains depicting violence and advocating extremism are the same in every country that blocks that type of content.

Finally, we utilize the ordered nature of the Alexa top domains to compare how each country's blocking changes with the popularity of a site, shown in Figure 15. While some countries show generally uniform distribution of blocking across the top $100,000$ domains, others show a tendency to select domains from the most popular. Countries demonstrating the tendency to block popular domains with greater frequency are China, Egypt, and Turkey, with the strongest trend being that of Turkey. This may reflect a reactive blocking strategy, in which domains are added to a blacklist when they are detected to be visited with some frequency by citizens.

While the Alexa Top 100k experiment is only one snapshot of the state of application-layer censorship taking place on HTTP and HTTPS, we believe that it demonstrates the flexibility and accuracy of our tool. In the future, it can be used to contribute valuable data to many diverse, longitudinal, and in-depth studies of application-layer censorship.

7 Discussion

This paper has proposed and validated a technique for measuring application-layer interference around the world. In this section, we discuss the limitations of the design and what additional research our tool enables.

Limitations Our system currently relies on echo servers to gain perspective into remote client experiences of the Internet. Existing remote measurement techniques can be detected and invalidated or blocked by middleboxes, and ours is no exception.

First, the censor could block all traffic through port 7. We have no information about who or what else might be using port 7 today, so we have very little idea of how much collateral damage blocking port 7 would cause. Fortunately, our system is not dependent on using the Echo Protocol specifically; there are several other protocols that offer an echo service, such as FTP, Telnet, and TLS. These other protocols would be much more difficult to block entirely, as they are used much more widely on the Internet. Many of these alternates do have the disadvantage of requiring a protocol-specific header, which may cause some middleboxes to stop responding to our probes.

Second, the censor could block our measurement machine by IP. One of the greatest advantages of our system is that it is portable; the measurements can be run from virtually any machine around the world. This means that any IP-based filtering of our measurements would likely be unsuccessful, as we could quickly and easily deploy in another location.

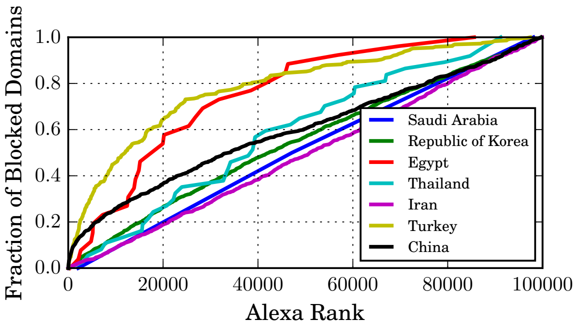

Figure 15: **Blocking by Alexa Rank**—The distributions of blocked domains relative to their Alexa rank varies by country. Egypt, Turkey, and China demonstrate a clear trend of blocking lower-ranked domains at a higher frequency. In contrast, Iran has a near uniform distribution of blocking across Alexa ranks.

Finally, a censor could watch for the direction a connection and block only connections originating from inside their network. However, such a policy would not prevent services pushing data to clients, as can occur in FTP. In practice, we are not aware of directional blocking of this nature, potentially because the complexity of AS peering blurs the distinction of internal and external networks at a nation-state level.

However, both distributed and remote censorship measurement systems in use today are differentiable and disruptable. Even if some censors decide to disrupt measurements, we will continue to have visibility into the rest of the world.

Another limitation is the difficulty of detecting countries with heterogeneous deployments of keyword blocking, because in this work we considered only widespread blocking. Future work can remove our final Classifying Interference step, and instead combine the raw data with that of other network disruption measurement techniques [34, 35] to increase the granularity of observations.

Another limitation of the measurements conducted in this study, but not to our technique in general is that we have false negatives where DPI boxes monitor only port 80 and port 443 for web traffic. We could have conducted all of our experiments with our client port set to the appropriate well-known port for the protocol we would measure; however, we believed the trade-off was best to follow the best common practice and use an ephemeral port for our client connections.

One consideration in using this work for global detection is that there are only on average 177 countries with echo servers, and only 74 with at least 15 vantage points. One potential way to increase the number of vantage points is to send our formatted requests to any server that accepts packets. For example, this could be done for HTTP by using all web servers. Then we would differentiate between the web server's error result and the interference behavior by country. However, this removes our ability to detect disruption that only inspects outbound packets from the network. Based on what we have observed in Section 6.3, this is a significant number of the countries that perform application-layer interference.

Finally, our work makes a trade-off to detecting censorship that is observed in multiple vantage points within each country, but this comes at the price of reduced granularity of observation. This means we will not regularly observe censorship that is heterogeneously implemented within a given country, and will not be able to reliably observe particular ISP policies.

Future Work This paper describes a new and useful technique that can be used to remotely measure network disruptions due to application-layer blocking. Disruption detection techniques can monitor DNS poisoning, IP-based blocking, and now application-layer censorship. When combined, these perspectives could produce valuable datasets for political scientists, activists, and other members of the Internet freedom community. Additionally, these remote measurement techniques complement in-country probes, such as OONI, in order to provide baselines and focus effort.

The system presented here is capable of continuous measurement. Rather than regularly running a large batch of keywords, such as the Alexa list, a different optimization would cycle through a set of interesting domains in each country at a reduced rate. This would enable longitudinal tracking of those domains, and help illuminate how and when application-layer censorship policies change.

Quack also stands to provide interesting insight into censorship of other application-layer data and can be generalized to use other protocols' echo behavior. While we only focus on HTTP and HTTPS in this paper, the Echo protocol's ability to send and receive arbitrary data could be used to explore interference in other areas, such as the mobile web and app ecosystems. Additionally, future work can be performed to use protocols other than the echo protocol. This would improve coverage of application-layer blocking measurement.

8 Conclusion

Application-layer interference is broadly deployed today, critically limiting Internet freedom. Unlike other techniques for censorship, we have not previously had broad and detailed visibility into its deployment. In this paper, we introduced Quack, a new system for remotely detecting application-layer interference at global scale, utilizing servers already deployed on the Internet, without the need to enlist volunteers to run network probes. We hope that this new approach will help close an important gap in censorship monitoring and move us closer to having transparency and accountability for network interference worldwide.

Acknowledgments

The authors are grateful to Bill Marczak and Adam Bates for insightful discussions, and to the anonymous reviewers for their constructive feedback. This material is based upon work supported by the U.S. National Science Foundation under grants CNS-1409505, CNS-1518888, and CNS-1755841, and by a Google Faculty Research Award.

References

[1] S. Afroz and D. Fifield. Timeline of Tor censorship, 2007. http://www1.icsi.berkeley.edu/~sadia/tor_timeline.pdf.

[2] Alexa Internet, Inc. Alexa Top 1,000,000 Sites. http://s3.amazonaws.com/alexa-static/top-1m.csv.zip.

[3] Anonymous. Towards a comprehensive picture of the Great Firewall's DNS censorship. In *Free and Open Communications on the Internet (FOCI)*. USENIX, 2014.

[4] S. Aryan, H. Aryan, and J. A. Halderman. Internet censorship in Iran: A first look. In *Free and Open Communications on the Internet (FOCI)*. USENIX, 2013.

[5] H. Asghari, M. Van Eeten, and M. Mueller. Unraveling the economic and political drivers of deep packet inspection. In *GigaNet 7th Annual Symposium*, 2012.

[6] S. Burnett and N. Feamster. Encore: Lightweight measurement of web censorship with cross-origin requests. In *ACM SIGCOMM Conference*, pages 653–667, 2015.

[7] A. Chaabane, T. Chen, M. Cunche, E. D. Cristofaro, A. Friedman, and M. A. Kaafar. Censorship in the wild: Analyzing Internet filtering in Syria. In *Internet Measurement Conference (IMC)*. ACM, 2014.

[8] Citizen Lab. Block test list. https://github.com/citizenlab/test-lists.

[9] R. Clayton, S. J. Murdoch, and R. N. M. Watson. Ignoring the Great Firewall of China. In *Privacy Enhancing Technologies (PETS)*, Cambridge, England, 2006. Springer.

[10] J. R. Crandall, M. Crete-Nishihata, J. Knockel, S. McKune, A. Senft, D. Tseng, and G. Wiseman. Chat program censorship and surveillance in China: Tracking TOM-Skype and Sina UC. *First Monday*, 18(7), 2013.

[11] J. R. Crandall, D. Zinn, M. Byrd, E. T. Barr, and R. East. ConceptDoppler: A weather tracker for Internet censorship. In *ACM Conference on Computer and Communications Security*, pages 352–365, 2007.

[12] J. Dalek, B. Haselton, H. Noman, A. Senft, M. Crete-Nishihata, P. Gill, and R. J. Deibert. A method for identifying and confirming the use of URL filtering products for censorship. In *Internet Measurement Conference (IMC)*. ACM, 2013.

[13] D. Dittrich and E. Kenneally. The Menlo Report: Ethical principles guiding information and communication technology research. Technical report, U.S. Department of Homeland Security, 2012.

[14] L. Dixon, T. Ristenpart, and T. Shrimpton. Network traffic obfuscation and automated Internet censorship. *IEEE Security & Privacy*, 14(6):43–53, Nov.–Dec. 2016.

[15] Z. Durumeric, E. Wustrow, and J. A. Halderman. ZMap: Fast internet-wide scanning and its security applications. In *22nd USENIX Security Symposium*, pages 605–620, 2013.

[16] D. Eastlake 3rd. Transport layer security (TLS) extensions: Extension definitions. RFC 6066, Jan. 2011.

[17] R. Ensafi, J. Knockel, G. Alexander, and J. R. Crandall. Detecting intentional packet drops on the Internet via TCP/IP side channels. In *International Conference on Passive and Active Network Measurement*, pages 109–118. Springer, 2014.

[18] R. Ensafi, P. Winter, A. Mueen, and J. R. Crandall. Analyzing the Great Firewall of China over space and time. *Proceedings on Privacy Enhancing Technologies*, 2015.

[19] A. Filastò and J. Appelbaum. OONI: Open Observatory of Network Interference. In *2nd USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2012.

[20] FortiNet. Fortiguard labs web filter. https://fortiguard.com/webfilter.

[21] Freedom House. Freedom on the net 2016, November 2016.

[22] K. Fu, C. Chan, and M. Chau. Assessing censorship on microblogs in China: Discriminatory keyword analysis and the real-name registration policy. *IEEE Internet Computing*, 17(3):42–50, 2013.

[23] G. Gebhart and T. Kohno. Internet censorship in Thailand: User practices and potential threats. In *IEEE European Symposium on Security and Privacy (EuroS&P)*, 2017.

[24] D. Gueorguiev, L. Shao, and C. Crabtree. Blurring the lines: Rethinking censorship under autocracy. 2017.

[25] B. Jones, R. Ensafi, N. Feamster, V. Paxson, and N. Weaver. Ethical concerns for censorship measurement. In *ACM SIGCOMM Conference*, pages 17–19, 2015.

[26] B. Jones, T.-W. Lee, N. Feamster, and P. Gill. Automated detection and fingerprinting of censorship block pages. In *Internet Measurement Conference (IMC)*. ACM, 2014.

[27] J. Knockel, J. R. Crandall, and J. Saia. Three researchers, five conjectures: An empirical analysis of TOM-Skype censorship and surveillance. In *FOCI*, 2011.

[28] R. MacKinnon. China's censorship 2.0: How companies censor bloggers. *First Monday*, 14(2), 2009.

[29] MaxMind. https://www.maxmind.com/.

[30] National Commission for the Protection of Human Subjects of Biomedical and Behavioral Research. *The Belmont Report: Ethical Principles and Guidelines for the Protection of Human Subjects of Research*. 1978.

[31] OpenNet Initiative. Jordan, August 2009. https://opennet.net/research/profiles/jordan.

[32] OpenNet Initiative. South Korea, August 2012. https://opennet.net/research/profiles/south-korea.

[33] C. Partridge and M. Allman. Addressing ethical considerations in network measurement papers. In *Workshop on Ethics in Networked Systems Research (NS Ethics@ SIGCOMM)*, 2015.

[34] P. Pearce, R. Ensafi, F. Li, N. Feamster, and V. Paxson. Augur: Internet-wide detection of connectivity disruptions. In *IEEE Symposium on Security and Privacy*, May 2017.

[35] P. Pearce, B. Jones, F. Li, R. Ensafi, N. Feamster, N. Weaver, and V. Paxson. Global measurement of DNS censorship. In *26th USENIX Security Symposium*, Aug. 2017.

[36] J. Postel. Discard protocol. RFC 863, May 1983.

[37] J. Postel. Echo protocol. RFC 862, May 1983.

[38] J. Postel and J. Reynolds. Telnet echo option. RFC 857, 1983.

[39] University of Oregon Route Views Project. www.routeviews.org.

[40] M. J. Salganik. *Bit by Bit: Social Research in the Digital Age*. Princeton University Press, 2017.

[41] W. Scott, T. Anderson, T. Kohno, and A. Krishnamurthy. Satellite: Joint analysis of CDNs and network-level interference. In *USENIX Annual Technical Conference (ATC)*, pages 195–208, 2016.

[42] R. Seggelmann, M. Tuexen, and M. Williams. Transport layer security (TLS) and datagram transport layer security (DTLS) heartbeat extension. RFC 6520, Feb. 2012.

[43] D. Springall, Z. Durumeric, and J. A. Halderman. FTP: The forgotten cloud. In *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 503–513, June 2016.

[44] The Tor Project. OONI: Open observatory of network interference. https://ooni.torproject.org/.

[45] United States Department of State. Ghana 2016 human rights report, 2016. http://www.state.gov/j/drl/rls/hrrpt/humanrightsreport/index.htm?year=2016&dlid=265260.

[46] P. Winter and S. Lindskog. How the Great Firewall of China is blocking Tor. In *Free and Open Communications on the Internet (FOCI)*. USENIX, 2012.

[47] X. Xu, Z. M. Mao, and J. A. Halderman. Internet censorship in China: Where does the filtering occur? In *Intl. Conference on Passive and Active Measurement (PAM)*, pages 133–142, 2011.

[48] T. Zhu, D. Phipps, A. Pridgen, J. R. Crandall, and D. S. Wallach. The velocity of censorship: High-fidelity detection of microblog post deletions. In *USENIX Security Symposium*, pages 227–240, 2013.

[49] J. Zittrain and B. Edelman. Internet filtering in China. *IEEE Internet Computing*, 7(2):70–77, 2003.

Appendix

Countries Tested Our test of all Citizen Lab domains completed against at least 15 servers in these countries:

Argentina, Australia, Austria, Bangladesh, Belarus, Belgium, Bolivia, Brazil, Bulgaria, Canada, Chile, China, Colombia, Croatia, Czechia, Denmark, Ecuador, Egypt, Finland, France, Georgia, Germany, Ghana, Greece, Hashemite Kingdom of Jordan, Hong Kong, Hungary, India, Indonesia, Iran, Ireland, Israel, Italy, Japan, Kazakhstan, Kenya, Kuwait, Malaysia, Mexico, Mongolia, Montenegro, Netherlands, New Zealand, Nigeria, Norway, Pakistan, Panama, Peru, Philippines, Poland, Portugal, Republic of Korea, Romania, Russia, Saudi Arabia, Serbia, Singapore, Slovak Republic, Slovenia, South Africa, Spain, Sweden, Switzerland, Taiwan, Thailand, Tunisia, Turkey, Ukraine, United Arab Emirates, United Kingdom, United States, Uzbekistan, Venezuela, and Vietnam.

Domain Classifications Below are the definitions for website classes as specified by the CLBL [8]:

Class	Definition
ANON	Tools used for anonymization, circumvention
COMT	Individual and group communications tools
DATE	Online dating services
FILE	Tools used to share files
GMB	Online gambling sites
GRP	Social networking tools and platforms
HACK	Sites dedicated to computer security
LGBT	Gay-lesbian-bisexual-transgender queer issues
MISC	Miscellaneous
MMED	Video, audio or photo sharing platforms
NEWS	Major, regional, and independent news outlets
POLR	Content that offers critical political viewpoints
PORN	Hard-core and soft-core pornography
SRCH	Search engines and portals