University of Massachusetts Amherst

ScholarWorks@UMass Amherst

Doctoral Dissertations | Dissertations and Theses

June 2022

MEASURING NETWORK INTERFERENCE AND MITIGATING IT WITH DNS ENCRYPTION

Seyed Arian Akhavan Niaki
University of Massachusetts Amherst

Follow this and additional works at: https://scholarworks.umass.edu/dissertations_2

Part of the OS and Networks Commons, and the Other Computer Sciences Commons

Recommended Citation

Akhavan Niaki, Seyed Arian, "MEASURING NETWORK INTERFERENCE AND MITIGATING IT WITH DNS ENCRYPTION" (2022). *Doctoral Dissertations*. 2495.
https://doi.org/10.7275/28406713 https://scholarworks.umass.edu/dissertations_2/2495

This Open Access Dissertation is brought to you for free and open access by the Dissertations and Theses at ScholarWorks@UMass Amherst. It has been accepted for inclusion in Doctoral Dissertations by an authorized administrator of ScholarWorks@UMass Amherst. For more information, please contact scholarworks@library.umass.edu.

MEASURING NETWORK INTERFERENCE AND MITIGATING IT WITH DNS ENCRYPTION

A Dissertation Presented

by

SEYED ARIAN AKHAVAN NIAKI

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2022

Robert and Donna Manning College of Information and Computer Sciences.

© Copyright by Seyed Arian Akhavan Niaki 2022

All Rights Reserved

MEASURING NETWORK INTERFERENCE AND MITIGATING IT WITH DNS ENCRYPTION

A Dissertation Presented

by

SEYED ARIAN AKHAVAN NIAKI

Approved as to style and content by:

Phillipa Gill, Chair

Amir Houmansadr, Member

Deepak Ganesan, Member

Jeremy Gummeson, Member

James Allan, Chair of the Faculty
Robert and Donna Manning College of
Information and Computer Sciences.

DEDICATION

to my family.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, Phillipa Gill, for her guidance and support. She always ensured that I had access to all the required resources for my research, introduced me to many of my collaborators, and always provided great feedback. I am genuinely grateful for that. This dissertation also owes a lot to my collaboration with Nguyen Phong Hoang. Your persistence and creativity in coming up with research problems to study are inspirational.

I would also like to thank my committee members, Deepak Ganesan, Amir Houmansadr, and Jeremy Gummeson, for their feedback on the work presented in this thesis.

Throughout this journey, I gained a lot of experience in collaborating with some amazing people on a variety of projects, Alan Mislove, Amir Houmansadr, Arash Molavi Kakhki, Bill Marczak, David Choffnes, Fangfan Li, Jakub Dalek, Jeffrey Knockel, Johanna Amann, Masashi Crete-Nishihata, Michalis Polychronakis, Nguyen Phong Hoang, Nicholas Weaver, Nicolas Christin, Pellaeon Lin, Sahand Farhoodi, Shinyoung Cho, Srikanth Sundaresan, and Zachary Weinberg. It truly was a privilege working with them; this work would not be possible without their assistance. I especially thank Abbas Razaghpanah and Narseo Vallina-Rodriguez for their mentorship in my first years as a graduate student and for pushing me to be a better researcher.

I thank all my lab-mates for the fruitful discussions and for providing a fun environment to work in: Rachee Singh, Abbas Razaghpanah, Shinyoung Cho, Albert Williams, Arun Dunna, Jerry Chai, Zack Weinberg, Nguyen Phong Hoang, and Rishab Nithyanand.

I owe a lot to my friends for being my family away from home. I am very fortunate to have all of you in my life: Abbas, Ahmad, Amirhossein, Ali, AliBana, Alireza, AlirezaS, Elmira, Erfan, Fatemeh, Hamed, Hamid, Hojjat, Kimia, Mahsa, MaryamA, MaryamR, Milad, Mohammad, Navid, Negara, Pardis, Pegah, Sadegh, Sahand, Shaghayegh, Shahrzad, Siavash, Soha, Yasaman, and many more.

Most importantly, I would like to thank my family. Mom, Dad, Armin, thank you for your love and support. I have learned many things from you, and I am who I am because of you. I appreciate everything you have done for me. Finally, I would like to express my gratitude to my best friend and wife, Negar. Thank you for being the best partner I could ever wish for; thank you for sticking with me through life's ups and downs. I am forever grateful to have you in my life.

This dissertation was supported by the Manning College of Information and Computer Sciences Dissertation Writing Fellowship.

ABSTRACT

MEASURING NETWORK INTERFERENCE AND MITIGATING IT WITH DNS ENCRYPTION

MAY 2022

SEYED ARIAN AKHAVAN NIAKI

B.Sc., SHARIF UNIVERSITY OF TECHONOLOGY

M.Sc., UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Phillipa Gill

The Internet has emerged as one of the most important tools of communication. With around 4.5 billion active users as of July 2020, it provides people the opportunity to access a vast treasure trove of information and express their opinions online. However, some countries consider the Internet as a critical communication medium and attempt to deploy network interference strategies. National governments, in particular, are notorious for their attempts to impose restrictions on online communication. Further, certain Internet service providers (ISPs) have been known to throttle specific applications and violate net neutrality principles.

Alongside the proliferation of network interference and an increasing awareness of the security and privacy of users over the Internet, we have seen a rise in the usage of network traffic encryption technologies. However, even with encryption enabled, network interference is still possible due to the information leakage of the DNS and TLS protocols. To this end, a rich ecosystem of DNS/TLS improvements has come to light with the purpose of improving user privacy by obfuscating the domains a user visits. These protocols have the potential to render certain forms of censorship ineffective.

In this dissertation, I will describe my contributions towards understanding network interference, including Internet censorship, as well as the throttling of specific network applications (traffic differentiation). I develop a network measurement platform that enables monitoring of network interference globally on an ongoing basis. I then focus on understanding the DNS censorship behavior of the Great Firewall of China (GFW) by leveraging remote network measurement techniques. Additionally, I investigate the prevalence of traffic differentiation practices and how they impact popular video streaming applications. I demonstrate that network interference is prevalent even with encryption enabled. This has led to the development of DNS and TLS improvements that aim to enhance user privacy and security. I review two recent proposals, namely DNS over HTTPS/TLS (DoH/DoT) and Encrypted Server Name Indication (ESNI), and investigate their potential to mitigate network interference and improve user privacy.

BIBLIOGRAPHICAL NOTES

Chapter 2 presents work done on globally studying Internet censorship in collaboration with Shinyoung Cho, Zachary Weinberg, Nguyen Phong Hoang, Abbas Razaghpanah, Nicolas Christin, and Phillipa Gill which was published in IEEE Security and Privacy 2020 [182]. Chapter 2 also presents two papers on studying the DNS censorship behavior of the GFW. Triplet Censors [183] in collaboration with Anonymous, Nguyen Phong Hoang, Phillipa Gill, and Amir Houmansadr which was published in FOCI 2020 and GFWatch [136] in collaboration with Nguyen Phong Hoang, Jakub Dalek, Jeffery Knockel, Pellaeon Lin, Bill Marczak, Masashi Crete-Nishihata, Phillipa Gill, and Michalis Polychronakis which was published in USENIX Security 2021.

Chapter 3 presents work done on studying traffic differentiation in collaboration with Fangfan Li, David Choffnes, Phillipa Gill, and Alan Mislove. This work was published in SIGCOMM'19 [165].

Chapter 4 presents work done on studying the privacy benefits of domain name encryption technologies in collaboration with Nguyen Phong Hoang, Nikita Borisov, Phillipa Gill, and Michalis Polychronakis. This work was published in ASIACCS'20 [129].

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Since its early days, the Internet has proven to be one of the most influential media allowing people around the world to access information and content, communicate with each other, and express their opinions online. However, governments and oppressive regimes have attempted to impose restrictions on online communication by censoring or disrupting the connections over the Internet [81], and certain Internet Service Providers (ISPs) deploy traffic differentiation techniques to throttle specific applications [165].

Network interference occurs when an adversary interferes with the communication between a user and a server, often disrupting the communication. Internet censorship and traffic differentiation are some instances of network interference over the Internet. In response, security researchers develop new network interference evasion strategies. These ongoing efforts to evade network interference have led to a cat-and-mouse game, as new evasion techniques are developed [41], and adversaries continue to detect and defeat these evasion techniques and find new ways to interfere with the communication. Developing more robust techniques to evade network interference requires that we first understand how network interference is being deployed. In this dissertation, I focus on developing measurement techniques and platforms that study network interference on a global scale and how they impact users.

With the proliferation of network interference and Internet surveillance in recent years [108], users have become more aware of their security and privacy over the Internet. This has led to the development of privacy-enhancing technologies and a rise in the usage of network traffic encryption. The number of Transport Layer Security (TLS) certificates issued by Let's Encrypt increased from 100 million to one billion from 2017 to 2020 [18]. Nevertheless, due to the information leakage of the DNS protocol, network interference is still feasible. To this end, a rich ecosystem of DNS improvements has been proposed to improve user privacy by obfuscating the domains a user visits. These protocols have the potential to render certain forms of censorship ineffective. In the final part of this thesis, I perform a longitudinal measurement study to investigate the potential of these protocols to improve user privacy and mitigate network interference.

1.1 Challenges

I now review the challenges to be overcome with the goal of better understanding how network interference is deployed and how domain name encryption technologies can help in mitigating it.

1.1.1 Network Interference

The literature is rich with studies of various aspects of Internet censorship [29, 30, 31, 32, 37, 47, 50, 69, 76, 77, 99, 100, 103, 116, 117, 144, 173, 186, 193, 195, 196, 222, 235, 239, 240, 259], but a global, longitudinal baseline of censorship covering a variety of censorship methods remains elusive. Most studies have been limited to a short period of time and/or a few countries; the few exceptions have traded off detail for breadth of coverage. Furthermore, the most recent large-scale audits of net neutrality practices were published a decade ago. These studies were focused on backbone networks [261] and targeted specific protocols [85]. Due to the shift towards mobile Internet users, I argue that a global, longitudinal monitoring of traffic differentiation practices over the Internet is necessary. I highlight three key challenges that must be addressed for studying network interference:

Access to vantage points. With few exceptions [1], measuring Internet censorship requires access to "vantage point" hosts within the region of interest. Similarly, in order to determine traffic differentiation policies, access to vantage points that can replay a recorded network trace is needed. With the shift towards mobile Internet users, it is crucial to study traffic differentiation on cellular networks. The simplest way to obtain vantage points is to recruit volunteers [103, 116, 223, 242]. Volunteers can run software that performs network measurements from each vantage point, but recruiting more than a few volunteers per country and retaining them for long periods is difficult. Further, volunteers may be exposed to personal risks for participating in censorship research. More recently, for studying Internet censorship, researchers have explored alternatives, such as employing DNS resolvers [196, 222], echo servers [240], web browsers visiting instrumented websites [50], and TCP side channels [97, 195]. These alternatives reduce the risk to volunteers and can achieve broader, longer-term coverage than volunteer labor. However, they cannot perform arbitrary network measurements; for instance, open DNS resolvers can only reveal DNS-based censorship.

Understanding what to test. Testing a single blocked URL can reveal that a censorship system exists within a country, but does not reveal the details of the censorship policy, how aggressively it is enforced, or all of the blocking techniques used. Even broad test lists, like those maintained by the Citizen Lab [67], may be insufficient [79]. Web pages are often short-lived, so tests performed in the present may be misleading [250].

Similar to the censorship case, for traffic differentiation, testing a single application can reveal that an ISP is deploying traffic differentiation. However, it does not reveal the policies of the ISP towards different applications. For instance, an ISP could use different rate limits and target a different set of applications.

[1] China filters inbound as well as outbound traffic, making external observation simpler.

Reliable detection. Censors can prevent access to content in various ways. For instance, censors may choose to overtly censor some material by presenting "block pages", or they can covertly censor other material by mimicking site outages [78, 116]. Many recent studies focus on a single technique [50, 97, 195, 196, 222, 230, 240]. This is valuable but incomplete, because censors may combine different techniques to filter different types of content. As the Internet evolves and new modes of access appear (e.g., mobile devices), censorship evolves as well, and monitoring systems must keep up [29, 178]. Ad-hoc detection strategies without rigorous evaluation are prone to false positives [259]. For instance, the detection of block pages requires taking regional differences in content into account [144].

Previous work on traffic differentiation focused on detecting differentiation on a per-device basis [178]. This technique is known as "Record and replay". However, individual tests are subject to confounding factors such as transient periods of poor network performance. In contrast, by leveraging test results from multiple users in the same ISP, identifying fixed-rate throttling can be done more accurately.

1.1.2 Domain Name Encryption Technologies

The domain name encryption technologies discussed, namely, DoH [137], DoT [140] and the ESNI [214] extension have been recently proposed and are in their early stages of adoption. As a result, there exist some challenges that need to be addressed before they can be beneficial to the community. I overview three of the challenges of domain name encryption technologies.

Full domain name confidentiality. In today's Internet, plaintext domain names are exposed through two channels: the SNI extension in TLS and traditional DNS name resolutions. The deployment of DoH/DoT is thus a prerequisite for ESNI. Equivalently, the use of DoH/DoT will not provide any meaningful privacy if domain names are still exposed through the (unencrypted) SNI extension in TLS handshake traffic. Recently, there is a push for the deployment of DoH/DoT, with major organizations already supporting it (e.g., Google, Cloudflare, Firefox), though this has not been followed by an equivalent effort for the deployment of ESNI. Unless the confidentiality of domain names is preserved on both channels (TLS and DNS), neither technology can provide any actual privacy benefit if deployed individually.

Trusting third-parties. Although the benefits of DoH/DoT against last-mile adversaries are clear, this comes with the cost of *fully trusting* a third-party operator of the DoH/ DoT resolver on which users have outsourced all their DNS resolutions [134]. Several companies already offer public DoH/DoT resolvers, including Google [118, 119] and Cloudflare [122]. One possible way to decentralized encrypted DNS resolution is to distribute DNS queries across multiple DoH/DoT resolvers.

Hosting Providers. Hosting and CDN providers are in a more privileged position to achieve meaningful impact in helping improve the potential privacy benefits of ESNI, as they can control the number of co-hosted domains per IP address, and the frequency of IP addresses rotation. Unless website owners prefer otherwise, providers could group more websites under the same IP address (which, understandably, may not be desirable for some websites). To improve user privacy, providers should cluster websites according to similarities in terms of traffic patterns and popularity ranking, to hinder website fingerprinting attempts.

1.2 Contributions

This dissertation tackles the problem of network interference and how recent domain name encryption technologies can mitigate it and improve user privacy. In order to do so, I first develop network measurement techniques and platforms that are able to overcome the current challenges of accurately measuring network interference, such as Internet censorship and traffic differentiation. Then I investigate the privacy benefits of domain name encryption technologies.

I present the system architecture of ICLab, a global longitudinal censorship measurement platform. ICLab primarily uses commercial Virtual Private Network servers (VPNs) as vantage points, after validating that they are in their advertised locations. Using the data collected by ICLab, I detect a variety of censorship mechanisms deployed in different regions and how censors treat different types of content differently. Along the same lines, I will present our longitudinal study of the GFW's DNS injection behavior, where I reveal several previously-unknown properties of China's filtering system. I then present Wehe, a mobile application that allows for large-scale crowdsourced measurements for traffic differentiation. Using the combined data collected by Wehe, I form high-confidence inferences of differentiation practices. Further, I investigate how throttling impacts video streaming resolution for popular video streaming providers. Next, I evaluate the privacy benefits of the recent domain name encryption technologies by using DNS data collected through active DNS measurements from nine countries and over a period of two months. I study the implications of the co-hosting degree of the current web on ESNI's privacy benefits.

Specifically, I make the following contributions:

A Global, Longitudinal Internet Censorship Measurement Platform. I present ICLab, an Internet measurement platform specialized for censorship research. It achieves a new balance between breadth of coverage and detail of measurements, by using commercial VPNs as vantage points distributed around the world. ICLab has been operated continuously since late 2016. It can currently detect DNS manipulation and TCP packet injection, and overt "block pages" however they are delivered. ICLab records and archives raw observations in detail, making retrospective analysis with new techniques possible. At every stage of processing, ICLab seeks to minimize false positives and manual validation. Within 53,906,532 measurements of individual web pages, collected by ICLab in 2017 and 2018, I observe blocking of 3,602 unique URLs in 60 countries. Using this data, I compare how different blocking techniques are deployed in different regions and/or against different types of content. Our longitudinal monitoring pinpoints changes in censorship in India and Turkey concurrent with political shifts, and our clustering techniques discover 48 previously unknown block pages. ICLab's broad and detailed measurements also expose other forms of network interference, such as surveillance and malware injection (Chapter 2).

Demystifying Great Firewall's DNS Censorship Behavior. I analyze the DNS injection behavior of the GFW over a period of nine months using the Alexa top 1M domains as a test list. I first focus on understanding the publicly routable IPs used by the GFW in forged DNS responses and observe groups of IPs used to filter specific sets of domains. I also see a sharp decline in public IPs injected by the GFW in November 2019. Then, I fingerprint three different injectors that were observed in our measurements. Notably, one of these injectors mirrors the IP TTL value from probe packets in its injected packets which has implications for the use of TTL-limited probes for localizing censors. Finally, I confirm that the observations generally hold across IP prefixes registered in China (Chapter 2).

How Great is the Great Firewall? Measuring China's DNS Censorship. I introduce GFWatch, a large-scale, longitudinal measurement platform capable of testing hundreds of millions of domains daily, enabling continuous monitoring of the GFW's DNS filtering behavior. I present the results of running GFWatch over a nine-month period, during which I tested an average of 411M domains per day and detected a total of 311K domains censored by GFW's DNS filter. I further reverse engineer regular expressions used by the GFW and find 41K innocuous domains that match these filters, resulting in overblocking of their content. I also observe bogus IPv6 and globally routable IPv4 addresses injected by the GFW, including addresses owned by US companies, including Facebook, Dropbox, and Twitter. Using data from GFWatch, I studied the impact of GFW blocking on the global DNS system. I found 77K censored domains with DNS resource records polluted in popular public

DNS resolvers, such as Google and Cloudflare. Finally, I propose strategies to detect poisoned responses that can (1) sanitize poisoned DNS records from the cache of public DNS resolvers, and (2) assist in the development of circumvention tools to bypass the GFW's DNS censorship (Chapter 2).

A large-scale analysis of deployed traffic differentiation practices. I conducted a one-year study of content-based traffic differentiation policies deployed in operational networks, using results from 1,045,413 crowdsourced measurements conducted by 126,249 users across 2,735 ISPs in 183 countries/regions. I develop and evaluate a methodology that combines individual per-device measurements to form high-confidence, statistically significant inferences of differentiation practices, including fixed-rate bandwidth limits (i.e., throttling) and delayed throttling practices. Using this approach, I identify differentiation in both cellular and WiFi networks, comprising 30 ISPs in 7 countries. I also investigate the impact of throttling practices on video streaming resolution for several popular video streaming providers. I find that throttling limits the maximum video resolution, but interestingly, apps' default settings and available resolutions also play a significant role (Chapter 3).

Assessing the Privacy Benefits of Domain Name Encryption. I assess the privacy benefits of two recent proposals, namely, DNS over HTTPS/ TLS (DoH/DoT) and Encrypted SNI (ESNI), by considering the relationship between hostnames and IP addresses, the latter of which are still exposed. I perform DNS queries from nine vantage points around the globe to characterize this relationship. I quantify the privacy gain offered by ESNI for different hosting and CDN providers using two different metrics, the k-anonymity degree due to co-hosting and the dynamics of IP address changes. I find that 20% of the domains studied will not gain any privacy benefit since they have a one-to-one mapping between their hostname and IP address. On the other hand, 30% will gain a significant privacy benefit with a k value greater than 100, since these domains are co-hosted with more than 100 other domains.

Domains whose visitors' privacy will meaningfully improve are far less popular, while for popular domains the benefit is not significant. Analyzing the dynamics of IP addresses of long-lived domains, I find that only 7.7% of them change their hosting IP addresses on a daily basis. I conclude by discussing potential approaches for website owners and hosting/CDN providers for maximizing the privacy benefits of ESNI (Chapter 4).

The work detailed in this dissertation addresses the challenges of studying Internet censorship and traffic differentiation longitudinally and globally, and discusses the privacy benefits of DoH/DoT and ESNI.

CHAPTER 2

NETWORK INTERFERENCE - INTERNET CENSORSHIP

2.1 Background

The Internet is crucial in today's social and political movements by allowing people to communicate with each other and practice their freedom of speech. Consequently, oppressive regimes in different parts of the world monitor and restrict access to the Internet. This is broadly known as Internet Censorship. We briefly review the techniques that these censors use to block access to information online, the different types of censors, and two different options for implementation as these are important in developing a measurement platform to study Internet censorship.

2.1.1 Network-level Blocking Techniques

All attempts to interfere with website access can be considered as man-in-the-middle (MITM) attacks on communications between a web browser and a web server. Depending on the location and configuration of their MITM devices, censors may interfere with traffic outside the borders of their own authority [47, 100, 239].

DNS manipulation. DNS resolution is the first step a web browser takes when visiting a website. It involves resolving the domain name to its IP address. DNS was not designed with security and privacy in mind, thus, DNS traffic is unencrypted and less than 1% of it is authenticated [243]. Censors can forge response DNS packets with non-routable IP addresses, the address of a server controlled by the censor or error codes such as "host not found" (NXDOMAIN). The GFW is one such censor that injects forged DNS responses to sensitive DNS queries.

IP-based blocking. Once the web browser obtains the IP address of a web server, it proceeds to establish a TCP connection to that server. Censors can drop packets destined for an IP address known to host censored content, reply with a TCP reset packet, or reroute them to a server controlled by the censor [131].

TCP packet injection. Some censors allow for the TCP handshake to be completed, and then inject a packet into the TCP stream, either breaking the connection before the response arrives [247], or superseding the first response from the legitimate server. This provides the opportunity for the censor to observe the first HTTP request sent by the client, and thus be able to block access to individual pages [72].

Transparent proxy. Censors can use a "transparent proxy" that intercepts all HTTP traffic leaving a country, and after decoding the content, they can choose whether or not to forward it [77]. However, since these proxies act as TCP peers that may modify HTTP traffic passing through them, they can be detectable [248]. These proxies allow a censor to make fine-grained decisions on how to block access to content, but they can only be used on HTTP traffic.

2.1.2 On-path and In-path Censors

An on-path censor observes a copy of all traffic passing through a network link. Thus, it cannot modify or discard packets that are already within the flow. It can only react by injecting packets into the link. These types of censors are relatively cheap and easy to deploy, however, they are easily detectable since injected packets appear alongside the legitimate traffic.

On the other hand, in-path censors operate on the actual traffic passing through a network link. Therefore, they can inject, modify, or discard packets. In order for in-path equipment to perform such actions, they need to operate at the line-rate of the backbone router, so they are more expensive, can have limited features, but are harder to detect.

2.1.3 Overt and Covert Censorship

Censors can opt to use either over or covert censorship depending on their intentions. In overt censorship, the censor blocks access to a website by sending a "block page" to the user instead of the material of the website. This can be done using a transparent HTTP proxy, an injected TCP packet, or a DNS response that redirects the web browser to a server controlled by the censor. In contrast, a covert censor causes a network error that could have occurred for other reasons (such as common network errors), thus avoids informing the user that the material was censored. Covert censorship can be accomplished by a transparent HTTP proxy, an injected TCP reset packet, an injected DNS error or non-routable IP address, or by discarding packets. Censors can block different categories of content in different ways. For instance, Yemen has been observed to overtly block pornography, and covertly block disfavored political content [116].

2.2 Related Work

The Internet filtering infrastructure of China, allegedly designed in the late 90s under the Golden Shield project [224, 258], is a system used by the Chinese government to regulate the country's domestic Internet access. The filtering system, commonly referred to as the Great Firewall [114], consists of middleboxes distributed across border autonomous systems [32, 72, 257], which are controlled in a centralized fashion [48, 80, 114, 168, 224, 266]. There are several filtering modules developed to control the free flow of information at different layers of the network stack, including TCP/IP packet filtering [69, 88, 99, 187, 193, 253] and application-level keyword-based blocking [69, 114, 207, 266].

Unencrypted and unauthenticated DNS traffic is widely targeted by censorship systems to interrupt communications between users and remote destinations where censored content or services are hosted [86, 182, 196, 222, 229]. Exploiting DNS

Table 2.1. A high-level comparison of censored domains and forged IPs detected by different studies/platforms. (*) The number of forged IPs from Satellite and OONI includes "anomalies" due to domains hosted on CDNs and localized filtering policies.

Study/Platform	**Duration**	**Longitudinal**	**Tested Domains**	**Censored Domains**	**Forged IPs**	**Common Forged IPs**
Zittrain et al. [266]	Mar 2002 - Nov 2002	○	204K	1K	1	1
Lowe et al. [168]	2007	○	951	393	21	3
Brown et al. [48]	Nov 2010	○	1	1	9	6
CCR'12 [229]	Nov 2011	○	10	6	28	
FOCI'14 [32]	Aug 2013 - Apr 2014	○	130M	35.3K	174	
Triplet Censors [183]	Sep 2019 - May 2020	○	1M	24.6K	1,510	1,462
OONI [103]	Apr 2020 - Dec 2020	●	3.3K	460	*710	593
Satellite [222]	Apr 2020 - Dec 2020	●	3.5K	375	*2,391	1,613
GFWatch	Apr 2020 - Dec 2020	●	534M	311K	1,781	-

insecurity, the GFW is designed as an on-path/man-on-the-side (MotS) system which takes advantage of UDP-based DNS resolution to inject fake responses when censored domains are detected in users' DNS queries. In its early days, the GFW only used a handful of forged IPs [168, 266]. However, later studies have noticed an increase in the number of forged IPs, from nine in 2010 [48], 28 in 2011 [229], 174 in 2014 [32], to more than 1.5K recently [183]. Except for [229] and [32] whose authors preferred to remain anonymous and the dataset URLs provided in their papers are no longer accessible, we were able to obtain data from other studies for comparison (Table 2.1). A common drawback of these studies is that their experiments are conducted only over limited time periods and the test domains are also static, i.e., obtained from a snapshot of Alexa top list or zone files.

Other countries receiving case studies include Iran [29, 37], India [259], Pakistan [152, 180], Syria [61], and Egypt and Libya [76]. Whenever researchers have had access to more than one vantage point within a country, they have found that the policy is not consistently enforced. There is always region-to-region and ISP-to-ISP variation.

Broader studies divide into two lines of research. One group of studies investigate worldwide variation in censorship: for instance, whether censorship mainly interferes with DNS lookups [196] or subsequent TCP connections, and whether the end-user is informed of censorship [125, 242]. In some cases, it has been possible to identify the specific software in use [77, 144]. Another line of work aims to understand what is censored and why [20, 49, 250], how that changes over time [30, 116], how people react to censorship [156, 256], and how the censor might react to being monitored [49].

Due to the difficulties with relying on volunteers, several groups of researchers have sought alternatives. CensMon [223] used Planet Lab nodes, Anderson et al. [30] used RIPE Atlas nodes, Pearce et al. [196] use open DNS resolvers and VanderSloot et al. [240] use open echo servers. Darer et al. [79] took advantage of the fact that the Chinese Great Firewall will inject forged replies to hosts located outside the country. Burnett and Feamster [50], Ensafi et al. [98], and Pearce et al. [195] all propose variations on the theme of using existing hosts as reflectors for censorship probes, without the knowledge of their operators, at different levels of the protocol stack.

Only a few studies have lasted more than a month. Five prominent exceptions are Encore [50], IRIS [196], OONI [103], Quack[240], Satellite [222], and Censored Planet [230] [1] all of which share goals similar to ICLab. Herdict [126] has also been active for years, but simply aggregates user reports of inaccessible websites. It does not test or report why the sites are inaccessible.

2.3 A Global, Longitudinal Internet Censorship Measurement Platform

Researchers have studied Internet censorship for nearly as long as attempts to censor contents have taken place. Most studies have however been limited to a short period of time and/or a few countries; the few exceptions have traded off detail for breadth of coverage. Collecting enough data for a comprehensive, global, longitudinal perspective remains challenging.

[1] Censored Planet combines the techniques described in Satellite and Quack.

In this chapter, we present ICLab, an Internet measurement platform specialized for censorship research. It achieves a new balance between breadth of coverage and detail of measurements, by using commercial VPNs as vantage points distributed around the world. ICLab has been operated continuously since late 2016. It can currently detect DNS manipulation and TCP packet injection, and overt “block pages” however they are delivered. ICLab records and archives raw observations in detail, making retrospective analysis with new techniques possible. At every stage of processing, ICLab seeks to minimize false positives and manual validation.

Within 53,906,532 measurements of individual web pages, collected by ICLab in 2017 and 2018, we observe blocking of 3,602 unique URLs in 60 countries. Using this data, we compare how different blocking techniques are deployed in different regions and/or against different types of content. Our longitudinal monitoring pinpoints changes in censorship in India and Turkey concurrent with political shifts, and our clustering techniques discover 48 previously unknown block pages. ICLab’s broad and detailed measurements also expose other forms of network interference, such as surveillance and malware injection.

2.3.1 Introduction

For the past 25 years, the Internet has been an important forum for people who wish to communicate, access information, and express their opinions. It has also been the theater of a struggle with those who wish to control who can be communicated with, what information can be accessed, and which opinions can be expressed. National governments in particular are notorious for their attempts to impose restrictions on online communication [81]. These attempts have had unintentional international consequences [31, 47, 173, 239], and have raised questions about export policy for network management products with legitimate uses (e.g., virus detection and protection of confidential information) that can also be used to violate human rights [77].

The literature is rich with studies of various aspects of Internet censorship [29, 30, 31, 32, 37, 47, 50, 69, 76, 77, 99, 100, 103, 116, 144, 173, 186, 193, 195, 196, 222, 235, 239, 240, 257, 259] but a global, longitudinal baseline of censorship covering a variety of censorship methods remains elusive. We highlight three key challenges that must be addressed to make progress in this space:

Challenge 1: Access to Vantage Points. With few exceptions,[2] measuring Internet censorship requires access to "vantage point" hosts within the region of interest.

The simplest way to obtain vantage points is to recruit volunteers [103, 116, 223, 242]. Volunteers can run software that performs arbitrary network measurements from each vantage point, but recruiting more than a few volunteers per country and retaining them for long periods is difficult. Further, volunteers may be exposed to personal risks for participating in censorship research.

More recently, researchers have explored alternatives, such as employing open DNS resolvers [196, 222], echo servers [240], Web browsers visiting instrumented websites [50], and TCP side channels [97, 195]. These alternatives reduce the risk to volunteers, and can achieve broader, longer-term coverage than volunteer labor. However, they cannot perform arbitrary network measurements; for instance, open DNS resolvers can only reveal DNS-based censorship.

Challenge 2: Understanding What to Test. Testing a single blocked URL can reveal that a censorship system exists within a country, but does not reveal the details of the censorship policy, how aggressively it is enforced, or all of the blocking techniques used. Even broad test lists, like those maintained by the Citizen Lab [67], may be insufficient [79]. Web pages are often short-lived, so tests performed in the present may be misleading [250].

[2] China filters inbound as well as outbound traffic, making external observation simpler.

Challenge 3: Reliable Detection. Censors can prevent access to content in several different ways. For instance, censors may choose to supply "block pages" for some material, which explicitly notify the user of censorship, and mimic site outages for other material (see §2.1.3) [78, 116].

Many recent studies focus on a single technique [50, 97, 195, 196, 222, 240]. This is valuable but incomplete, because censors may combine different techniques to filter different types of content.

As the Internet evolves and new modes of access appear (e.g., mobile devices), censorship evolves as well, and monitoring systems must keep up [29, 178]. Ad-hoc detection strategies without rigorous evaluation are prone to false positives [259]. For example, detecting filtering via DNS manipulation requires care to deal with CDNs [196, 222] and detection of block pages requires taking regional differences in content into account [144].

2.3.2 Contributions

We present ICLab, a censorship measurement platform that tackles these challenges. ICLab primarily uses commercial Virtual Private Network servers (VPNs) as vantage points, after validating that they are in their advertised locations. VPNs offer long-lived, reliable vantage points in diverse locations, but still allow detailed data collection from all levels of the network stack. ICLab also deploys volunteer-operated devices (VODs) in a handful of locations.

ICLab is extensible, allowing us to implement new experiments when new censorship technologies emerge, update the URLs that are tested over time, and re-analyze old data as necessary. To date ICLab has only been used to monitor censorship of the web, but it could easily be adapted to monitor other application-layer protocols (e.g., using techniques such as those in Molavi Kakhki et al. [178]). Besides ICLab itself, and its collected data, we offer the following contributions:

Global, longitudinal monitoring. Since its launch in 2016, ICLab has been continuously conducting measurements in 62 countries, covering 234 autonomous systems (ASes) and testing over 45,000 unique URLs over the course of more than two years. The platform has detected over 3,500 unique URLs blocked using a variety of censorship techniques. We discuss our discoveries in more detail in Section 2.3.5.

Enhanced detection accuracy. ICLab collects data from all levels of the network stack and detects multiple different types of network interference. By comparing results across all the detection techniques, we can discover inaccuracies in each and refine them. We have eliminated all false positives from our block page detector. DNS manipulation detection achieves a false positive rate on the order of 10^{-4} when cross-checked against the block page detector (see Section 2.3.4.1). Similar cross-checking shows a negligible false positive rate for TCP packet injection (see Section 2.3.4.2).

Semi-automated block page detection. We have developed a new technique for discovering both variations on known block pages and previously unknown block pages. These explicit notifications of censorship are easy for a human to identify, but machine classifiers have trouble distinguishing them from other short HTML documents expressing an error message. Existing systems rely on hand-curated sets of regular expressions, which are brittle and tedious to update.

ICLab includes two novel machine classifiers for short error messages, designed to facilitate manual review of groups of suspicious messages, rather than directly deciding whether each is a block page. Using these classifiers we discovered 48 previously undetected block page signatures from 13 countries. We describe these classifiers and their discoveries in more detail in Section 2.3.4.3.

2.3.3 System Architecture

ICLab is a platform for measuring censorship of network traffic. As shown in Figure 2.1, it consists of a central control server and a set of vantage points distributed worldwide. The central server schedules measurements for each vantage point to perform, distributes test lists, and collects measurement results for analysis. The vantage points send and receive network traffic to perform each measurement, and upload their observations to the central server. All analysis is done centrally after the measurements have completed. Raw observations, including complete packet logs, are archived so that new analysis techniques can be applied to old data. There are two types of vantage points: volunteer-operated devices (VODs) configured by us and installed in locations of interest by our volunteers,[3] and VPN-based clients, which forward traffic through commercial VPN proxies located in various countries.

2.3.3.1 Design Goals

We designed ICLab to achieve the following properties:

Global, continuous monitoring. The techniques used for Internet censorship, the topics censored, and the thoroughness with which censorship is enforced are known to vary both among [49, 77, 79, 116, 125, 196] and within [20, 98, 152, 255, 257] countries. Therefore, the system should operate vantage points in multiple locations within each of many countries, to produce a comprehensive global view of censorship. Censorship may ratchet upward over time [81, 105], may change abruptly in response to political events [76] and may even cease after governing parties change [116]. Therefore, the system should perform its measurements continuously over a period of years, to detect these changes as they happen.

Reproducible and extensible. The basic techniques for censoring network traffic (described in §2.1.1) are well-known [242, 247] but new variations appear regularly [29, 100]. The short lifetime of "long tail" content means that the current content of a website may bear no relationship to what it was when it was originally censored [250]. Therefore, the system needs to be extensible with new types of measurement, and

[3] Most of these are low-cost Raspberry Pi devices.

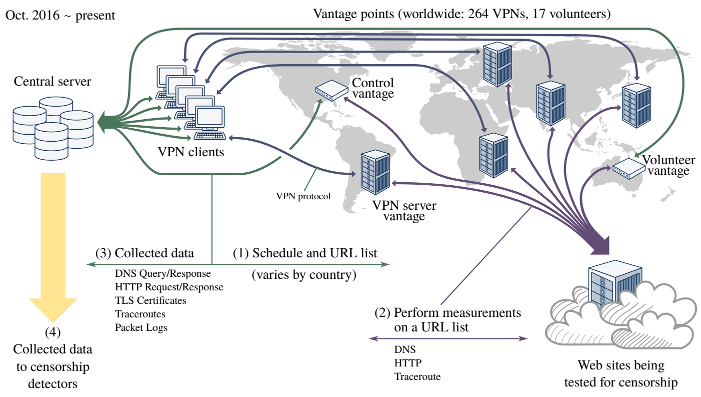


Figure 2.1. Architecture of ICLab. (1) The central server sends a measurement schedule along with an associated test list to vantage points. (2) The vantage points perform measurements. (3) Collected data is uploaded to the central server. (4) Censorship detection is done centrally.

should record as much information as possible with each measurement (e.g., packet traces and detailed contextual information).

Minimal risk to volunteers. Censorship monitoring involves accessing material that is forbidden in a particular country, from that country, and provoking a response from the censor. The response we expect is one of the MITM attacks described in §2.1.1, but legal or extralegal sanctions aimed at the volunteer operating the vantage point are also possible. The risk may be especially significant for volunteers already engaged in human rights reporting or advocacy. Use of commercial VPNs as vantage points is intended to mitigate these risks. VODs are only deployed in locations where we believe legal or extralegal sanctions are unlikely, and we obtain informed consent from the volunteers who operate them.

2.3.3.2 Vantage Points

Of ICLab's 281 vantage points, 264 are VPN-based, obtaining access to locations of interest via commercial VPN services. 17 vantage points are VODs. The measurement software is the same for both types of vantage; the only difference is that VPN-based vantages route their traffic through a VPN while performing measurements.

VPN-based vantages. ICLab uses VPN-based vantages whenever possible, because of their practical and ethical advantages. We do not need to recruit volunteers from all over the world, or manage physical hardware that has been distributed to them, but we still have unrestricted access to the network, unlike, for instance, phone or web applications [50, 103]. The VPN operator guarantees high availability and reasonable bandwidth, and they often offer multiple locations within a country. For 75% of the countries where we use VPN-based vantages, the VPNs give us access to at least two ASes within that country.

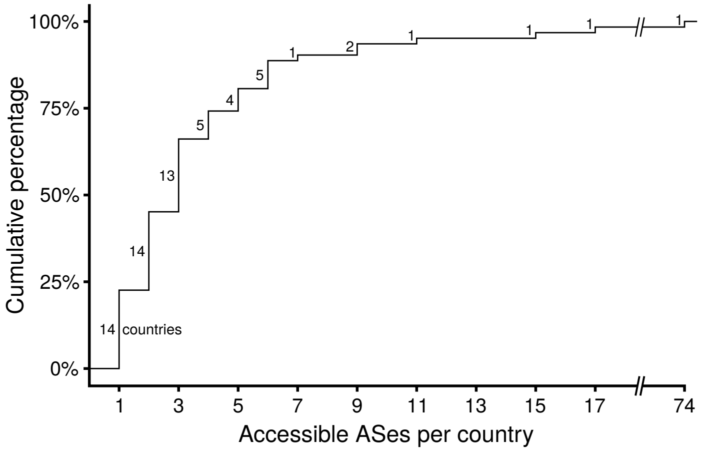


Figure 2.2. CDF of number of accessible AS(es) per country

Censorship policies are known to vary from region to region and network to network within a single country [20, 98, 152, 255, 257]. Therefore, comprehensive monitoring requires vantage points located in diverse locations within a country. Some VPN services offer servers in several physical locations within a single country, making this simple. Even when they don't advertise several physical locations, we have found that they often load-balance connections to a single hostname over IP addresses in several different ASes and sometimes different physical locations as well. When possible, we increase diversity further by subscribing to multiple VPN services. Figure 2.2 shows a CDF of the number of networks we can access in each country, combining all the above factors; we are able to access two or more networks in 75% of all countries, and three or more networks in 50%.

On the ethical side, a commercial VPN operator is a company that understands the risks of doing business in each country it operates in. It is unlikely that they would deploy a server in a country where the company or its employees might suffer legal or extralegal sanctions for the actions of its users.

A disadvantage of VPNs is that they only supply a lower bound on the censorship experienced by individuals in each country, because their servers are hosted in commercial data centers. There is some evidence that network censorship is less aggressively performed by data centers' ISPs than by residential ISPs [22, 257]. According to the CAIDA AS classification [53], 41% of the networks hosting our VPN-based vantages are "content" networks, which are the most likely to be subject to reduced levels of censorship. However, we have visibility into at least one other type of AS in 83% of the countries we can observe. In countries where we have both VPNs and VODs, we have observed identical block pages from both, indicating that all types of ASes are subject to similar blocking policies in those countries.

User-hosted VPNs (e.g., Geosurf [113], Hola [138], Luminati [171]) would offer access to residential ISPs, but ICLab does not use them, as they have all the ethical concerns associated with VODs, with less transparency. Also, there are reports of illicit actions by the operators of these VPNs, such as deploying their software as a viral payload, and facilitating distributed denial of service (DDoS) attacks [176], making it even more unethical to use these services.

Commercial VPN services cannot be relied on to locate all of their servers in the countries where they are advertised to be [251]. ICLab therefore checks the location of each VPN server before using it for measurements. We assume that packets are not able to travel faster than 153 km/ms ($0.5104\,c$) over long distances. We measure the round-trip time from each VPN server to a set of landmark hosts in known locations, drawn from the RIPE Atlas measurement constellation [217]. If any packet would have had to travel faster than 153 km/ms to reach the advertised country and return in the measured time, we assume the server is not in its advertised location, and we do not use it as a vantage point.

The VPN services we subscribe to collectively advertise endpoints in 216 countries. Our checker is only able to confirm the advertised location for 55 countries (25.5% of the total). Compared to the results reported by Weinberg et al. [251], who tackled the same problem with more sophisticated techniques, our method rejects significantly more proxies (10% more on average when we experiment across multiple providers). Possibly some of those proxies could be used after all, but we do not want to attribute censorship to the wrong country by accident, so we are being cautious.

Volunteer-operated device vantages. VODs are more difficult to keep running, and require a local volunteer comfortable with the risks associated with operating the device. Since ICLab does not collect personally identifiable information *about* the volunteers, our IRB has determined that this project is not human subjects research. However, we are guided by the principles of ethical human subjects research, particularly the need to balance potential benefits of the research against risks undertaken by volunteers. Most of our VODs have been deployed opportunistically through collaborations with NGOs and organizations interested in measuring Internet censorship from a policy perspective. For each deployed VOD, we maintain contact with the volunteer, and monitor the political situation in the country of deployment. We have deemed some countries too risky (for now) to recruit volunteers in (e.g., Iran, Syria).

Freedom House and Reporters Without Borders Scores. The international organization Freedom House, which promotes civil liberty and democracy worldwide, issues a yearly report on “freedom on the Net,” in which they rate 65 countries on the degree to which online privacy and free exchange of information online are upheld in that country [105]. Each country receives both a numerical score and a three-way

Table 2.2. Country Coverage of ICLab. The number of countries and ASes on each continent where we have vantage points with validated locations, since 2017. Oceania includes Australia. VPNs: virtual private network servers. VODs: volunteer-operated devices. NF, PF, F: of the countries with vantage points, how many are politically not free, partially free, or free.

Continent	VPNs	VODs	Countries	ASes	NF	PF	F
Asia	64	4	14/32	54	5	7	2
Africa	9	10	9/72	19	1	6	2
N. America	87	1	5/17	81	0	1	4
S. America	9	0	5/20	6	1	3	1
Europe	83	2	27/42	64	1	5	21
Oceania	12	0	2/ 6	11	0	0	2
Total	264	17	62/189	234	8	22	32

classification: 16 of the 65 countries are considered "free," 28 are "partly free," and 21 are "not free." Unfortunately, 33 of the countries studied by ICLab are not included in this report. The international organization Reporters Without Borders (RWB) issues a similar yearly report on freedom of the press. This report covers 189 countries and territories, including all 65 of the countries rated by Freedom House, and all 62 of the countries studied by ICLab [213] Each country receives a numerical score and a color code (best to worst: 16 countries are coded white, 42 yellow, 59 orange, 51 red, and 21 black). Press freedom is not the same as online freedom, and the methodologies behind the two reports are quite different, but the scores from the two reports are reasonably well correlated (Kendall's $\tau = 0.707$, $p \approx 10^{-16}$). We used a simple linear regression to map RWB scores onto the same scale as FH scores, allowing us to label all of the countries studied by ICLab as "free" (72), "partly free" (85), or "not free" (32) in the same sense used by Freedom House.

Breadth of coverage. As of this writing, ICLab has VPN-based vantage points in 55 countries, and volunteer-operated clients in 13 countries. 6 countries host both types of clients, so ICLab has vantage points in 62 countries overall. ICLab seeks to achieve both geographic and political diversity in its coverage. Table 2.2 summarizes our current geographic diversity by continent, and political diversity by a combination of two scores of political freedom, developed by Freedom House [105] and Reporters Without Borders [213].

It is easier to acquire access to vantage points in Europe, North America, and East Asia than in many other parts of the world. We have plans for expanded coverage in Africa and South America in the near future, via additional VPN services. It is also easier to acquire access to vantage points in “free” and “partially free” than “not free” countries, because it is often too risky for either VPN services or volunteer-operated devices to operate in “not free” countries. Expanding our coverage of “not free” countries is a priority for future development of ICLab, provided we can do it safely.

Internet censorship does happen in the “partly free” and “free” countries, and is not nearly as well documented as it is for specific “not free” countries (most notably China). Our broad coverage of these classes of countries gives us the ability to track changes over time, across the full spectrum of censorship policy, worldwide.

2.3.3.3 Test Lists

At present, ICLab’s measurements are focused on network-level interference with access to websites. ICLab’s vantage points test connectivity to the websites on three lists: the Alexa global top 500 websites (ATL) [25], the websites identified as globally sensitive by the Citizen Lab [67] and the Berkman Klein Center [40][4] (CLBL-G), and, for each country, the websites identified as locally sensitive in that country by Citizen Lab and Berkman Klein (CLBL-C). We only use the global top 500 sites from Alexa’s ranking, because its “long tail” is unstable [162, 221]. All test lists are updated weekly.

[4] The lists maintained by Citizen Lab and Berkman Klein are formally independent but have substantial overlap, so we combine them.

ICLab has tested a total of 47,000 unique URLs over the course of its operation. Because all of the vantage points test ATL and CLBL-G, there is more aggregated data for these sites: 40% of our data is from sites on ATL, 40% from sites on CLBL-G, and 20% from sites on CLBL-C. Individual vantage points test anywhere from 3,000 to 5,700 URLs per measurement cycle, depending on the size of CLBL-C for the vantage point's country. This is by no means the complete set of sites blocked in any one country [79], and we have plans to broaden our testing, as described further in Section 2.3.8.

2.3.3.4 Data Collection

A *measurement* of a URL is an attempt to perform an HTTP GET request to that URL, recording information about the results from multiple layers of the network stack: (1) The complete DNS request and response or responses for the server hostname (using both a local resolver and a public DNS resolver). (2) Whether or not a TCP connection succeeded. (3) For HTTPS URLs, the certificate chain transmitted by the server. (4) The full HTTP response (both headers and body). (5) A traceroute to the server. (6) A comprehensive packet trace for the duration of the measurement. This allows us to identify anomalies that would not be apparent from application-layer information alone. For instance, when packets are injected by on-path censors, we can observe both the injected packets and the legitimate responses they conflict with (see Section 2.3.4.2).

Each vantage point measures connectivity to all of the sites on its test list at least once every three days, on a schedule controlled by the central server. Depending on the size of the test list, a cycle of measurements typically runs for 1–2 hours.

Figure 2.3 depicts ICLab's measurements over time in each country. Operating ICLab over a multi-year period has not been easy; several outages are visible in Figure 2.3. For instance, we lost access to our vantage points in Iran in May 2017 due

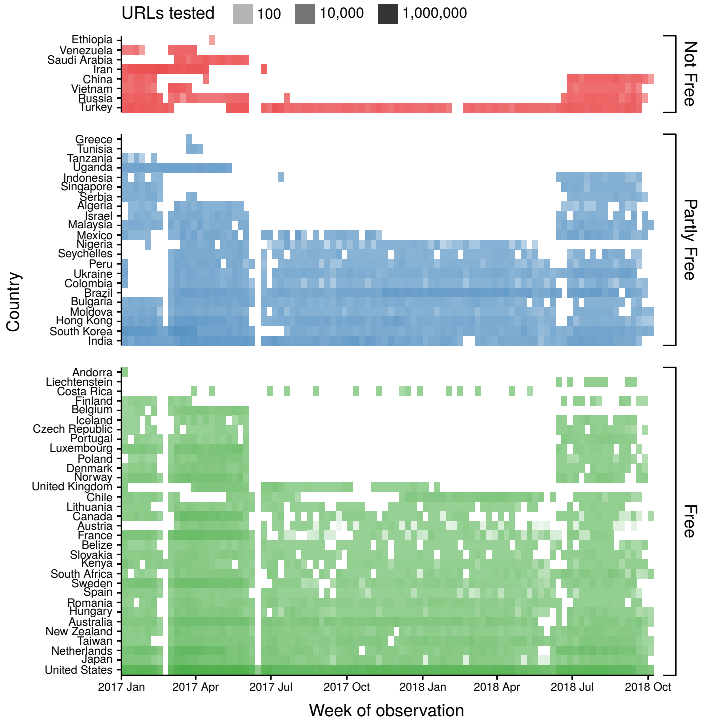


Figure 2.3. Measurements since 2017 by country. For each of the 62 countries where we have, or had, vantage points since 2017, the total number of measurements per week.

to a change in the international sanctions imposed on Iran, and we suffered a year-long, multi-country outage due to one VPN provider making configuration changes without notice. The latter incident led us to improve our internal monitoring and our tracking of VPN configuration changes.

Between January 2017 and September 2018, ICLab conducted 53,906,532 measurements of 45,565 URLs in 62 countries and 234 ASes. We publish our data for use by other researchers,[5] with periodic updates as we continue operation.

2.3.3.5 Control Nodes

Many tests of censorship rely on comparison of measurements between the vantage point and a "control" location, where there is not anticipated to be censorship.

[5] Available online at `https://iclab.org/`.

We repeat all the measurements performed by our vantage points on a *control node* located in an academic network in the USA. This network allows access to all the sites we test for accessibility. The control node has also suffered outages. In this paper, we use public data sets compiled by other researchers to fill in the gaps, as described in Section 2.3.4. We have since deployed three more control nodes in Europe, Asia and the USA to improve reliability and geographic diversity.

2.3.4 Censorship Detection

Next, we describe how ICLab detects manipulated DNS responses (§2.3.4.1), packets injected into TCP streams (§2.3.4.2) and HTML-based block pages (§2.3.4.3). All of ICLab's detection algorithms are designed to minimize both false negatives, in which a censored site is not detected, and false positives, in which ordinary site or network outages, or DNS load balancing are misidentified as censorship [97, 116, 144, 247].

2.3.4.1 DNS Manipulation

To access a website, the browser first resolves its IP address with a DNS query. To detect DNS manipulation, ICLab records the DNS responses for each measurement, and compares them with responses to matching DNS queries from our control node, and with DNS responses observed by control nodes OONI [103] operates. ICLab applies the following heuristics, in order, to the observations from the vantage point and the control nodes.

Vantage point receives two responses with different ASes. If a vantage point receives two responses to a DNS query, both with globally routable addresses, but belonging to two different ASes, we label the measurement as DNS manipulation. This heuristic detects on-path censors who inject a packet carrying false addresses [32]. Requiring the ASes to differ avoids false positives caused by a DNS load balancer picking a different address from its pool upon retransmission.

Vantage point receives NXDOMAIN or non-routable address. If a vantage point receives either a “no such host” response to a query (NXDOMAIN, in DNS protocol terms [43]), or an address that is not globally routable (e.g., `10.x.y.z`) [42], but the control nodes consistently receive a globally routable address (not necessarily the same one) for the domain name, over a period of seven days centered on the day of the vantage point’s observation, we label the test as DNS manipulation. The requirement for consistency over seven days is to avoid false positives on sites that have been shut down, during the period where a stale address may still exist in DNS caches.

Vantage point receives addresses from the same AS as control nodes. If a vantage point receives a globally routable address, and the control nodes also receive globally routable addresses assigned to the same AS (not necessarily the exact same address), we label the measurement as *not* DNS manipulation. Variation within a single AS is likely to be due to load-balancing over a server pool in a single location.

Vantage point and control nodes receive addresses in different ASes. The most difficult case to classify is when the vantage point and the control nodes receive globally routable addresses assigned to different ASes. This can happen when DNS manipulation is used to redirect traffic to a specific server (e.g., to display a block page). However, it can also happen when a content provider or CDN directs traffic to data centers near the client [196].

We distinguish censors from CDNs using the observation that censors tend to map many blocked websites onto just a few addresses [31, 116]. If a set of websites resolve to a single IP address from the vantage point, but resolve to IPs in more than θ ASes from the control nodes, we count those websites as experiencing DNS manipulation. θ is a tunable parameter which we choose by cross-checking whether these measurements also observed either a block page or no HTTP response at all. Taking this cross-check as ground truth, Figure 2.4 shows how the false positive

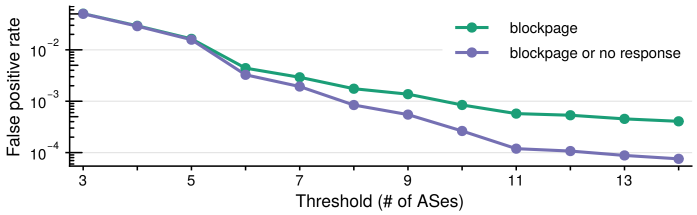


Figure 2.4. DNS manipulation false positives. The false positive rate for the DNS manipulation detector, as a function of the threshold parameter θ.

rate for DNS manipulation varies with θ. For the results in Section 2.3.5, we use a conservative $\theta = 11$ which gives a false positive rate on the order of 10^{-4}.

2.3.4.2 TCP Packet Injection

Censors may also allow DNS lookup to complete normally, but then inject packets that disrupt the TCP handshake or subsequent traffic. ICLab detects this form of censorship by recording packet traces of all TCP connections during each test, and analyzing them for (1) evidence of packet injection, and (2) evidence of intent to censor (e.g., block page content or TCP reset flags in injected packets). By requiring both types of evidence, we minimize false positives. Short error messages delivered by the legitimate server will not appear to be injected, and packets that, for innocuous reasons, appear to be injected, will not display an intent to censor.

Evidence of packet injection. If an end host receives two TCP packets with valid checksums and the same sequence number but different payloads, the operating system will generally accept the first packet to arrive, and discard the second [199]. An on-path censor can therefore suppress the server's HTTP response by injecting a packet carrying its own HTTP response (or simply an RST or FIN), timed to arrive first. Because ICLab records packet traces, it records both packets and detects a conflict. This is not infallible proof of packet injection; it can also occur for innocuous reasons, such as HTTP load balancers that do not send exactly the same packet when they retransmit.

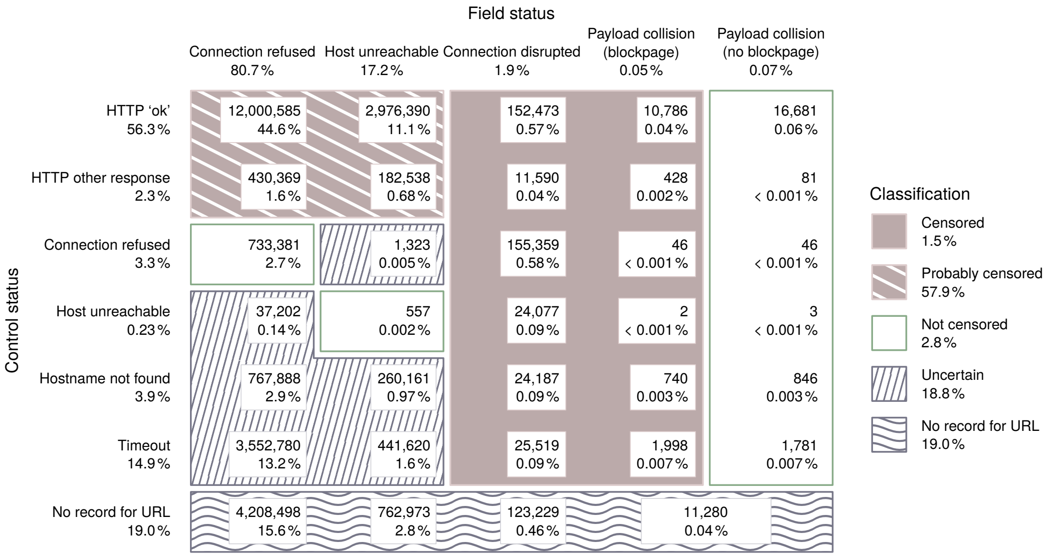


Figure 2.5. Classification of packet anomalies by comparison to control observations.

Intent to censor: RST, FIN, or block page. When we detect a pair of conflicting packets, we inspect them for evidence of intent to censor. An injected packet can disrupt/censor communication by carrying a TCP reset (RST) or close (FIN) flag, causing the client to abort the connection and report a generic error [72, 247]; or it can carry an HTTP response declaring the site to be censored (a "block page," discussed further in Section 2.3.4.3), which will be rendered instead of the true contents of the page the client requested [77, 144].

As with DNS manipulation, we compare each observation from a vantage point that shows evidence of packet injection, with matching observations from a control node. We apply the following heuristics, in order, to pairs of observations. The various outcomes of these heuristics are shown in Figure 2.5.

No matching control observation. When a TCP stream from the vantage point shows evidence of packet injection, but does not seem to correspond to any observation taken by the control node, we abandon any attempt to classify it. This is the "No record for URL" row of Figure 2.5.

This filtering is necessary because of a limitation in our packet trace analyzer. When a website transfers all of its traffic to another domain name, either via a CNAME record in DNS or using HTTP redirects, the trace analyzer cannot tell that TCP connections to the second domain name are associated with an attempt to test the first domain name. We conservatively do not consider these cases as censorship.

Packet collision after handshake, with RST or FIN. When a TCP stream from the vantage point shows evidence of collisions in TCP sequence numbers after successful completion of the three-way handshake, one side of the collision has its RST or FIN bit set and the other side has neither bit set, we label the measurement as censored by packet injection, regardless of what the control node observed. This is the “connection disrupted” column of Figure 2.5. We have high confidence that all of these are true positives.

Packet collision after handshake, with payload conflict. When a TCP stream from the vantage point shows evidence of TCP sequence number collisions after successful completion of the three-way handshake, but neither side of the collision has the RST or FIN bit set, we inspect the contents of the packets for a block page signature, as described in §2.3.4.3. We label the measurement as censored by packet injection only if a known block page signature appears in one of the packets. These cases are the “payload collision (blockpage)” and “payload collision (no blockpage)” columns of Figure 2.5. Again, we have high confidence that these are true positives and negatives.

Matching RST or ICMP unreachable instead of SYN-ACK. When a TCP SYN from the vantage point receives either a TCP RST or an ICMP unreachable packet, instead of a SYN-ACK, and the control node observes the same network error, we conclude the site is down for everyone, and label the measurement as *not* censored. These cases are the matching “connection refused” and “host unreachable” cells on the left-hand side of Figure 2.5, and we have high confidence that they are true negatives.

RST or ICMP unreachable instead of SYN-ACK, at vantage only. When a TCP SYN from the vantage point receives either a TCP RST or an ICMP unreachable packet in response, instead of a SYN-ACK, but the control node is able to carry out a successful HTTP dialogue, this *probably* indicates IP-based censorship observed by the vantage point. However, there are other possible explanations, such as a local network outage at the vantage point, or a site blocking access from specific IP addresses on suspicion of malice [175]. Manual spot-checking suggests that many, but not all, of these observations are censorship. These cases are labeled as "probable censorship" in Figure 2.5, and we discuss them separately in Section 2.3.5.

Mismatched network errors, or timeout or DNS error at control node. When the vantage point and the control node both received a network error in response to their initial SYN, but not the same network error; when the control node's initial SYN received no response at all; and when the control node was unable to send a SYN in the first place because of a DNS error; we cannot say whether the measurement indicates censorship. These cases are the cells labeled "uncertain" in the lower left-hand corner of Figure 2.5. We are conservative and do not consider these as censorship in our analysis.

2.3.4.3 Block Page Detection and Discovery

Block page contents vary depending on the country and the technology used for censorship. Known block pages can be detected with regular expressions applied to the TCP payloads of suspicious packets, but these will miss small variations from the expected text, and are no help at all with unknown block pages.

Nonetheless, ICLab uses a set of 308 regular expressions to detect known block pages. We manually verified these match specific, known block pages and nothing else. 40 of them were developed by the Citizen Lab [68], 24 by OONI [103], 144 by Quack [240], and 100 by us.

Anomalous packets that do *not* match any of these regular expressions are examined for block page variations and unknown block pages, as described below; when we discover a block page that was missed by the regular expressions, we write new ones to cover them.

Self-contained HTTP response. To deliver a block page, the protocol structure of HTTP requires a censor to inject a single packet containing a complete, self-contained HTTP response. This packet must conflict with the first data packet of the legitimate response. Therefore, only packets which are both involved in a TCP sequence number conflict, and contain a complete HTTP response, are taken as candidate block pages for the clustering processes described next.

HTML structure clusters. The HTML tag structure of a block page is characteristic of the filtering hardware and software used by the censor. When the same equipment is used in many different locations, the tag structure is often an exact match, even when the text varies. We reduce each candidate block page to a vector of HTML tag frequencies (1 `<body>`, 2 `<p>`, 3 `<em>`, *etc.*) and compare the vectors to all other candidate block pages' vectors, and to vectors for pages that match the known block page regular expressions. When we find an exact match, we manually inspect the matching candidates and decide whether to add a new regular expression to the detection set. Using this technique we discovered 15 new block page signatures in five countries.

Textual similarity clusters. Within one country, the legal jargon used to justify censoring may vary, but is likely to be similar overall. For example, one Indian ISP refers to "a court of competent jurisdiction" in its block pages, and another uses the phrase "Hon'ble Court" instead. Small variations like this are evidently the same page to a human, but a regular expression will miss them. We apply *locality-sensitive* *hashing* (LSH) [263] to the text of the candidate block pages, after canonicalizing the HTML structure. LSH produces clusters of candidate pages, centered on pages that do match the known block page regular expressions. As with the tag frequency vectors, we manually inspect the clusters and decide whether to add new regular expressions to the curated set. Using this technique, we discovered 33 new block page signatures in eight countries. Figure 2.6 shows an example group of block pages detected by textual similarity clustering, including two variations on the Indian legal jargon mentioned in Section 2.3.4.3, but also messages mimicking generic HTTP server errors. This demonstrates how similarity clustering can detect covert as well as overt censorship.

HTML structure

```
ACK+PSH
HTTP/1.1 200 OK
Connection: close
Content-Length: nnnn
Content-Type: text/html; charset="utf-8"
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML
4.01//EN">
<html>
<head<title></title></head>
<body><h0><font color="black">
visible message
</font></h0></body></html>
```

Visible message

```
\This URL has been blocked under instructions
of a
competent Government Authority or in
compliance with
the orders of a Court of competent
jurisdiction.

***This URL has been blocked under
Instructions of the
Competent Government Authority or Incompliance
to
the orders of Hon’ble Court.*** [sic]

*\Error 403:  Access Denied/Forbidden"*

404.  That’s an error.

HTTP Error 404 - File or Directory not found

HTTP Error 404 - File or Directory not found =
http://...
```

Figure 2.6. Example cluster of block pages. All of the messages in the right-hand column were observed with the HTTP response headers and HTML structure shown on the left.

URL-to-country ratio. To discover wholly unknown block pages we take each LSH cluster that is *not* centered on a known block page, count the number of URLs that produced a page in that cluster, and divide by the number of countries where a page in that cluster was observed. This is essentially the same logic as counting the number of websites that resolve to a single IP from a test vantage point but not a control vantage point, but we do not use a threshold. Instead, we sort the clusters from largest to smallest URL-to-country ratio and then inspect the entire list manually. The largest ratio associated with a newly discovered block page was 286 and the smallest ratio was 1.0.

2.3.5 Findings

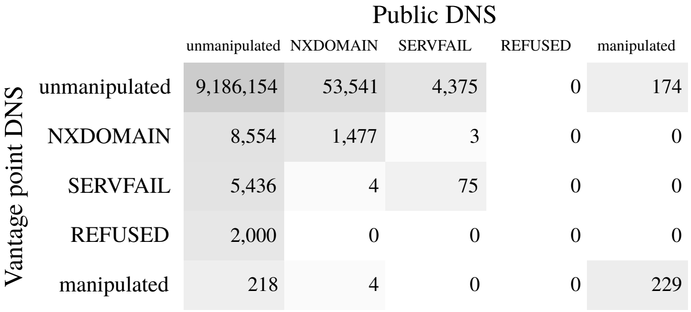
Between January 2017 and September 2018, ICLab conducted 53,906,532 measurements of 45,565 URLs in 62 countries. Because we do not have continuous coverage of all these countries (see §2.3.3.4), in this paper we present findings only for countries where we successfully collected at least three months' worth of data prior to September 2018. Among those countries, five stand out as conducting the most censorship overall: Iran, South Korea, Saudi Arabia, India, and Kenya. When considering specific subsets of our data, sometimes Turkey or Russia displaces one of these five.

2.3.5.1 Specific Results

We first present details of our observations for each of the three censorship techniques that we can detect.

DNS manipulation. We observe 15,007 DNS manipulations in 56 countries, applied to 489 unique URLs. 98% of these cases received NXDOMAIN or non-routable addresses.

Figure 2.7 compares DNS responses from a vantage point's local recursive resolver with those received by the same vantage point from a public DNS utility (e.g., `8.8.8.8`). The upper left-hand cell of this chart counts cases where there is no DNS censorship; the other cells in the left-hand column count cases where censorship is being performed by the local DNS recursive resolver. The top rightmost cell counts the number of observations where censorship is being performed only by a public DNS utility, and the bottom rightmost cell counts cases where censorship is being performed by both a local recursive resolver and a public DNS utility. We observe censorship by public DNS utilities only for a few sites from Russia, Bulgaria, and Iran.



Public DNS

Vantage point DNS	unmanipulated	NXDOMAIN	SERVFAIL	REFUSED	manipulated
unmanipulated	9,186,154	53,541	4,375	0	174
NXDOMAIN	8,554	1,477	3	0	0
SERVFAIL	5,436	4	75	0	0
REFUSED	2,000	0	0	0	0
manipulated	218	4	0	0	229

Figure 2.7. Comparison of DNS responses for the same domain between local and public nameservers from the same vantage point.

The middle three columns could be explained as either censorship or as unrelated DNS failures.

Packet injection. We observe 19,493,925 TCP packet injections across 55 countries, applied to 11,482 unique URLs. However, after applying the filtering heuristics described in §2.3.4.2, only 0.7% of these are definitely due to censorship: 143,225 injections, in 54 countries, applied to 1,205 unique URLs. (The numbers in Figure 2.5 are higher because they do not account for all the filtering heuristics.) Packet injections are usually used to disrupt a connection without delivering a block page; block pages are delivered by only 3.4% of the injections we attribute to censorship.

Another 15,589,882 packet injections—58% of the total—are network errors received instead of a SYN-ACK packet. These are described as "probable censorship" in Figure 2.5. They could indicate an in-path censor blocking hosts by IP address, but there are many other possible explanations. Our synthetic results (below) might be quite different if we were able to classify these more accurately.

Block pages. We observe 232,183 block pages across 50 countries, applied to 2,782 unique URLs. Iran presents block pages for 24.9% of the URLs it censors, more than any other country. In all of the countries we monitor, block pages are most likely to be used for URLs in the pornography and news categories (see below).

Table 2.3. Censorship by Test List and Category. For each of the three test lists we use (see §2.3.3.3), the top ten countries censoring the most URLs from that list, the top three FortiGuard categories for their censored URLs (abbreviations defined in Table 2.4), and the percentage of URLs from that list that are censored.

Overall			**Alexa Global (ATL)**			**Globally Sensitive (CLBL-G)**			**Per-Country Sensitive (CLBL-C)**		
Country	**Category**	**Pct.**	**Country**	**Category**	**Pct.**	**Country**	**Category**	**Pct.**	**Country**	**Category**	**Pct.**
Iran	NEWS	13.1%	Iran	NEWS	14.0%	Iran	PORN	11.6%	Iran	NEWS	21.0%
	PORN	9.2%		PORN	12.7%		NEWS	9.4%		BLOG	17.6%
	BLOG	7.5%		ENT	10.3%		PROX	6.8%		POL	7.2%
South Korea	PORN	15.4%	South Korea	SHOP	14.2%	Saudi Arabia	PORN	31.0%	India	ENT	19.0%
	NEWS	8.4%		PORN	13.7%		GAMB	13.5%		STRM	14.3%
	ORG	7.4%		NEWS	10.8%		PROX	12.2%		NEWS	10.8%
Saudi Arabia	PORN	29.5%	Saudi Arabia	PORN	70.0%	South Korea	PORN	15.6%	Saudi Arabia	NEWS	54.0%
	NEWS	11.3%		ILL	6.6%		ORG	10.4%		POL	7.7%
	GAMB	10.1%		GAMB	6.6%		NEWS	5.7%		RELI	7.7%
India	ENT	13.3%	Turkey	PORN	66.0%	Kenya	PORN	14.5%	Russia	BLOG	16.5%
	STRM	10.8%		ILL	4.0%		GAMB	10.8%		NEWS	14.4%
	NEWS	10.4%		FILE	4.0%		PROX	9.0%		GAMB	12.4%
Kenya	PORN	15.5%	India	ILL	35.5%	Turkey	PORN	47.0%	Turkey	NEWS	29.4%
	GAMB	10.1%		IT	8.8%		GAMB	22.6%		PORN	13.7%
	PROX	8.3%		STRM	6.6%		ILL	3.2%		GAMB	9.8%
Turkey	PORN	40.2%	Kenya	ILL	28.1%	India	NEWS	10.3%	South Korea	PORN	16.7%
	GAMB	16.6%		PORN	25.0%		ILL	9.2%		NEWS	16.1%
	NEWS	9.2%		GAME	6.2%		IT	8.0%		SHOP	11.8%
Russia	GAMB	23.4%	Russia	PORN	26.3%	United States	NEWS	8.0%	China	NEWS	46.1%
	PORN	10.0%		SHOP	21.0%		IT	6.9%		ORG	46.1%
	NEWS	7.6%		STRM	10.5%		SEAR	6.3%		RELI	7.7%
Uganda	PORN	42.6%	Japan	SEAR	19.0%	Uganda	PORN	42.6%	Hong Kong	ORG	100.0%
	ADUL	11.7%		NEWS	9.5%		ADUL	11.7%			
	LING	10.3%		GAME	9.5%		LING	10.3%			
Netherlands	NEWS	13.4%	Netherlands	ILL	15.3%	Russia	GAMB	39.4%	Poland	GAMB	100.0%
	ILL	8.5%		NEWS	15.3%		PORN	14.9%			
	SEAR	8.5%		SEAR	15.3%		RELI	5.3%			
Japan	NEWS	11.0%	Sweden	SEAR	27.2%	Netherlands	NEWS	13.0%	Singapore	PROX	66.6%
	GAME	9.6%		BLOG	9.1%		ILL	7.2%		GAME	33.3%
	SEAR	9.6%		STRM	9.1%		GAME	7.2%			

2.3.5.2 Synthetic Analysis

Combining observations of all three types of censorship gives us a clearer picture of what is censored in the countries we monitor, and complements missing events in each.

We use the "FortiGuard" URL classification service, operated by FortiNet [6], to categorize the contents of each test list. This service is sold as part of a "web filter" for corporations, which is the same software as a nation-state censorship system, but on a smaller scale. The URLs on all our lists, together, fall into 79 high-level categories according to this service; the 25 most common of these, for URLs that are censored at least once, are listed in Table 2.4, along with the abbreviated names used in other tables in this section.

Table 2.4. FortiGuard Categories and Abbreviations. The 25 most common categories for the URLs on our test lists that were censored at least once, with the abbreviated names used in Tables 2.3 and 2.5, and the percentage of URLs in each category. CLBL includes both global and per-country test lists.

Abbrev.	**Category**	**ATL %**	**CLBL %**
ADUL	Other Adult Materials	0.91	0.77
ARM	Armed Forces	0.76	0.31
BLOG	Personal websites and blogs	2.00	8.97
DOM	Domain Parking	0.21	0.28
ENT	Entertainment	2.66	2.25
FILE	File Sharing and Storage	1.89	0.55
GAME	Games	2.62	0.83
GAMB	Gambling	1.73	1.18
HEAL	Health and Wellness	2.02	1.04
ILL	Illegal or Unethical	1.85	0.40
IM	Instant Messaging	0.49	0.14
IT	Information Technology	9.31	4.17
ITRA	Internet radio and TV	0.39	0.59
LING	Lingerie and Swimsuit	0.76	0.14
NEWS	News and Media	10.03	18.87
ORG	General Organizations	6.82	4.77
POL	Political Organizations	1.56	5.28
PORN	Pornography	3.87	2.45
PROX	Proxy Avoidance	1.71	0.57
RELI	Global Religion	3.19	2.58
SEAR	Search Engines and Portals	3.93	2.36
SHOP	Shopping	4.86	1.40
SOC	Social Networking	1.19	1.34
SOLI	Society and Lifestyles	0.76	0.97
STRM	Streaming Media and Download	1.83	1.42

Table 2.3 shows the three most censored categories of URLs for the five countries conducting the most censorship, based on the percentage of unique URLs censored over time. It is divided into four columns, showing how the results vary depending on which of our test lists are considered: all of them, only ATL, only CLBL-G, or only CLBL-C.

Iran takes first place in all four columns, and Saudi Arabia is always within the top three. The other countries appearing in Table 2.3 are within the top five only for some test lists. The top three categories blocked by each country change somewhat from list to list. For instance, pornography is much less prominent on the country-specific lists than on the global list. Iran's censorship is more uniformly distributed over topics

Table 2.5. Censorship Variation by Technique. For each of the three techniques we can detect, the five countries observed to censor the most URLs using that technique, and the top three FortiGuard categories for those URLs (abbreviations defined in Table 2.4). Percentages are of all unique URLs tested.

Technique	**Country**	**Categories**	**Pct.**
Block page	Iran	NEWS, PORN, BLOG	24.95%
	Saudi Arabia	PORN, NEWS, GAMB	11.1%
	India	ENT, STRM, NEWS	6.4%
	Kenya	PORN, GAMB, PROX	4.8%
	Turkey	PORN, GAMB, NEWS	4.6%
DNS manipulation	Iran	BLOG, PORN, PROX	5.5%
	Uganda	PORN, ADUL, LING	1.7%
	Turkey	ILL, GAMB, STRM	0.3%
	Bulgaria	ILL, ARM, DOM	0.2%
	Netherlands	ILL, IM, DOM	0.2%
TCP packet injection	South Korea	PORN, ORG, NEWS	9.3%
	India	NEWS, ILL, IT	2.3%
	Netherlands	NEWS, SEAR, GAME	0.9%
	Japan	NEWS, GAME, SEAR	0.9%
	Australia	SEAR, NEWS, ILL	0.8%

than the other countries, where censorship is concentrated on one or two categories. These results demonstrate how the choice of test lists can change observations about censorship policy.

Table 2.5 shows the top five countries conducting the most censorship, for each of the three censorship techniques that ICLab can detect, with the top three categories censored with that technique. This shows how censors use different techniques to censor different types of content, as we mentioned in §2.3.1. For example, Turkey uses DNS manipulation for categories ILL and STRM, but uses block pages for PORN and NEWS.

Figure 2.8 shows how often the various blocking techniques are combined. For instance, in Iran we detect some URLs being redirected to a block page via DNS manipulation (comparing with Table 2.5, we see that these are the URLs in the PORN and BLOG categories), but for many others, we detect only the block page.

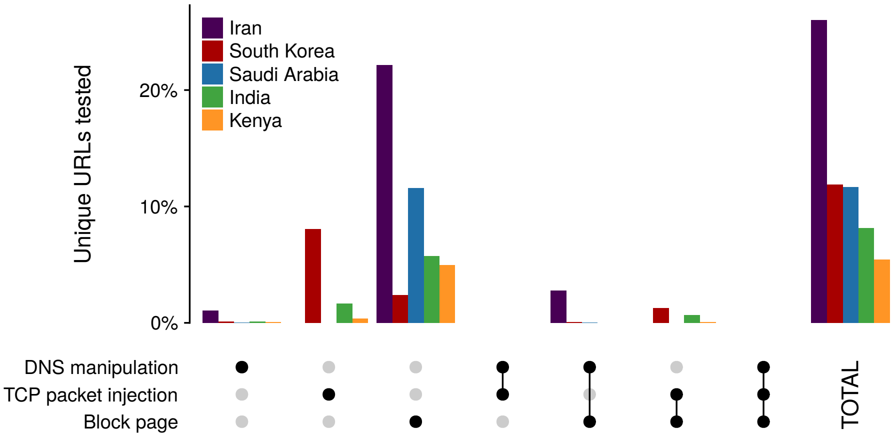


Figure 2.8. Combinations of Censorship Techniques. For the five countries performing the most censorship overall, which combinations of the three phenomena ICLab can detect are observed. Except for "TOTAL," each group of bars is mutually exclusive—URLs counted under "DNS manipulation and packet injection" are not also counted under either "DNS manipulation" or "packet injection."

This could be because Iran uses a technique we cannot detect for those URLs (e.g., route manipulation), or because our analysis of packet injection is too conservative (see Section 2.3.4.2).

2.3.5.3 Longitudinal Analysis

Collecting data for nearly two years gives us the ability to observe changes in censorship over time. Figure 2.9 shows censorship trends for the six countries ICLab can monitor that block the most URLs from the global test lists (ATL and CLBL-G), plus a global trend line computed from aggregate measurements from all the other monitored countries. We do not have complete coverage for Iran and Saudi Arabia, due to the outages mentioned in §2.3.3.4. The large dip in several of the trend lines in February 2017 is an artifact due to month-to-month churn within the Alexa rankings (see Scheitle et al. [221]).

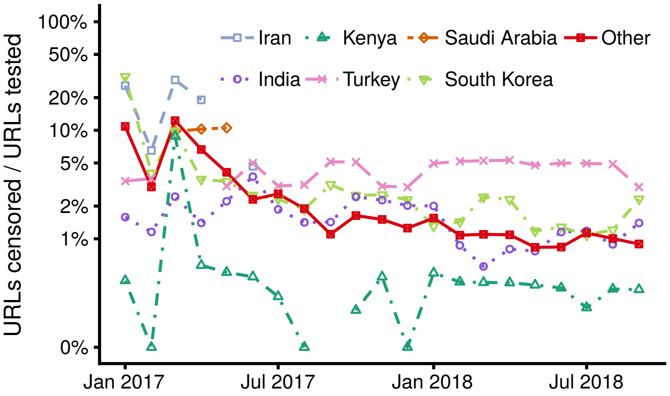


Figure 2.9. Logarithmic Plot of Longitudinal Trends. Changes over time in the level of censorship, within the six countries where we observe the most censorship of URLs from ATL and CLBL-G, plus the aggregate of all other monitored countries.

The global trend line shows a steady decreasing trend, which we attribute to the rising use of secure channel protocols (e.g., TLS) and circumvention tools. This trend is also visible for South Korea but not for the other top five countries.

Iran blocks 20–30% of the URLs from ATL, more than any other country. This is due to extensive blocking of URLs in the NEWS and BLOG categories. Saudi Arabia consistently blocks roughly 10% of ATL and CLBL-G URLs, mostly from the PORN and GAMB categories with some NEWS as well. South Korea applies a similar level of blocking for the PORN and GAMB categories; it is a much more democratic nation than Saudia Arabia, but it nonetheless has applied draconian restrictions to “indecent Internet sites” (including both pornography and gambling sites) since before 2008 [189].

Censorship in Kenya is stable at a rate of roughly 0.4% except for March 2017, where the rate spikes to 10%. This is an artifact; for that one month, our VOD in Kenya was connected to a network that applied much more aggressive “filtering” to porn, gambling, and proxy sites than is typical for Kenya, using a commercial product.

At the beginning of 2018, we observe a drop in the level of filtering in India, from 2% to 0.8%, followed by a slow rise back to about 1.5% after about four months. This coincides with political events: India's telecommunications regulator announced support for "net neutrality" at the end of 2017 [218, 259], and most ISPs suspended their filtering in response. However, when a detailed regulation on net neutrality was published in mid-2018 [234], it became clear that the government had not intended to relax its policy regarding content deemed to be illegal, and the filtering was partially reinstated.

Similarly, we see a rise in the level of filtering in Turkey in June 2017, from an earlier level of 3% to 5%. Although it is not visible on this chart, the topics censored also change at this time. Prior to the rise, most of the blocked sites in Turkey carried pornography and other sexual content; after the rise, many more news sites were blocked. This, too, coincides with political events. Following a controversial referendum which increased the power of the Turkish Presidency, the government has attempted to suppress both internal political opposition and news published from other countries. International news organizations took notice of the increased level of Turkish online censorship in May of 2017 [106, 228], while ICLab detected it around the end of April.

2.3.5.4 Heuristic False Positive

We manually reviewed the results of all of our heuristic detectors for errors. Manual review cannot detect false negatives, because we have no way of knowing that we *should* have detected a site as censored, but false positives are usually obvious. Here we discuss the most significant cases we found, and how we adjusted the heuristics to compensate.

DNS Manipulation. We manually verified the detection results identified by each heuristic. The only heuristic producing false positives was the rule for when a vantage point and control nodes receive addresses in different ASes. As we mentioned in Section 2.3.4.1, this heuristic gives a false positive rate on the order of 10^{-4} with the value of θ we selected.

Packet injection. As with DNS manipulation, we manually reviewed the results of each heuristic for false positives. We found many false positives for RST or ICMP unreachable instead of SYN-ACK, leading us to reclassify these as only "probable" censorship and not include them in the synthetic analysis above. We also found cases in all of the categories where a packet anomaly was only observed once, for a URL that seemed unlikely to be censored from that vantage (e.g., connection disrupted to an airline website from a VPN vantage in the USA). We therefore discount all cases where a packet anomaly has only been observed once for that URL in that country.

Block pages. Our set of regular expressions did initially produce some false positives, for instance on news reportage of censorship, quoting the text of a block page. We manually reviewed all of the matches and adjusted the regular expressions until no false positives remained. It was always possible to do this without losing any true positives.

2.3.6 Other Cases of Network Interference

In this section, we describe three cases of network interference discovered with ICLab, that are different from the form of censorship we set out to detect: Geoblocking by content providers (§2.3.6.1), injection of a script to fingerprint clients (§2.3.6.2), and injected malware (§2.3.6.3).

2.3.6.1 Geoblocking and HTTP 451

HTTP status code 451, "Unavailable for Legal Reasons," was defined in 2016 for web servers to use when they cannot provide content due to a legal obstacle (e.g., the Google restricts access to clients from Iran to enforce US sanctions [175]) or requests from foreign governments [45].

We observe 23 unique websites that return status 451, from vantages in 21 countries. Six of these cases appear to be `wordpress.com` complying with requests from Turkey and Russia (for blogs related to political and religious advocacy). Along with the HTTP 451 status, they also serve a block page, explaining that `wordpress.com` is complying with local laws and court orders. Two more websites (both pornographic) were observed to return status 451 from Russia, with HTTP server headers indicating the error originates from the Cloudflare CDN, but without any explanation. Since the adoption of the GDPR [112] we have observed a few sites returning status 451 when visited from European countries.

Since status 451 is relatively new, the older, more generic status 403 ("Forbidden") is also used to indicate geoblocking for legal reasons. Applying the tag frequency clustering technique described in Section 2.3.4.3 to the accompanying HTML, we were able to discover six more URLs, in four countries, where status 403 is used with a block page stating that access is prohibited from the client's location. Three of these were gambling sites, with the text of the block page stating that the sites are complying with local regulations.

We also observe a related phenomenon at the DNS level. From a single VPN server located in the USA, we observed `netflix.com` resolving to an IANA-reserved IP address, `198.18.0.3`. This could be Netflix refusing to provide their service to users behind a VPN.

2.3.6.2 User Tracking Injection

Our detector for block pages (§2.3.4.3) flagged a cluster of TCP payloads observed only in South Korea. Upon manual inspection, these pages contained a script that would fingerprint the client and then load the originally intended page. We observed injections of this script over a five-month period from Oct. 2016 through Feb. 2017, from vantage points within three major Korean ISPs, into 5–30% of all our test page loads, with no correlation with the content of the affected page. By contrast, censorship in South Korea affects less than 1% of our tests and is focused on pornography, illegal file sharing, and North Korean propaganda.

These scripts could be injected by the VPN service, the ISPs, or one or more of their transit providers. The phenomenon resembles techniques used by ad networks for recording profiles of individual web users [21]. This demonstrates the importance of manual checking for false positives in censorship detection. All of the detection heuristics described in Sections 2.3.4.2 and 2.3.4.3 triggered on these scripts, but they are not censorship.

2.3.6.3 Cryptocurrency Mining Injection

Our block page detector also flagged a set of suspicious responses observed only in Brazil. The originally intended page would load, but it would contain malware causing the web browser to mine cryptocurrency. (As of mid-2018, this is a popular way to earn money with malware [157].) We were able to identify the malware as originating with a botnet infecting MikroTik routers (exploiting CVE-2018-14847), initially seen only in Brazil [149] but now reported to affect more than 200,000 routers worldwide [94]. Infected routers inject the mining malware into HTTP responses passing through them.

The malware appears in ICLab's records as early as July 21st, 2018—ten days before the earliest public report on the MikroTik botnet that we know of. If ICLab's continuous monitoring were coupled with continuous analysis and alerting (which is planned) it could also have detected this botnet prior to the public report. This highlights the importance of continuously monitoring network interference in general.

2.3.7 Comparison with other Platforms

Other censorship measurement platforms active, at the time of writing, include Encore [50], Satellite-Iris [196, 222], Quack [240], and OONI [103]. Table 2.6 shows

Table 2.6. High-level comparison of ICLab with five other censorship monitoring platforms.

Platform	Packet	Vantage Point Types			Detection Capabilities			Coverage (avg/max)		
	Capture	VPNs	ORs[a]	VODs	DNS	TCP	Blockpage	Countries	ASes	URLs
Encore [50]			✓					Unknown[b]	Unknown	23
Satellite-Iris [196, 222]			✓		✓			174 / 179	3,261 / 3,617	2,094 / 2,423
Quack [240]			✓					75 / 76	3,528 / 4,135	2,157 / 2,484
OONI [103]	○[c]			✓	✓		✓	113 / 156	670 / 2,015	13,582 / 20,258
ICLab	✓	✓		✓	✓	✓	✓	42 / 50	48 / 62	16.964 / 23,992

[a] Open Relays: Internet hosts that will relay a censorship probe from researchers' computers without any prior arrangement.

[b] Due to privacy concerns, Encore does not record this information.

[c] The OONI client can optionally collect packet traces. However, OONI's servers do not record traces, due to privacy concerns.

the high-level features provided by each of these platforms, and a comparison of their country, AS, and URL coverage for the two-month period of August and September 2018. (August 1, 2018 is the earliest date for which data from Quack and Satellite-Iris has been published.) All platforms suffer some variation from day to day in coverage, so we report both a weekly average and the maximum number of covered countries, ASes, and URLs. While many of the platforms described in Table 2.6 have chosen to emphasize breadth of country and AS coverage at the expense of detail. ICLab takes the opposite approach, collecting detailed information from a smaller number of vantage points.

2.3.7.1 Quack

Quack relies on public echo servers to measure censorship. It requires at least 15 echo servers within the same country for robust measurements [240]. Currently, these are available to Quack from 75 countries. 95 more countries have at least one echo server, which can still provide some measurements.

Quack aims to detect censorship of websites, but it does not send or receive well-formed HTTP messages. Instead it sends packets that mimic HTTP requests, which the echo server will reflect back to the client. It expects the censor to react to this reflection in the same way that it would to a real HTTP message. The designers of Quack acknowledge the possibility of false negatives when the censor only looks for HTTP traffic on the usual ports (80 and 443). More seriously, manual inspection of the Quack data set reveals that in 32.6% of the tests marked as *blocked*, the client did not successfully transmit a mimic request in the first place. We have reported this apparent bug to the Quack team.

2.3.7.2 Satellite-Iris

Satellite-Iris [56] combines Satellite [222] and Iris [196]. It focuses on DNS manipulation, measuring from open DNS resolvers. It compares the responses received from these vantage points with responses observed from a control node. It also retrieves corresponding TLS certificates from the Censys [90] data set and checks whether they are valid. It applies several heuristics to each response, *all* of which must be satisfied for the response to be judged as censorship. We now highlight two cases where their heuristics lead to false negatives.

If Satellite-Iris can retrieve a TLS certificate from *any* of the IP addresses in the open resolver's response, and that certificate is valid for *any* domain name, it considers the response not to be censored. This means Satellite-Iris will not detect any case where the censor supplies the address of a server for a different domain in forged DNS responses. In Satellite-Iris's published data set, 0.01% of the measurements are affected by this. Specifically, measurements from China and Turkey show domains belonging to the BBC, Google, Tor and others resolving to an IP address belonging to Facebook's server pool.

Despite the design bias toward false negatives, we also find that 83.8% of the DNS responses considered censored by Satellite-Iris may be false positives. Satellite-Iris depends on the Censys data set to distinguish DNS poisoning from normal IP variation (e.g., due to geotargeting and load balancing by CDNs). When Censys

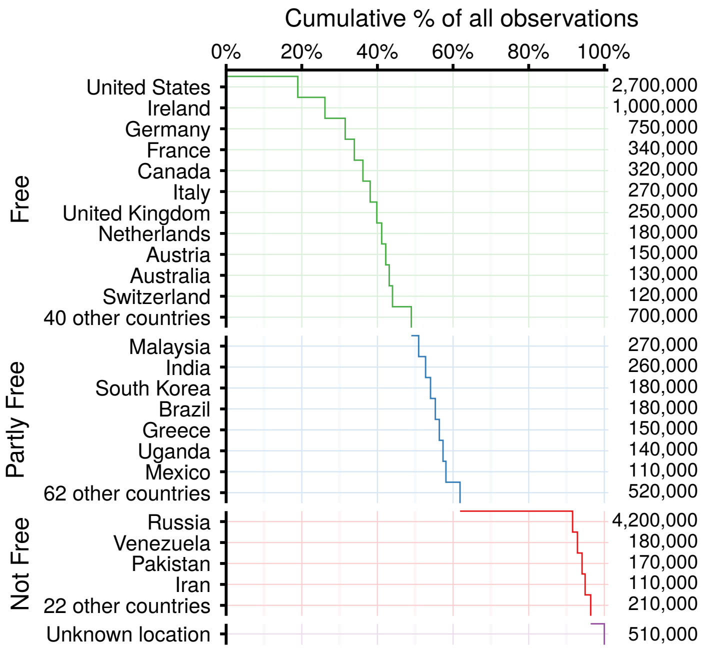


Figure 2.10. Distribution of OONI observations by country in August and September of 2018, grouped by Freedom House classification.

information is unavailable, it falls back on a comparison to a single control resolver, which is inadequate to rule out normal variation, as discussed in §2.3.4.1.

2.3.7.3 OONI

OONI [103] relies on volunteers who run a testing application manually. The application is available on all major desktop and mobile operating systems except Windows. In August and September of 2018, OONI's volunteers conducted 14,000,000 measurements from 156 countries, and reported 29,982 unique URL-country pairs as blocked.

OONI's reliance on volunteers, and on manual operation of the testing application, means that its coverage is not evenly distributed over websites or countries. Their "primary web connectivity" test suite also tests ATL and CLBL-G, but the mobile phone version of the application only tests a short subsample on each run, in order to limit the time and bandwidth consumed by the test. 62% of the measurements for the

August and September 2018 period tested fewer than 80 URLs. Figure 2.10 shows the distribution of countries covered by OONI. 86% of all observations originate from the 23 countries named in this figure, and 48% are from just two countries—Russia and the USA. For another 4% of the observations, no location could be identified. Comparing Figure 2.10 to Figure 2.3 shows that OONI does not achieve better coverage of "partly free" and "not free" countries than ICLab does. Volunteers whose Internet access is unreliable or misconfigured may submit inaccurate results. Indeed, more than 100,000 of the observations are tagged inconclusive due to local network errors (e.g., disconnection during the test).

Because OONI's testing application runs without any special privileges, it normally cannot record packet traces. OONI's detection heuristics are rudimentary, leading to a high level of false positives. OONI's DNS consistency test will flag *any disagreement* between the client's local recursive resolver and a public DNS utility as censorship [188]. OONI's block page detector relies on the "30% shorter than uncensored page" heuristic proposed by Jones et al. [144], but innocuous server errors are also short compared to normal page.

Yadav et al. [259] reported very high levels of inaccuracy in OONI's results for India, with an 80% false positive rate and a 11.6% false negative rate. We confirm a high false positive rate for OONI's block page detector: of the 12,506 unique anomalous HTTP responses reported by OONI's volunteers in August and September 2018, our block-page detector only classifies 3,201 of them as censorship, for a 74.4% false positive rate. The most common cause of false positives is a response with an empty HTML body, which can occur for a wide variety of innocuous reasons as well as censorship [23, 175, 236].

2.3.8 Limitations

In this section, we discuss ICLab's limitations and how we have addressed them.

Discrimination against VPN users. Some websites may block access from VPN users [175, 222]. We sometimes observe this discrimination against our VPN clients (see §2.3.6.1 for an example), and are careful not to confuse it for censorship.

Malicious VPN Providers. Some VPN providers engage in surveillance and traffic manipulation, for instance to monetize their service by injecting advertisements into users' traffic [151]. We avoid using VPN providers that are known to do this. Our block page detectors are designed not to confuse dynamic content (e.g., advertisements, localization) with censorship, as described in §2.3.4.3. In §2.3.6.2 and 2.3.6.3 we describe surveillance and malware injections that required manual inspection to distinguish from censorship.

VPN providers are also known to falsely advertise the location of their VPN servers [251]. We verify all server locations using the technique described in §2.3.3.2.

Bias in Test Lists. ATL suffers from sampling bias and churn [221]. CLBL-G and CLBL-C may suffer from selection bias, since they are manually curated by activists. We have plans to revise the test lists and add more URLs as needed.

CLBL-G and CLBL-C are updated slowly. It is not unusual for more than half of the sites on a country-specific list to no longer exist [250]. This is not as much of a limitation as it might seem, because censors also update their lists slowly. Several previous studies found that long-gone websites may still be blocked [20, 180, 259].

Coverage of "Not Free" Countries. As discussed in Section §2.3.3.2, the risks involved with setting up many vantage points in certain sensitive ("not free") countries prevent us from claiming we can obtain complete coverage at all times. However, the set of countries ICLab covers gives us a good, if imperfect, longitudinal overview of worldwide censorship.

Evading Censorship Detection. Censors are known to try to conceal some of their actions ("covert" censorship). ICLab can detect some covert censorship, as discussed in §2.3.4, but not all of it. The "uncertain" and "probably censored" cases of TCP packet injection (Figure 2.5 in §2.3.4.2) are priorities for further investigation. Censors could further conceal their actions by disabling filtering for IP addresses that appear to be testing for censorship. Comparing results for vantage points in the same country gives us no reason to believe any country does this today.

2.3.9 Conclusion

We presented ICLab, a global censorship measurement platform that is able to measure a wide range of network interference and Internet censorship techniques.

By using VPN-based vantage points, ICLab provides flexibility and control over measurements, while reducing risks in measuring Internet censorship at a global scale. Between January 2017 and September 2018, ICLab has conducted 53,906,532 measurements over 45,565 URLs in 62 countries.

ICLab is able to detect a variety of censorship mechanisms as well as other forms of network interference. Other longitudinal measurement platforms may have more vantage points and accumulated data than ICLab, but also more errors, and/or they only focus on a specific type of censorship. Our platform can more reliably distinguish normal network errors from covert censorship, and our clustering techniques discovered 48 previously unknown block pages.

As we continue to operate ICLab and interact with relevant political science and civil society organizations, ICLab will not only make new technical observations, but also place qualitative work in this area on a firm empirical footing.

2.4 Demystifying Great Firewall's DNS Censorship Behavior

The Great Firewall of China (GFW) has long used DNS packet injection to censor Internet access. In this chapter, we analyze the DNS injection behavior of the GFW over a period of nine months using the Alexa top 1M domains as a test list. We first focus on understanding the publicly routable IPs used by the GFW and observe groups of IPs used to filter specific sets of domains. We also see a sharp decline in public IPs injected by the GFW in November 2019. We then fingerprint three different injectors that we observe in our measurements. Notably, one of these injectors mirrors the IP TTL value from probe packets in its injected packets which has implications for the use of TTL-limited probes for localizing censors. Finally, we confirm that our observations generally hold across IP prefixes registered in China.

2.4.1 Introduction

Many countries are known to use injection of DNS responses to implement censorship [37, 60, 116, 182, 242] with China's use of DNS injection in the Great Firewall (GFW) being a popular topic for study[32, 33, 87, 100, 115, 129, 132, 168, 196, 229, 260]. While other countries tend to use NXDOMAIN or reserved IP address space [37, 43, 60, 179], China's use of a range of public IP addresses owned by a variety of organizations is notable. This use of public IP addresses can complicate detection of DNS-based censorship in China [57, 103, 182] and can make evading inadvertent DNS cache poisoning by the GFW challenging [87, 229].

While there have been numerous studies of China's DNS censorship [32, 33, 87, 100, 115, 129, 132, 168, 196, 229] (owing in part to the fact that the GFW will inject replies to clients outside of the country), in this study, we take a longitudinal approach focusing on China's use of public IPs for filtering. We measure China's DNS injector for a period of nine months which allows us to observe changes in the set of public IP addresses used by the GFW (§2.4.2). We further perform targeted measurements to fingerprint the behavior of the GFW's DNS packet injector and consider the generalizability of our results across 36K prefixes announced by Chinese ASes (§2.4.8).

Our study reveals several previously-unknown properties of China's filtering system:

IP groups. First, we observe groups of IP addresses that are used in injected replies to specific sets of domains (§2.4.6). These groups may point to groups of domains that are being blocked by a common infrastructure or blocking process. We discuss these groups in the context of blocked domains and IPs used for blocking over time (§2.4.6.2)

Three distinct injectors. We also observe that a single DNS query can result in multiple injected DNS replies from the GFW. Using IP ID, IP TTL, DNS TTL and DNS flags, we were able to fingerprint these multiple replies and identify three distinct packet injectors acting on DNS requests (§2.4.7.1).

TTL-echoing in injected packets. In the process of fingerprinting the censors, we observe one of the packet injectors will actually echo the TTL of the probe packet which has implications on the popular technique of using TTL-limited probe packets to localize network censors (§2.4.7.3).

2.4.2 Methodology

We now describe our methodology for monitoring DNS-based censorship in China on a longitudinal basis (§2.4.3) and how we extend this method to understand regional differences in filtering (§2.4.4). We also discuss steps taken to address ethical concerns while conducting our experiment (§2.4.5).

2.4.3 Baseline Longitudinal Experiment

We use the commonly employed tactic of issuing DNS queries for potentially sensitive domains from a host outside of China towards IP addresses located in China (specifically, those not hosting DNS servers). This allows us to trigger the GFW as our packet crosses the GFW, and the targeting of IP addresses not hosting DNS servers means that any response to our query can be inferred to be injected by the GFW. We issue queries from a Virtual Private Server (VPS) running Ubuntu 18.04 LTS located in a US academic network. We then send DNS queries towards a VPS under our control located in China with the same configuration as our US host. We perform our queries using the standard DNS port (53). We performed an initial test over ports 1-65535 and only observed censorship on DNS queries sent on port 53.

With this source and destination host, we then issue DNS queries for a set of tested domains. In our case, a set of 1 million domains is extracted from the Alexa top million Web sites list (accessed on Feb. 22, 2019). For any domains without the prefix "www." we add this prefix as the GFW does not consistently inject DNS replies in the absence of this prefix [32, 62]. We query these domains every two hours between September 2019 and May 2020. In total, we sent 2.8 billion DNS queries and observed 119.6 million forged responses from the GFW.

2.4.4 Multi-path Experiment

A limitation of our baseline methodology, is that we will only observe filtering on the path between our VPS in the US and our VPS in China. To complement this methodology, we perform an additional experiment where we direct DNS queries towards a broad range of Chinese IP prefixes. We identify Chinese IP prefixes by using CAIDA's AS-to-organization dataset [58] to identify ASNs registered in China. We then use CAIDA's prefix-to-AS mapping tool [59] to collect IP prefixes announced by these ASes, for a total of 36,629 prefixes.

Within each prefix, we select one IP address at random, ensuring that there is not a host at this IP address that will respond to DNS queries. To determine this, we send 10 queries for a non-sensitive domain `www.baidu.com` to the candidate IP address. If there is no reply to any of our DNS queries, we infer that this IP is not hosting a DNS server and proceed with our tests. We exclude an IP prefix from testing if we fail to find a non-responding IP address after 50 attempts. In total, we select 36,146 IP prefix, belonging to 417 Chinese ASes.

Table 2.7. FortiGuard Categories. The 10 most common categories for the domains on Alexa 1M test list, and the percentage of censored domains in each category.

Category	Alexa%	Category	Censored%
Business	27.7	Proxy Avoidance	46.0
Information Technology	13.3	Personal Websites	43.0
Shopping	5.9	Explicit Violence	20.5
Education	5.7	Extremist Groups	10.0
Personal Websites	4.4	Other Adult Material	9.4
News and Media	4.1	Content Servers	9.3
Entertainment	3.5	Dynamic DNS	7.3
Pornography	2.8	Pornography	6.2
Health and Wellness	2.7	Distcrimination	5.3
Government and Legal Orgs	2.6	Instant Messaging	4.2

For this test, we focus on a single domain `www.google.sm` that we observe triggers censorship by the three packet injectors observed in our baseline experiment (§2.4.7) since our goal is to understand the behavior of multiple network paths. We attempt 100 queries for this domain towards each of the Chinese prefixes we identify.

2.4.5 Ethics

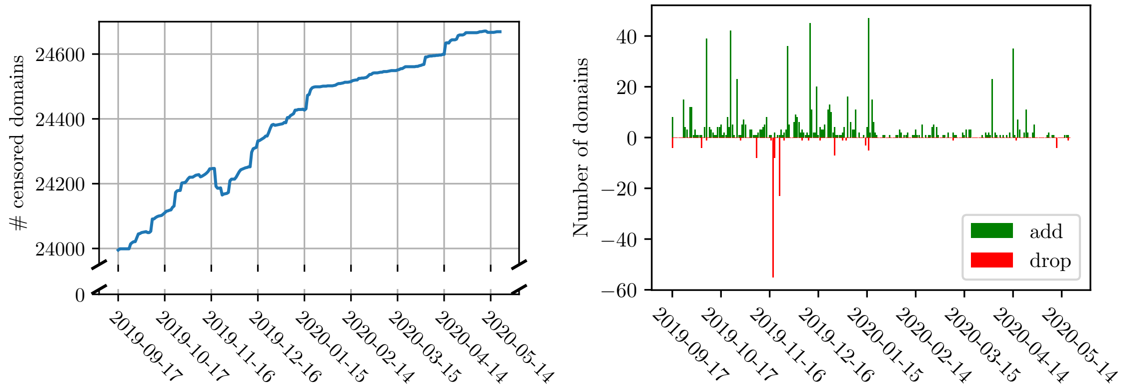
For our baseline experiment, the two hosts that we sent DNS queries to and from are machines under our control. For our multi-path experiment, we first verify that no DNS service was running on the selected IP address. We also note that our experiments are initiated from a host outside of China, thus to the GFW it appears that queries are coming from an external (academic) network, as opposed to any host within China. Finally, our multi-path experiment limits the amount of traffic sent to each IP address to 1 MB.

2.4.6 Characterizing DNS Injection

In this section, we characterize domains filtered over time (§2.4.6.1) as well as the IP addresses in the injected replies (§2.4.6.2).

2.4.6.1 Censored Domains

We see that there exists an increasing trend in the number of domains being censored by the GFW. The number of censored domains increases from 23,995 to 24,636



(a) Number of censored domains observed. (b) Changes of the number of censored domains.

Figure 2.11. Censored domain name changes among Alexa top 1 million from September 2019 to May 2020.

(2.8% increase) over our nine-month measurement study. Figure 2.11(a) presents the number of unique domains censored over time. Interestingly, previous work [32] has also shown a 10% increase in the number of censored domains over time in their 2014 study (also using the Alexa top million as their test domains).

Figure 2.11(b) depicts the daily number of domains from the Alexa top 1 million that get added and removed from the set of domains that we observe being blocked. We manually analyzed the dates in which more than 20 domains were removed from blocked set, on November 18 a group of 50 domains that all have the keyword `youtube.com` were removed and on November 22 a group of 22 domains with the keyword `line.me` were removed from the blocked set. This suggests that the GFW still operates on keywords to censor domains as opposed to curating a fixed set of domains.

Category of censored domains. We leveraged the "FortiGuard" URL classification service, operated by FortiNet [6] to categorize the Alexa top domains. The top categories within the Alexa list are listed in the left column of Table 2.7. We further analyze the percentage of censored domains in each category of the Alexa top million list. The top 10 categories with the highest percentage of domains censored

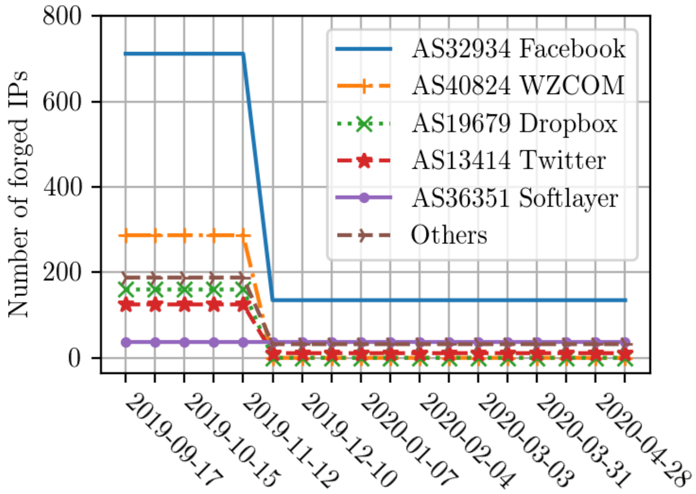


Figure 2.12. Top ASNs and the number of injected IP addresses used by the GFW belonging to each of them.

are shown in the right column of Table 2.7. We can see that 46% of the domains in the "Proxy Avoidance" category are censored by the GFW. The high number (42.9% of domains censored) for the "Personal Websites" category is because 42.7% of the censored domains within the "Personal Websites" category are domains containing the keywords `.blogspot.com`, or `.tumblr.com` which appear to be filtered by the GFW. We further analyzed and found that this is in fact a keyword based block list, i.e any domain that ends in `.blogspot.com` or `.tumblr.com` will be censored by the GFW.

2.4.6.2 Injected IPs

Longitudinal trends. We observe a set of 1,510 distinct IP addresses returned in type A DNS records injected by the GFW. While the majority of responses we observe are type A DNS records, we observe injected CNAME records for a single domain (`www.sunporno.com`). We focus on the type A records in this paper and plan to dig into the use of CNAME records by the GFW in future work.

Figure 2.12 shows the top ASes associated with the IPs injected by the GFW. We observe a total of 41 ASes associated with the injected IP addresses. Most of these ASes correspond to organizations in the US, particularly Facebook, WZCOM, Dropbox and Twitter. We note a striking decrease in the number of distinct IPs injected by the GFW on November 23, 2019 from 1,510 IPs (associated with 41 ASes) to only 216 IPs (associated with 21 ASes). We investigate this drop in injected IPs further in Section 2.4.7.

Groups of injected IPs. One property of the injected IPs that we note, is that certain subsets of blocked domains resolve to a fixed set of public IPs. That is, a group of public IPs is used to filter a given group of censored domains. Table 2.11 depicts the six distinct groups of domains we identified. We further categorized the domains in each group. The top category of domains in group 1, 2, and 3 belong to the "Proxy Avoidance" category, while 97% of the domains from group 4 and 5, include the word `google`, belonging to the "Search Engines" category. Group 6 consists of the remaining websites that are censored on the Alexa 1M that are mostly `blogspot` and `tumblr` related websites. We analyzed the IPs that were dropped from the IP pool on November 23 and found that 99% of the domains that received those IPs currently receive 197 injected IPs (Group 6), the majority (99%) of these domains have the keyword `tumblr.com` in them.

Reachability of the injected IP addresses. Given that China is using publicly routable IP addresses, a natural question is whether these IPs are hosting content or are otherwise reachable on the broader Internet. We test the reachability of the injected IPs from our VPS in China and the United States by initiating TCP handshakes on port 80 and port 443. We perform this experiment daily for 7 days and present the averaged result in Figure 2.13. We note that each days results looked similar. In the majority of cases (60.9%), the TCP handshake attempt results in a TIMEOUT both for source hosts in the US and China, indicating there is likely no

Table 2.8. Overview of the relationship between the sensitive domain, forged IP groups and injectors after the decrease in the number of injected IP addresses.

Group	Domains	IPs	Top categories%
1	8	3	Proxy Avoidance 50.0% Business 25.0% Personal Websites 12.5%
2	53	4	Proxy Avoidance 36.0% News and Media 9.4% Instant Messaging 7.5%
3	48	10	Proxy Avoidance 79.2% Information Technology 10.4% Info and Computer Security 2.1%
4	33	4	Search Engines 96.9% Dynamic DNS 3.1%
5	54	201	Search Engines 96.3% Business 1.8% Unknown 1.8%
6	˜24K	197	Personal Websites 76.7% Pornography 6.3% Information Technology 2.8%

content being served from these IPs at the time of our measurements. It is possible these IPs were observed serving content at some point in the past which resulted in their addition to the set of injected IPs.

2.4.7 Understanding the GFW Injectors

We now characterize cases where multiple injected DNS replies are observed. We are able to fingerprint these replies and identify three distinct injection processes (§2.4.7.1). We characterize longitudinal trends of the injectors (§2.4.7.2). Finally, We also localize these injectors and observe peculiar mirroring of the probe-TTL value by one injector (§2.4.7.3).

2.4.7.1 Fingerprinting the Injectors

In our measurements, we observed cases where a single DNS query may result in multiple injected DNS replies. Upon closer inspection, we were able to identify three distinct fingerprints within these multiple injected replies based on IP Do-not-Fragment (DF), IP TTL , DNS Authoritative Answer (AA), and DNS TTL fields.

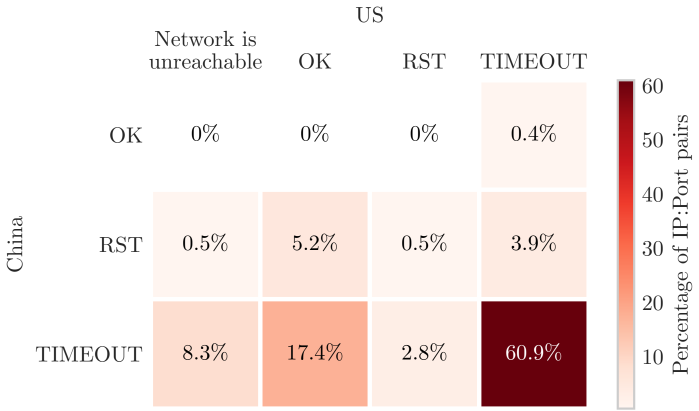
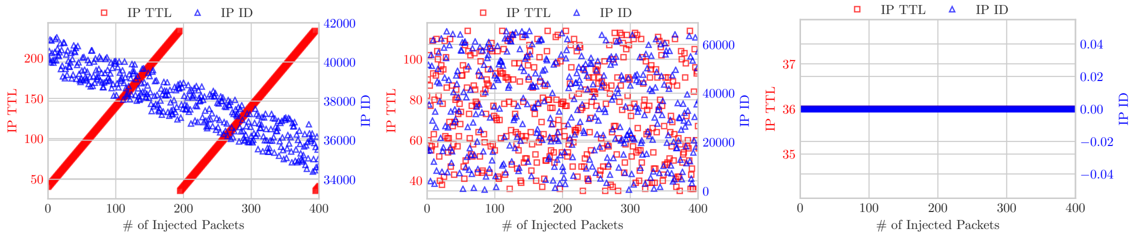


Figure 2.13. Reachability of the ports 80 and 443 of the injected IPs from China and from the US. The numbers are averaged over seven days.



(a) Injector 1

(b) Injector 2

(c) Injector 3

Figure 2.14. IPID and TTL values observed for the three DNS Injector behaviors observed in our measurements. Injector 1 is similar to what has previously been observed in [32]. We observe that the third injector reflects the IP TTL value, leading to a fixed value when the initial IP TTL values of our queries are not varied.

Table 2.9 summarizes the fingerprints of the three injectors and Figure 2.14 plots the IPID and TTL values for these three injectors when queries are sent in rapid succession[6]. We also find that the three injectors also behave slightly differently in how they format their DNS responses. Specifically, Injector 1 uses the domain from the query as-is in the DNS response, whereas Injectors 2 and 3 use a "compression pointer" [177] to reduce repetition of the query domain in the response, perhaps a sign of these injectors using a different code base in their operation.

[6] In this test, we injected packets as fast as we could using a multi-threaded Python program while using tcpdump to capture the response packets.

Table 2.9. Summary of the three DNS injectors. "DNS AA" refers to the DNS Authoritative Answer flag. "IP DF" refers to the IP "do not fragment" flag.

Injector	Description	IPs	Domains	IP Group
1	DNS: TTL=60; AA=1 IP: DF=0 incrementing IP TTL	4	88	4, 5, 6
2	DNS: AA=0 IP: DF=1 randomized IP TTL	1,506	24,729	1, 2, 3 5, 6
3	DNS: AA=0 IP: DF=0; ID=0 fixed IP TTL	958	22,948	1, 2, 3, 5

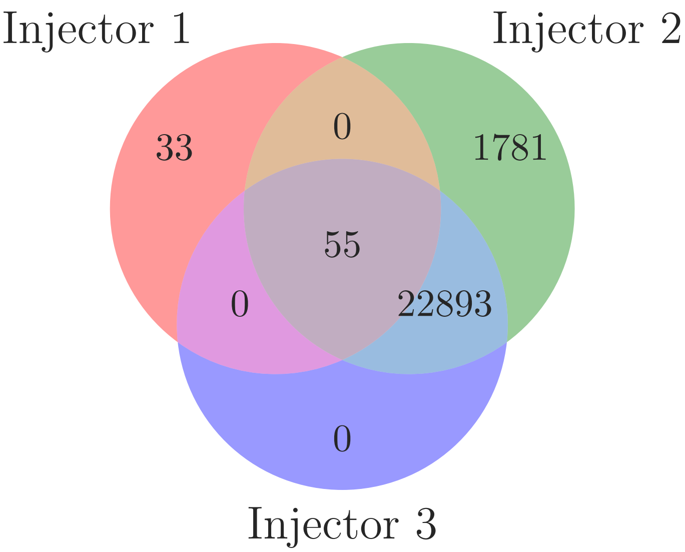


Figure 2.15. Venn diagram showing the number of domains receiving different combinations of injected responses by the three observed DNS injectors.

Similar to prior work [32], we observe Injector 1 with an incrementing IP TTL value between subsequent packets. However, we see this injector is considerably less active in terms of the number of domains it filters. Figure 2.15 shows the number of domains that observed an injected reply from each injector. We can see that Injector 1, which most closely resembles the injector seen in 2014 [32], only filtering a total of 88 domains.

Interestingly, we do not observe any domains that *only* trigger Injector 3, with it acting on a subset of Injector 2's domains. When we consider the relationship

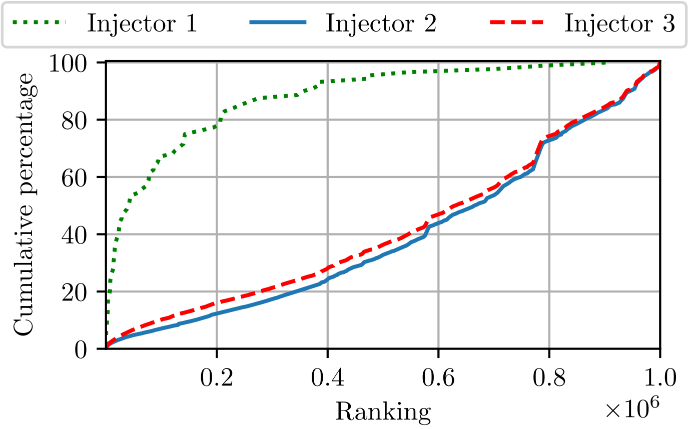


Figure 2.16. CDF of the popularity ranking of censored domains by each injector.

between the Injectors and the IP/domain groups (Table 2.9), we see that Injector 1 is the only injector filtering IPs in the fourth IP/Domain group with 33 domains that are mostly in the "Search Engines" category (cf. Table 2.11).

While Figure 2.15 gives a sense of the number of domains filtered by each injector, it doesn't necessarily reflect how often the injector would be triggered. For this, we consider the popularity of domains that each injector acts on. Figure 2.16 shows the cumulative percentage of domains filtered by each injector relative to their Alexa ranking. Here we see that domains filtered by Injector 1 tend to be more popular than those filtered by the other injectors. Most of the domains (97%) censored by Injector 1 are domains that contain the keyword `google`, and 90% of them are in the top 350K domains in the Alexa top 1M list. While, the majority (80%) of domains censored by Injectors 2 and 3 are `*.blogspot` and `.*tumblr` domains which are in the long tail of the Alexa 1M list [221].

2.4.7.2 Longitudinal trends

Halting interval of injectors. Figure 2.17 shows the total number of injected packets on a daily basis. Due to the frequency of our measurements, we are not able to discover any gaps less than two hours. When analyzing the data on a bi-hourly

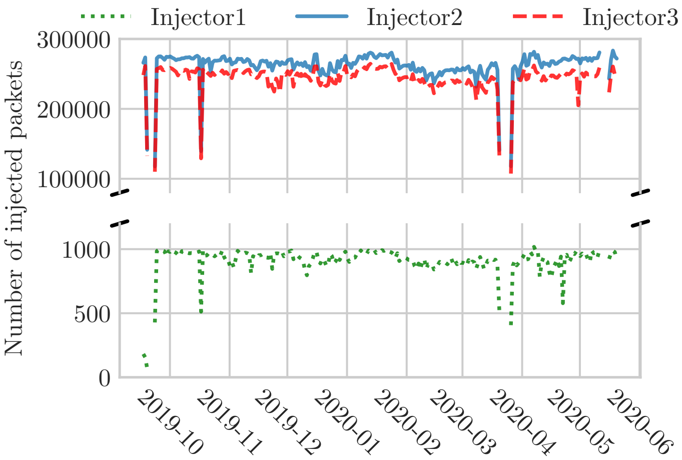


Figure 2.17. Total number of injected packets per injector received each day across time. The gaps are all due to disruptions of the measurements.

basis, we discover that while Injector 2 has been working consecutively, Injector 1 and Injector 3 occasionally stopped working for a few hours. Specifically, the three halting intervals of Injector 1 are between 13:00 and 15:22 on September 18, 2019; between 9:26 and 13:00 September 19, 2019; and between 17:06 to 10:22 on September 19, 2019. The only halting intervals of Injector 3 are between 2:36 and 8:00 on May 1 (in Beijing Time). We note the actual halts are likely to be a sub-interval of what we have discovered. All of these occasionally happened halts lasted less than 6 hours and most of them happened during work hours in China.

Relationship between injectors and the IP drop seen in Figure 2.12. We analyzed the IPs used by the injectors over time, specifically before and after the decrease in the number of distinct IPs injected on November, 2019. The decrease has no effect on Injector 1 as it always uses the same four distinct IPs. However, Injector 2 and Injector 3 initially use a pool of 958 and 1,506 IPs to send injected DNS replies, respectively. After the drop, both Injector 2 and 3 use the same IP pool (with 212 IPs) for their injected DNS replies.

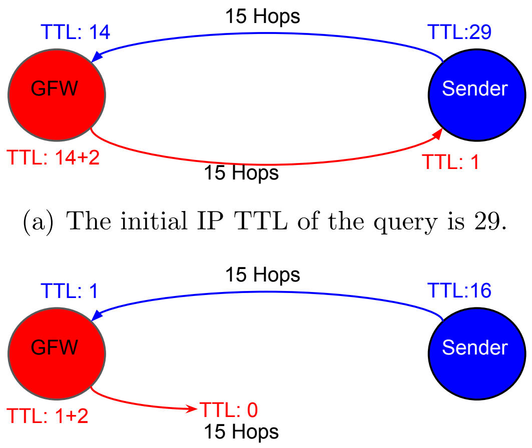
2.4.7.3 Localizing the Injectors

We next attempt to localize the three injectors identified in §2.4.7.1. We use the commonly employed method of sending packets with incrementing IP TTL values until we receive an injected DNS reply to identify where on our path the packet injector lies [32, 168, 201, 229, 257]. For this test, we focus on a single domain that we observed to trigger all three injectors: `www.google.sm`. We then send DNS queries for this domain from our VPS in the US to the VPS in China.

Based on these TTL limited probes, we were able to observe that Injectors 1 and 2 are located 15 hops away from our US VPS. For comparison, our Chinese VPS is 25 hops from our US VPS. However, we observed an unusual behavior with Injector 3, where we did not see an injected DNS reply from Injector 3 until the initial TTL on our probe packet is set to 29. Given that the destination IP of our probe packet was only 25 hops away, this behavior seemed unusual. However, upon closer inspection, we determined that this behavior stemmed from Injector 3 echoing the incremented TTL of the probe packet in its injected reply.

Figure 2.18 illustrates this phenomenon. We find that when the probe packet has a TTL of 29, the injected reply has an IP TTL of 1 when it reaches our US host. Similarly, when the probe packet has a TTL of 30 the TTL of the injected reply is 2, and so on. The precise probe TTL needed to observe this behavior is $2n - 1$ where n is the number of hops between the probing host and the packet injector. We note, that this discussion implicitly assumes symmetric paths between the injector and the probing host. This behavior could potentially be used to identify asymmetric routing on paths (when a domain that will trigger multiple injectors is used), but we leave more in depth analysis of this to future work.

We also compare the time between sending our DNS query and when we receive the injected reply to get a sense of where the injectors are located. Specifically, we compare the delays of the three injectors and find that more than 90% of the time



(a) The initial IP TTL of the query is 29.

(b) The initial IP TTL of the query is 16.

Figure 2.18. Illustration of how Injector 3 mirroring the IP TTL of the DNS query impacts the results of TTL-limited probing. Figure 2.18(a) shows that when the IP TTL of the DNS query is 29 the corresponding injected packet has a high enough TTL to reach the sender. Figure 2.18(b) shows that when the IP TTL of the DNS query is below 29, the initial IP TTL of the forged response is too small to reach the sender.

the delays are within 0.2 ms of each other. This would support the theory that these three devices are installed in the same physical location.

We repeat these experiments from seven hosts outside of China (our VPS in the US and cloud-hosted VMs in the Netherlands, Singapore, UK, France, Canada and India) with consistent results.

2.4.8 Multi-path Results

In Section 2.4.2, we describe our method to send our DNS queries to 36K Chinese prefixes. Our goal here is to confirm that our results are robust, to the location of the host that we focus on for our longitudinal experiment. Figure 2.19 shows the result: each bar corresponds to the combination of injectors that were observed and the height of the bar corresponds to the percent of prefixes where this combination of injectors was observed.

Of the 36K prefixes we direct our query towards, we find that 62% of them observe all three DNS injectors. We observe 12% of cases where two of the three injectors are observed, and 13% of cases where only one of the three injectors are observed. For each IP address, we send 100 queries which suggests that these cases are not just caused by transient packet loss. We also observe some injectors that are not seen in our longitudinal data in this broader study (denoted by Injector X in Figure 2.19). In total, there are around 4% of the prefixes where we observe injectors not matching our fingerprints.

Interestingly, we see 8% of the prefixes, registered to 134 ASes, where no DNS injector is triggered. Using the RIPE NCC AS visibility tool [216], we find 22% of these prefixes have less than 15% visibility, suggesting our queries may never reach these prefixes. For the remaining prefixes, we use the RIR-based IP-to-ASN mapping provided by Team Cymru [233] and find that half of these prefixes are registered outside of China (e.g., a Chinese-based company registering an IP address with ARIN). In these cases, the prefixes may be located outside of China and not subject to censorship. It is worth noting that there are still 1,027 IP prefixes that seem to be within China's territory, but with no injected packet observed. These IP prefixes correspond to 120 ASes. Upon closer inspection we find that these ASes tend to be related to technology companies or government agencies.

2.4.9 Conclusion

In this work, we analyze the DNS poisoning behavior of the GFW across nine months. We observe groups of IPs used to censor specific groups of domains and identify three distinct DNS packet injectors. We localize and characterize the behavior of these injectors and identify one injector mirroring the TTL of the probe packets which has implications for studies that use TTL-limited packets to localize DNS censors. We have released our code and dataset to maintain reproducibility and to

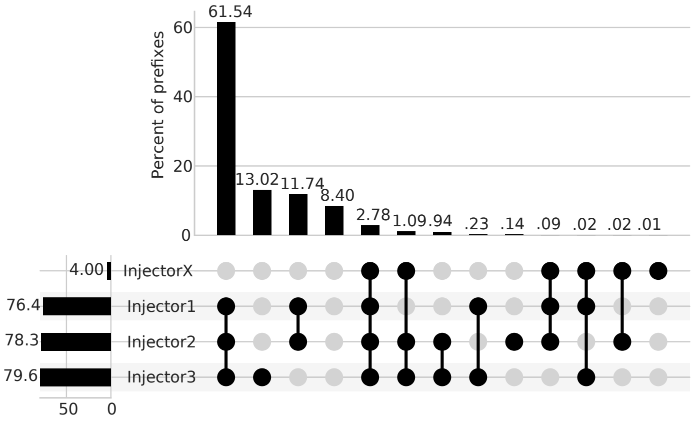


Figure 2.19. Number of unique IP prefixes responding with different types of responses. InjectorX refers to the injectors that have fingerprints other than the summarized ones.

stimulate future work, obtainable at `https://gfw.report/publications/foci20_dns/en/`.

2.5 How Great is the Great Firewall? Measuring China's DNS Censorship

The DNS filtering apparatus of China's Great Firewall (GFW) has evolved considerably over the past two decades. However, most prior studies of China's DNS filtering were performed over short time periods, leading to unnoticed changes in the GFW's behavior. In this study, we introduce GFWatch, a large-scale, longitudinal measurement platform capable of testing hundreds of millions of domains daily, enabling continuous monitoring of the GFW's DNS filtering behavior.

We present the results of running GFWatch over a nine-month period, during which we tested an average of 411M domains per day and detected a total of 311K domains censored by GFW's DNS filter. To the best of our knowledge, this is the largest number of domains tested and censored domains discovered in the literature. We further reverse engineer regular expressions used by the GFW and find 41K innocuous domains that match these filters, resulting in overblocking of their content.

We also observe bogus IPv6 and globally routable IPv4 addresses injected by the GFW, including addresses owned by US companies, such as Facebook, Dropbox, and Twitter.

Using data from GFWatch, we studied the impact of GFW blocking on the global DNS system. We found 77K censored domains with DNS resource records polluted in popular public DNS resolvers, such as Google and Cloudflare. Finally, we propose strategies to detect poisoned responses that can (1) sanitize poisoned DNS records from the cache of public DNS resolvers, and (2) assist in the development of circumvention tools to bypass the GFW's DNS censorship.

2.6 Introduction

Among the censorship regimes on the Internet, China is one of the most notorious, having developed an advanced filtering system, known as the Great Firewall (GFW), to control the flow of online information. The GFW's worldwide reputation [107] and ability to be measured from outside the country, has drawn the attention of researchers from various disciplines, ranging from political science [38, 64, 80, 82] to information and computer science [32, 88, 96, 173, 183, 253].

Unlike many other DNS censorship approaches, the GFW is known to return globally routable IP addresses in its injected responses. Recent studies [129, 133, 183] have observed injected IP addresses belonging to popular US companies, including Facebook, Dropbox, and Twitter. The use of routable IPs is in contrast to countries such as Bahrain, Korea, Kuwait, Iran, Oman, Qatar, Thailand, or Yemen [111, 133, 144, 182, 205], where DNS censorship redirects users to blockpages that inform users about the blocked content. It is also in contrast to censors using fixed DNS responses such as NXDOMAIN [43, 180, 182, 196] or addresses from private IP ranges (e.g., 10.0.0.0/8) [37, 61, 196]. This use of globally routable IPs by the GFW has implications for censorship detection, which needs to carefully distinguish censored from legitimate DNS responses, and also makes detecting and mitigating leaked DNS responses from public resolvers non-trivial.

Despite the many previous studies that examine the technical strategies employed by the GFW, such as TCP/IP packet filtering [69, 88, 96, 193, 253] and DNS poisoning [32, 86, 100, 229], there has yet to be a large-scale, longitudinal examination of China's DNS filtering mechanism. This lack of visibility is apparent as the number of censored domains and the pool of IP addresses used by the GFW in forged DNS responses have been reported differently by previous studies [32, 48, 100, 168, 183, 196, 229, 266]. In particular, the number of fake IPs observed in poisoned responses has been increasing from nine in 2010 [48], 28 in 2011 [229], 174 in 2014 [32], to more than 1.5K recently [183]. To that end, it is necessary to have a system for continuous, long-term monitoring of the GFW's filtering policy that will provide timely insights about its blocking behavior and assist censorship detection and circumvention efforts.

In this work, we developed GFWatch (§2.7), a large-scale, longitudinal measurement platform to shed light on DNS filtering by the GFW and assess its impact on the global Internet. By building GFWatch, our primary goal is not only to answer the questions of *(1) how many censored domains are there* and *(2) what are the forged IP addresses used in fake DNS responses*, but also to assess *(3) the impact of the GFW's DNS censorship policy on the global Internet*, and ultimately design *(4) strategies to effectively detect and circumvent the GFW's DNS censorship.*

Using GFWatch, we tested a total of 534M distinct domains (averaging 411M domains per day) and detected a total of 311K censored domains (§2.8). We then used the set of censored domains to design a probing method that is able to reverse-engineer the actual blocklist used by the GFW's DNS filter (§2.8.1). Using this list, we observed that 270K out of the 311K censored domains are censored as intended, whereas the remaining 41K domains appear to be innocuous despite matching regular expressions used by the GFW. Through our measurements, we discovered 1,781 IPv4 and 1,799 IPv6 addresses used by the GFW in forged DNS responses (§2.9). To the best of our knowledge, these are the largest sets of censored domains and forged IP addresses ever discovered.

We also found evidence of geographic restrictions on Chinese domains, with the GFW injecting DNS replies for domains based in China (e.g., `www.beian.gov.cn`) (§2.10). While previous studies attribute leakage of Chinese DNS censorship to cases where a DNS resolver's network path transits through China's network [48, 229], we found that geoblocking and cases where censored domains have at least one authoritative name server located in China are also a significant cause of pollution of external DNS resolvers (§2.10.1).

Based on the observed censored domains (§2.8) and forged IP addresses (§2.9), we propose strategies to effectively detect poisoned DNS responses injected by the GFW (§2.10.2). These techniques will not only help public DNS resolvers and other DNS-related services to sanitize tainted records (§2.10.2), but can also assist future development of circumvention tools to bypass the GFW's DNS censorship (§2.11).

2.7 GFWatch Design

We designed GFWatch according to the following requirements: (1) the platform should be able to discover as many censored domains and forged IPs as possible in a timely manner. More specifically, GFWatch should be able to obtain and test new domain names *as they appear on the Internet*. (2) As a longitudinal measurement platform, once a domain is discovered to be censored, GFWatch should continuously keep track of its blocking status to determine whether the domain stays censored or becomes unblocked at some point in the future. (3) By measuring many domains with sufficient frequency, GFWatch is expected to provide us with a good view into the pool of forged IPs used by the GFW.

2.7.1 Test Domains

We are interested in the timely discovery of as many censored domains as possible because we hypothesize that the GFW does not block just well-known domains (e.g., `facebook.com`, `twitter.com`, `tumblr.com`) but also less popular or even unranked ones that are of interest to smaller groups of at-risk people (e.g., political dissidents, minority ethnic groups), who are often suppressed by local authorities [17]. Therefore, we opt to curate our test list from top-level domain (TLD) zone files obtained from various sources, including Verisign [4] and the Centralized Zone Data Service operated by ICANN [2], which we refresh on a daily basis. Using zone files not only provides us with a good coverage of domain names on the Internet, but also helps us to fulfill the first design goal of GFWatch, which is the capability to test new domains as they appear on the Internet.

Since TLD zone files contain only second-level domains (SLDs), they do not allow us to observe cases in which the GFW censors subdomains of these SLDs. As we show later, many subdomains (e.g., `scratch.mit.edu`, `nsarchive.gwu.edu`, `cs.colorado.edu`) are censored but their SLDs (e.g., `mit.edu`, `gwu.edu`, `colorado.edu`) are not. We complement our test list by including domains from the Citizen Lab test lists (CLTL) [67], the Tranco list [162], and the Common Crawl project [1]. Between April and December 2020, we tested a total of 534M domains from 1.5K TLDs, with an average of 411M domains daily tested.

2.7.2 Measurement Approach

When filtering DNS traffic, the GFW does not consider the direction of request packets. As a result, even DNS queries originating from outside the country can trigger the GFW if they contain a censored domain, making this behavior a popular topic for measurement studies [32, 48, 183, 229]. Based on the observation of this filtering policy, we design GFWatch to probe the GFW from outside of China to dis-

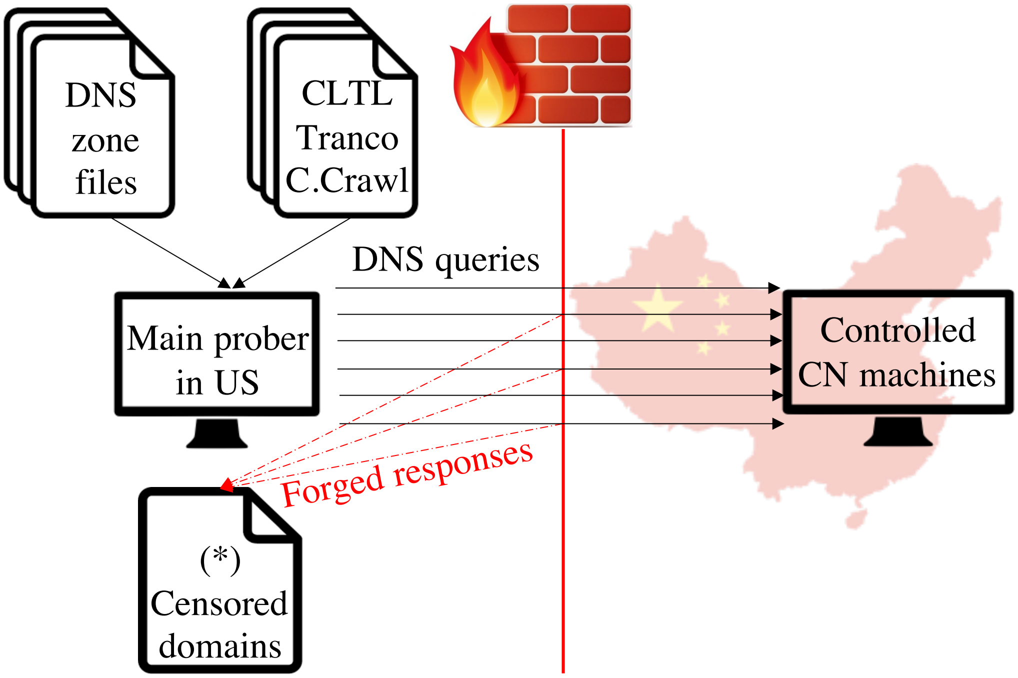


Figure 2.20. Probing the GFW's DNS poisoning from outside.

cover censored domains and verify their blockage again from our controlled machines located in China to validate our findings.

Prior work has shown that the GFW does not filter DNS traffic on ports other than the standard port 53 [168, 183], we thus design our probe queries using this standard destination port number. We observe that for major UDP-based DNS query types (e.g., `A`, `CNAME`, `MX`, `NS`, `TXT`), the GFW injects the forged responses with an IPv4 for type `A` queries and a bogus IPv6 for type `AAAA` queries. In some rare cases, injections of forged static CNAME records are also observed for a small number of censored domains (§2.9.4).

For TCP-based queries that carry censored domains, RST packets are injected instead of DNS responses [246]. Since UDP is the default protocol for DNS in most operating systems, we choose to probe the GFW with UDP-based queries. While using both TCP-based and UDP-based queries would still allow us to detect censored domains, we opt to use UDP-based queries because they also allow us to (1) collect the forged IPs used in the injected DNS responses, and (2) conduct our measurement at scale, which would be otherwise more challenging to achieve because a TCP-based measurement at the same scale would require more computing and network resources to handle stateful network connections.

As shown in Figure 2.20, GFWatch's main prober is a machine located in an academic network in the United States, where DNS censorship is not anticipated. `A` and `AAAA` DNS queries for the test domains are sent towards two hosts in China, which are under our control and do not have any DNS resolution capabilities. Therefore, any DNS responses returned to the main prober come from the GFW.

While prior studies have confirmed the centralized blocking policy of the GFW [80, 114, 224], to make sure this behavior is still consistent and to detect any future changes, the two hosts in China are located in two different autonomous systems (ASes). From our measurement results, we confirm that the DNS blocking policy continues to be centralized, with the same censored domains detected via the two probing paths.

After the main prober completes each probing batch, detected censored domains are transferred to the Chinese hosts and probed again from inside China towards our control machine, as shown in Figure 2.21. This way, we can verify that censored domains discovered by our prober in the US are also censored inside China.

Since GFWatch is designed to probe using UDP, which is a stateless and unreliable protocol, packets may get lost due to factors that are not under our control (e.g., network congestion). Moreover, previous studies have reported that the GFW sometimes fails to block access when it is under heavy load [96, 183]. Therefore, to minimize the impact of these factors on our data collection, GFWatch tests each domain at least three times a day.

For this paper, we use data collected during the last nine months of 2020, from April to December. As of this writing, GFWatch is still running and collecting data

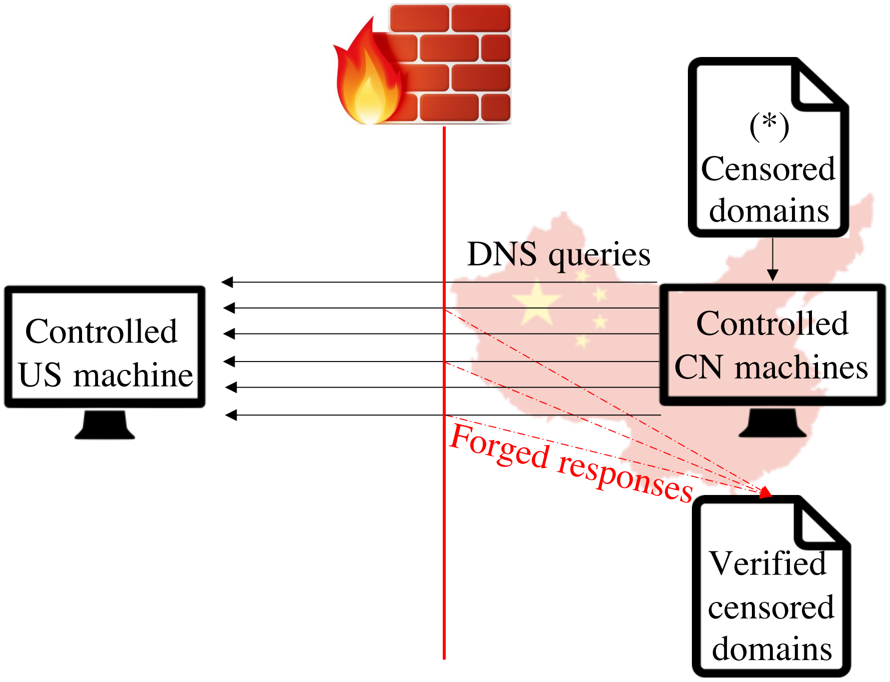


Figure 2.21. Verifying poisoned domains from inside the GFW.

every day. The data collected will be made available to the public on a daily basis through a dedicated web service.

2.8 Censored Domains

Over the nine months of our study, we tested a total of 534M distinct domains, finding 311K domains triggering the GFW's DNS censoring capability. Figure 2.22 summarizes the cumulative number of censored domains over time, as well as the number of domains added and removed from the set of censored domains each day. We note a sharp increase in domains on August 31st because of the addition of more than 30K subdomains from the previously censored namespaces (e.g., `*.googlevideo.com`, `*.appspot.com`) to our test domains. In this section, we describe our technique for identifying the specific strings that trigger GFW's DNS censorship (§2.8.1). We use this technique to remove unrelated domains that match the blocking rules ("overblocked" domains) and then characterize domains censored by the GFW in Section 2.8.2.

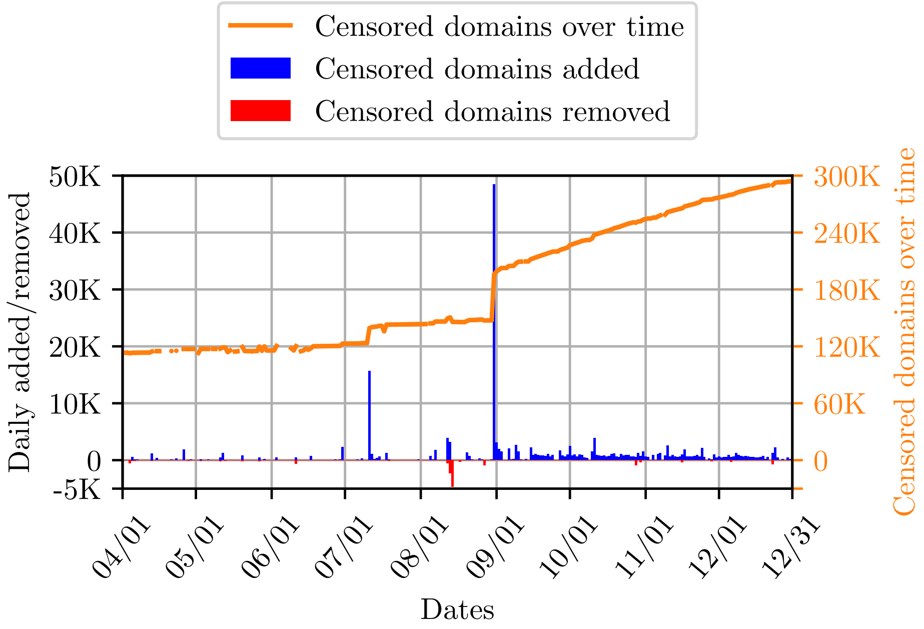


Figure 2.22. Cumulative censored domains discovered over time and daily added/removed censored domains.

2.8.1 Identifying Blocking Rules

When considering the domains filtered by the GFW, there are many with common second-level and top-level domains (e.g., numerous blocked domains of the form `*.blogspot.com` or `*.tumblr.com`). This observation led us to develop a clustering method for domains that are blocked based on the same underlying rule. For example, if `subdomain.example.com` and all subdomains of `example.com` are blocked, we consider `example.com` as the blocked domain. We note that when a subdomain is blocked, the covering domains may not be blocked (e.g., `cs.colorado.edu` is blocked, whereas `colorado.edu` is not (§2.8.2)).

Inspired by a previous study of GFW's DNS censorship [32], we use the following technique to identify the strings that trigger blocking (i.e., the most general string such that all domains containing this string are blocked). For a given domain, we test the following permutations of each censored domain and random strings:

- Rule 0 `censored_domain`
- Rule 1 `censored_domain`{*`.rnd_str`*}

- Rule 2 censored_domain{*rnd_str*}
- Rule 3 {*rnd_str.*}censored_domain
- Rule 4 {*rnd_str*}censored_domain
- Rule 5 {*rnd_str.*}censored_domain{*.rnd_str*}
- Rule 6 {*rnd_str.*}censored_domain{*rnd_str*}
- Rule 7 {*rnd_str*}censored_domain{*.rnd_str*}
- Rule 8 {*rnd_str*}censored_domain{*rnd_str*}

Among these rules, only Rules 1 and 3 are correct forms of a domain with a different top-level domain (Rule 1) or subdomain (Rule 3). In contrast, the rest represents unrelated (or non-existent) domains that happen to contain the censored domain string. We refer to censored domains that are grouped with a shorter domain string via rules other than Rules 1 or 3 as being *overblocked*, because they are not subdomains of the shorter domain, but are actually unrelated domains that are textually similar (e.g., the censored domain mentorproject.org contains the shorter domain string torproject.org that actually triggers censorship).

Using these rules to generate domains and testing them with GFWatch, we identify the most general form of each censored domain that triggers censorship. We refer to these shortest censored domains as the "base domain" from which the blocking rule is generated. We discovered a total of 138.7K base domains from the set of 311K censored domains.

Considering base domains allows us to observe growth in the underlying blocking rules as opposed to the raw number of domains. We also observe fewer new base domains over time and avoid sudden jumps in censored domains when large numbers of subdomains of an existing base domain are observed. Figure 2.23 shows the cumulative number of base domains discovered over the nine-month period and the daily addition and removal of these domains. As of December 31st, 126K base domains are still being censored.

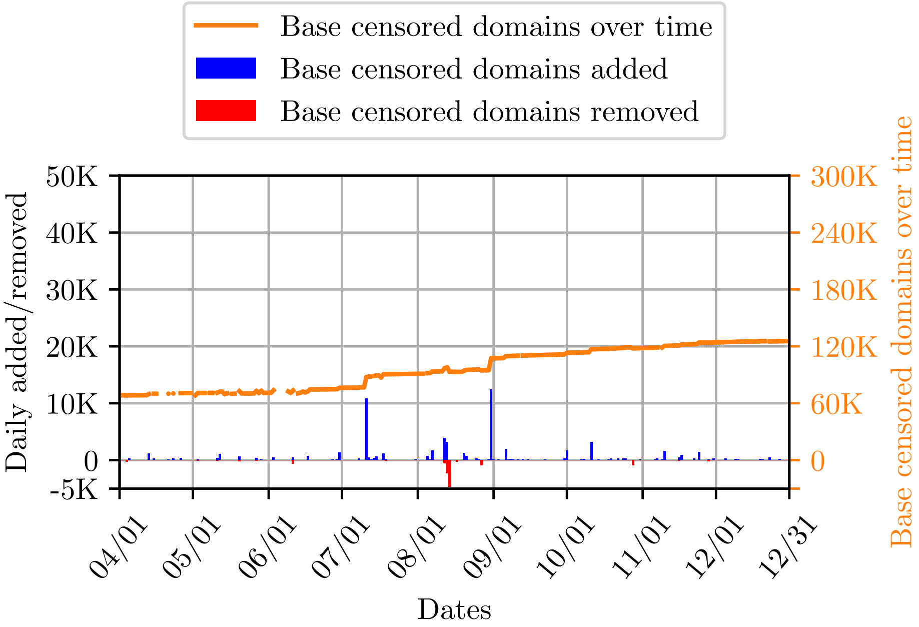


Figure 2.23. Cumulative base censored domains discovered over time and daily added/removed base censored domains.

Of 138.7K base domains, 11.8K are censored independently (Rule 0). In other words, these domains are censored as they are, but do not trigger GFW's DNS censorship when concatenated with random strings. However, in an ascending order of severity, we find that 4, 113.8K, 10.9K, 1.4K, and 696 distinct base domains are blocked under Rules 2, 3, 4, 6, and 8, respectively. There are no domains for Rules 1, 5, and 7, since domains blocked under these rules are already covered by other more general rules. While the vast majority of base censored domains fall under Rule 3, there are more than 13K base domains blocked under other rules, causing unrelated domains to be overblocked.

We utilize the base domains to identify cases of overblocking, where an unrelated domain matches a more general censored domain string. Specifically, we consider domains that match a base domain, but are not subdomains of the base domain, as being overblocked. This is because these domains are unrelated to the base domain despite being textually similar. With this definition, we find that 41K of the 331K censored domains are overblocked. The top three base domains that cause the most

Table 2.10. Top base censored domains that cause most overblocking of innocuous domains.

# domains impacted	Base censored domains	Sample innocuous domains
11,227	919.com	455919.com, rem99919.com niwa919.com, xaa919.com
2,346	jetos.com	ccmprojetos.com, csprojetos.com itemsobjetos.com, dobobjetos.com
1,837	33a.com	87833a.com, 280333a.com xn---72caa7c0a9clrce0a1fp33a.com xn---zck4aye2c2741a5qvo33a.com
1,574	9444.com	mkt9444.com, 15669444.com 3329444.com, 5719444.com
1,547	sscenter.net	dentalwellnesscenter.net, swisscenter.net chesscenter.net, childlosscenter.net
1,487	1900.com	faber1900.com, salah1900.com phoenixspirit1900.com, interiors1900.com
1,392	98a.com	p98a.com, 72898a.com, 1098a.com xn---1-ieup4b2ab8q5c0dxj6398a.com
1,144	ss.center	hss.center, icass.center limitless.center, ass.center
1,089	reddit.com	bestiptvreddit.com, booksreddit.com cachedreddit.com, geareddit.com
789	visi.tk	erervisi.tk, yetkiliservisi.tk buderuservisi.tk, bodrumklimaservisi.tk

overblocking are 919.com, jetos.com, and 33a.com. These three domains are responsible for a total of 15K unrelated domains being blocked because they end with one of these three base domains (and are not subdomains of them). Table 2.10 provides more details on the base domains responsible for the most overblocking. Domain owners may consider refraining from registering domain names containing these base domains to avoid them being inadvertently blocked by the GFW.

Table 2.10 shows the top ten base censored domains blocked under Rule 4 that we have discussed in §2.8.1. The blocking rule applied on these ten domains results in overblocking of more than 24K innocuous domains, which is more than half of all innocuous domains. The third column shows some samples of innocuous censored domains that GFWatch has discovered. The impacted innocuous domains presented in this table are all active and hosting some contents at the time of writing this paper. Except those that do not allow Web Archive's crawler, we have also saved a snapshot of these domains at `https://web.archive.org` for future reference in case these domains become inactive. In contrast, most base censored domains shown in the second column are not currently hosting any content. Therefore, one may wonder why many seemingly inconsequential domains are being censored.

To make sure that these seemingly inconsequential censored domains were not blocked because the GFW was using an imprecise classifier (e.g., a Bloom filter) for fast classification, we tested 200M randomly generated nonexistent domains and found that none were censored. It is worth noting that many censored domains discovered by GFWatch have been blocked before the launch of our platform. Prior to our testing, they might have served "unwanted" content that we were not aware of. Moreover, the GFW is known to conduct blanket blocking against websites that run editorials on "unwanted" topics without carefully verifying their contents. Once domains are censored, they are often kept in the GFW's blocklist for a long time regardless of their activity [197].

As can be seen from the table, the GFW's overblocking design affects not only usual ASCII-based innocuous domains, but also Internationalized Domain Names (IDNs), i.e., those starting with `''xn---''`. Of 41K innocuously blocked domains, we find a total of 1.2K IDNs are overblocked. Our finding shows that the current DNS-based blocking policy of the GFW has a widespread negative impact on the domain name ecosystem.

2.8.2 Characterizing Censored Domains

We now characterize the 138.7K base domains identified in §2.8.1. We focus on these base domains to avoid the impact of domains with numerous blocked subdomains on our results. Focusing on base domains also allows us to avoid analyzing innocuous domains that are overblocked based on our previous analysis.

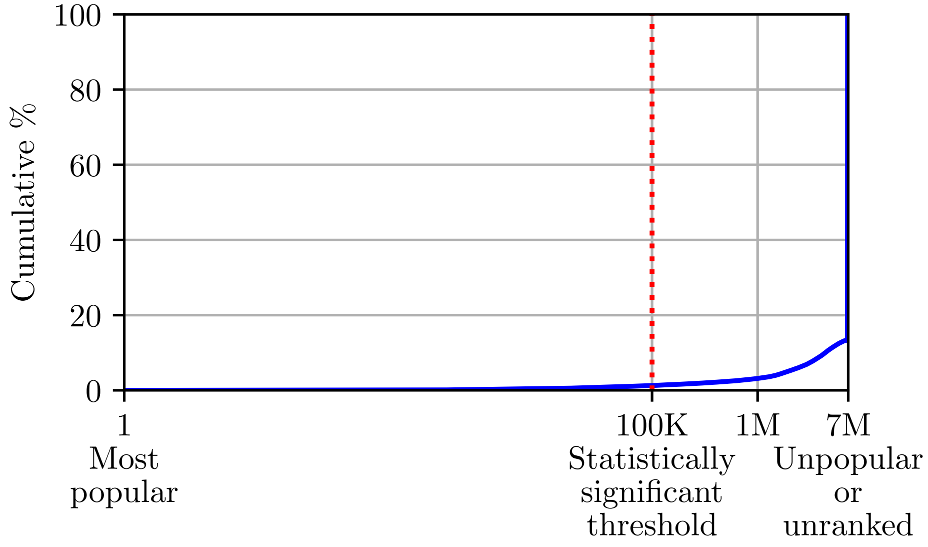


Figure 2.24. CDF of the popularity ranking for base censored domains (in log scale).

Popularity of censored domains. We find that most domains blocked by the GFW are unpopular and do not appear on lists of most popular websites. We use the rankings provided by the Tranco list [162], which combines four top lists (Alexa [26], Majestic [172], Umbrella [237], and Quantcast [202]) in a way that makes it more stable and robust against malicious manipulations [198]. The daily Tranco list contains about 7M domains ranked by the Dowdall rule [?].

Figure 2.24 shows the CDF of the popularity ranking for the 138.7K blocked base domains. Only 1.3% of them are among the top 100K most popular domains, which is the statistically significant threshold of the popularity ranking as suggested by both top-list providers and previous studies [27, 219]. Even when considering all domains ranked by the Tranco list, only 13.3% of the base censored domains fall within the list's ranking range, while the remaining are unranked. This finding highlights the importance of GFWatch's use of TLD zone files to enumerate the set of potentially censored domains.

Types of censored content. For domain categorization, we use a service provided by FortiGuard [6], which has also been used by other censorship measurement stud-

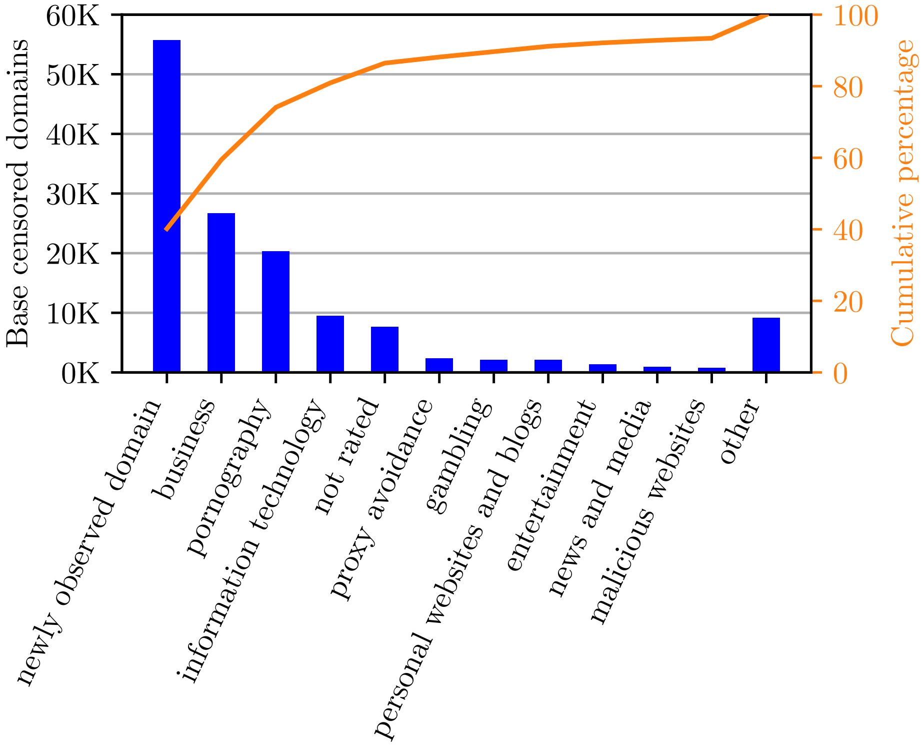


Figure 2.25. Top ten categories of domains censored by the GFW.

ies [182, 183, 230], to make our analysis comparable. Figure 2.25 shows the top-ten domain categories censored by the GFW. We find that nearly half of the domains we observe are not currently categorized by FortiGuard, with 40% categorized as *"newly observed domain,"* and 5.5% categorized as *"not rated."* This is a result of the large number of domains in our dataset, many of which may not be currently active (§2.11.3).

Apart from the *"newly observed domain"* and *"not rated"* categories, we find that *"business," "pornography,"* and *"information technology"* are within the top-five dominant categories. This finding is different from the results reported by the most recent related work to ours [183], which observed *"proxy avoidance"* and *"personal websites and blogs"* as the most blocked categories. This difference stems from the counting process used in [183], which does not aggregate subdomains, while their test list is a fixed snapshot of 1M domains from the Alexa list, which contains many subdomains of `*.tumblr.com` and `*.blogspot.com`.

COVID-19 related domains. On December 19th, 2020, the New York Times reported that the Chinese Government issued instructions for suppressing the free flow of information related to the COVID-19 pandemic [212]. GFWatch has detected numerous domains related to COVID-19 being censored by the GFW through DNS tampering, including `covid19classaction.it`, `covid19song.info` `covidcon.org`, `ccpcoronavirus.com`, `covidhaber.net`, and `covid-19truth.info`.

While most censored domains are discovered to be blocked soon after they appear in our set of test domains, we found that there was some delay in blocking `ccpcoronavirus.com`, `covidhaber.net`, and `covid-19truth.info`. Specifically, `ccpcoronavirus.com` and `covidhaber.net` first appeared on our test lists in April but are not blocked until July and September, respectively. Similarly, `covid-19truth.info` appeared in our dataset in September but was not censored until October. The large difference in the time the GFW takes to censor different domains shows that the blocklist is likely to be curated by both automated tools and manual efforts.

Educational domains. In 2002, Zittrain et al. [266] reported DNS-based filtering of several institutions of higher education in the US, including `mit.edu`, `umich.edu`, and `gwu.edu`. While *"education"* is not one of the top censored categories, we find numerous blocked education-related domains, including `armstrong.edu`, `brookings.edu`, `citizenlab.ca`, `feitian.edu`, `languagelog.ldc.upenn.edu`, `pori.hk`, `soas.ac.uk`, `scratch.mit.edu`, and `cs.colorado.edu`.

Although censorship against some of these domains is not surprising, since they belong to institutions well-known for conducting political science research and may host content deemed as unwanted, we are puzzled by the blocking of `cs.colorado.edu`. While the University of Colorado's computer science department is not currently using this domain to host their homepage, the blocking of this domain and its entire namespace `*.cs.colorado.edu` would prevent students in China from accessing other department resources (e.g., `moodle.cs.colorado.edu`). This is another evidence of the overblocking policy of the GFW, especially during the difficult time of the COVID-19 pandemic when most students need to take classes remotely.

2.9 Forged IP Addresses

The use of publicly routable IPs owned by foreign entities not only confuses the impacted users and misleads their interpretation of the GFW's censorship, but also hinders straightforward detection and circumvention [120]. Therefore, knowing the forged IPs and the pattern in which they are injected (if any) is essential. In this section, we analyze the IPs collected by GFWatch to examine whether there exists any specific injection pattern based on which we can develop strategies to effectively detect and bypass the GFW's DNS censorship.

2.9.1 Forged IP Addresses over Time

Extracting the forged IPs from all poisoned DNS responses captured by GFWatch, we find a total of 1,781 and 1,799 unique forged IPv4 and IPv6 addresses from poisoned type-A and type-AAAA responses, respectively. The forged IPv4 addresses are mapped to multiple ASes owned by numerous non-Chinese entities, including 783 (44%) IPs of Facebook, 277 (15.6%) IPs of WZ Communications Inc., 200 (11.2%) IPs of Twitter, and 180 (10.1%) IPs of Dropbox. On the other hand, all IPv6 addresses are bogus and belong to the same subnet of the predefined Teredo prefix [142], `2001::/32`. Therefore, we will focus our analysis on the forged IPv4 addresses hereafter because the pattern of IPv6 injection is obvious and thus should be trivial to detect and circumvent.

Figure 2.26 shows the number of unique IPv4 addresses that GFWatch has discovered over the measurement period considered in this paper. The gray bar plot shows the number of unique IPs observed daily, and the blue bar plot shows the number of

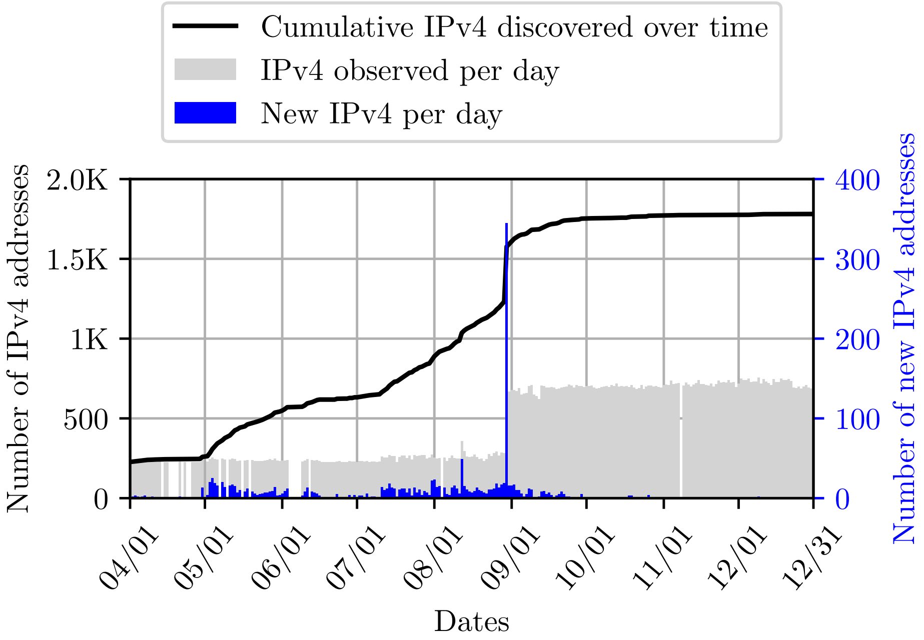


Figure 2.26. Number of forged IPv4 addresses detected over time by GFWatch.

new IPs that were not observed previously. We add a second y-axis on the right side of the figure for better visibility of the blue bars.

Our initially collected data overlaps with the data collected during the final month of [183], which is the most recent related work to our study. During this period, our observation aligns with the result reported in Figure 2 of [183], i.e., the number of unique forged IPs is about 200 with no new IPs detected. However, starting in May, GFWatch began to detect more forged IPs every day until September, with about 10–20 new IPs added daily. These gradual daily additions, together with a significant increase of more than 300 previously unobserved IPs at the end of August, have brought the total number of forged IPs to more than 1.5K. The number of forged IPs converges to 1.7K over the last four months of 2020.

Comparing the IPs observed by GFWatch with the ones reported in [183], we find that all IPs observed by [183] have been used again in poisoned DNS responses, regardless of the major drop reported on November 23rd, 2019. In addition, we find 188 new IPs that were not observed previously in [183]. Given how close the timeline is between our work and [183], this finding of the unpredictable fluctuation in the number of forged IPs emphasizes the importance of having a large-scale longitudinal measurement system to keep track of erratic changes in the GFW's blocking behavior. Therefore, we are committed to keeping GFWatch running as long as possible, rather than just creating it as a one-off effort.

Prior reports [80, 114, 224] and our detection of the same censored domains via two different network paths (§2.7) have confirmed the centralized blocking policy of the GFW in terms of the domains being censored. Nevertheless, we are also interested in investigating whether the forged IPs are consistent at different network locations, because our ultimate goal is to collect as many forged IPs as possible and demystify their injection pattern to assist us in developing effective strategies for censorship detection and circumvention. Therefore, we have also conducted an extra measurement by probing across different network locations in China to confirm that the pool of forged IPs discovered by GFWatch is representative enough.

2.9.2 Consistency of Forged IP Addresses Across Different Network Locations

To confirm whether the pool of forged IPs discovered by GFWatch (§2.9) is representative enough, we probe different network locations in China to compare the forged IPs observed from these locations and the ones seen by GFWatch. For this experiment, we obtain the daily updated *pfx2as* dataset provided by CAIDA [59], and extract prefixes located in China by checking them against the MaxMind dataset [174], which we also update biweekly. Unlike the measurement conducted between our own controlled machines located at two sides of the GFW, this task requires us to send DNS queries, encapsulating censored domains, to destinations we do not own. Although similar large-scale network probing activities are widely conducted nowadays

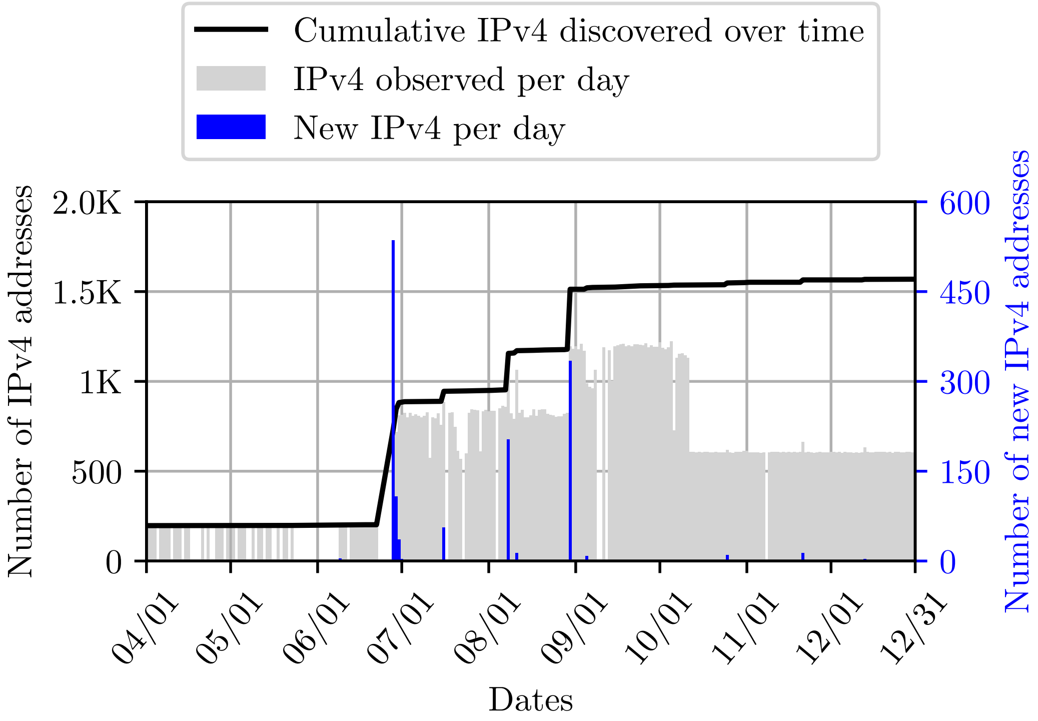


Figure 2.27. Number of forged IPv4 addresses detected over time by probing different network prefixes in China.

by both academia [89, 196, 230, 240] and industry [7, 208], our measurement must be designed in a careful and responsible manner.

Our sole purpose of this measurement is to deliver probing queries passing through the GFW's infrastructure at different network locations to trigger censorship, instead of having the probing packets completely delivered to any alive hosts. Therefore, we craft our probing packets using the routing address of a given prefix as the destination IP. According to the best current practice [109], except for the case of a `/32` subnet with only one IP, the routing address of a subnet should not be assigned to any device because it is solely used for routing purposes. For example, given the prefix `1.92.0.0/20` announced in the *pfx2as* dataset, we craft our probing packet with the destination as `1.92.0.0`. With this probing strategy, we can reduce the risk that our packets will hit an alive host while still being able to deliver them across the GFW's infrastructure at different network locations. To reduce the risk even further, we opt to only probe prefixes whose subnet is less-specific than `/24`.

In spite of the standardized practices in assigning IP and the extra care that we have taken in designing our measurement, we also follow a common practice that is widely used in research activities that involve network scanning, i.e., allowing opt-out. More specifically, we accompany our probing DNS queries with a non-censored domain under our control, from which the information about our study and a contact email address can be found to request opt-out from our measurement. Since the launch of GFWatch, we have not received any complaints or opt-out requests.

Figure 2.27 show the cumulative number of forged IPs discovered daily and over the whole period of our measurement. Similar to Figure 2.26, the number of forged IPs addresses observed initially in April is also about 200. However, we did not see any gradual increase in the number of forged IPs from May as seen in Figure 2.26. After waiting about two months without seeing any new IPs observed from probing different prefixes, we have learned that this is due to the fact that we only use *one* known censored domain for probing the prefixes. This is because of an earlier precaution that these probed destinations are not owned by us, thus we should try to limit the amount of probing traffic as much as possible. However, it turned out that we need to probe more than just one domain to be able to obtain a similar set of forged IP addresses detected earlier by GFWatch.

We then decide to add more domains to this test, probing a total of 22 censored domains per prefix. These domains are selected from several categories, including advocacy organizations, proxy avoidance, news and media, social network, personal websites and blogs, shopping, instant messaging, etc. As expected, the cumulative number of forged IPs immediately increases to almost 1K the day we revise our test domains. Similar to Figure 2.26, the cumulative number of forger IPs also increase gradually towards the end of August. With a major increase of more than 300 forged

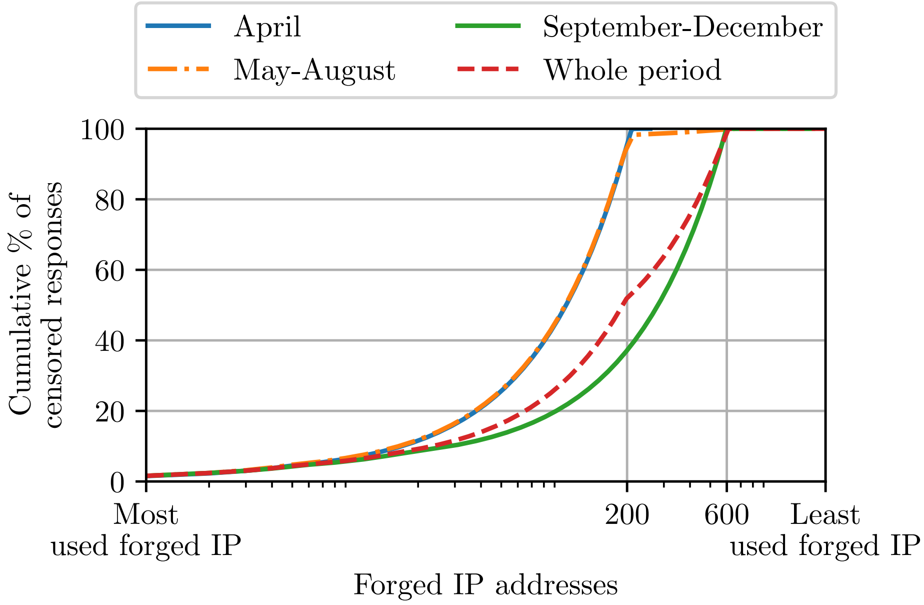


Figure 2.28. CDF of censored responses with respect to the injection frequency of forged IPv4 addresses detected by GFWatch.

IPs, the number of all forged IPs observed from our prefixes probing measurement also converges to above 1.5K by the end of December.

While the number of forged IPs obtained from probing the prefixes on some days, especially from July to September, is higher than what GFWatch observed during this period, we find that 96% of the forged IPs observed from prefixes probing have already detected by GFWatch. Conducting the same injection frequency analysis on these forged IPs gives us the same results as found in §2.9.3. In other words, the most frequently injected IPs discovered by GFWatch and from probing different prefixes are the same. To this end, we could confirm that the coverage of forged IPs discovered by GFWatch is representative and sufficient for us to develop effective detection (§2.10.2) and circumvention strategies (§2.11).

2.9.3 Injection Frequency of Forged IPs

Due to the erratic changes in the number of forged IPs over time, prior studies have often concluded that forged IPs are injected randomly. Through the longitudinal measurement conducted at scale, GFWatch has tested and detected a large enough number of censored domains and forged IPs that allows us to provide more insights into this aspect. Analyzing the injection frequency of each forged IP, we find that not all forged IPs are equally injected in censored responses, i.e., their injection pattern is not entirely random.

Figure 2.28 shows the CDF of censored responses with respect to the injection frequency of forged IPs observed in these responses. The x-axis (in log scale) indicates the number of forged IPs, sorted by their injection frequency. There are three periods during which the cumulative number of forged IPs shows different patterns (i.e., April, May to August, and September to December, as shown in Figure 2.26). Thus, we analyze the injection frequency of these three periods independently and compare them with the injection frequency of all forged IPs discovered over the whole period of our measurement.

We can see that the forged IPs' injection frequencies are similar (almost overlapping) between the April and May–August lines. In other words, although the number of forged IPs increases from about 200 at the end of April to more than 1.5K over the May–August period, the initial 200 forged IPs are still responsible for 99% of censored responses. On the other hand, the additional 1.3K new forged IPs discovered from May to August are in the long tail and only used in 1% of all censored responses. Similarly, even after the remarkable increase to more than 1.7K forged IPs at the end of August, only 600 of them are frequently injected from September to December, occupying 99% of the censored responses. Finally, when looking at all the censored responses and forged IPs discovered over the whole period, the 200 most frequently injected forged IPs discovered in April are still responsible for more than 50% of all censored responses, whereas only 600 (33.6%) out of 1,781 forged IPs are responsible for 99% of all censored responses, the remaining 1.1K forged IPs in the long tail are used in only 1% of censored responses.

Table 2.11. Groupings of censored domains with respect to different sets of forged IPs injected in their poisoned responses.

G	# Domains	# IPs	Forged IPs/CNAMEs
0	41	0	cathayan.org, mijingui.com, upload.la, yy080.com
1	12	1	why.cc → 216.139.213.144
2	7	1	yumizi.com → 66.206.11.194
3	57	1	46.38.24.209, 46.20.126.252, 61.54.28.6, 89.31.55.106 122.218.101.190, 123.50.49.171, 173.201.216.6, 208.109.138.55
4	3,295	3	4.36.66.178, 64.33.88.161, 203.161.230.171
5	1,711	4	8.7.198.45, 59.24.3.173, 243.185.187.39, 203.98.7.65
6	2,724	4	8.7.198.46, 59.24.3.174, 46.82.174.69, 93.46.8.90
7	4	7	4.36.66.178, 64.33.88.161, 203.161.230.171, 59.24.3.174 8.7.198.46, 46.82.174.69, 93.46.8.90
8	9	7	4.36.66.178, 64.33.88.161, 203.161.230.171, 8.7.198.45 59.24.3.173, 243.185.187.39, 203.98.7.65
9	4,551	10	23.89.5.60, 49.2.123.56, 54.76.135.1, 77.4.7.92 118.5.49.6, 188.5.4.96, 189.163.17.5, 197.4.4.12 249.129.46.48, 253.157.14.165
10	remaining $\sim 300K$ domains	>560	[Omitted due to the large number of forged IPs] Supplementary data will be made publicly available and updated on a daily basis.

2.9.4 Static and Dynamic Injections

One of the GFW behaviors is injecting different sets of forged IPs for different groups of censored domains. This behavior was first reported in [183], where the authors identify a total of six groups of censored domains that are poisoned with different sets of forged IPs. From data collected by GFWatch, we have discovered a total of 11 groups shown in Table 2.11. Comparing these groups with those reported in [183], we find five similar groups that have the same set of forged IPs/CNAMEs, including Groups 0, 4, 5, 6, and 9. Understandably, we discover more groups because our test list covers far more domains compared to [183], where a fixed Alexa top list of only 1M domains was used for the whole measurement period.

An instance of forged response containing a CNAME was reported in [183] but excluded from the analysis since it did not seem to be prevalent. However, with a larger dataset, we find that the injection of CNAME in forged responses can happen in three different groups of censored domains, triggering the GFW to inject six different CNAME answers. As depicted in Table 2.11, there are 41 censored domains that can trigger the injection of *either one of the four* CNAMEs listed. Domains in Groups 1 and 2 can trigger a CNAME injection, accompanied by an IP in the forged response. Note that these two IPs are not the actual IPs of the two CNAMEs. Similarly, there are eight distinct subgroups of domains within Group 3 that can constantly trigger *either one of the eight* forged IP listed. For example, `qcc.com.tw` will always trigger a forged response of `89.31.55.106`. The same pattern applies in other Groups from 4 to 9, i.e., resolving domains within these groups will always trigger the GFW to inject one of the forged IPs listed on the 4th column. The remaining of about 300K censored domains are grouped together since they trigger the GFW to dynamically inject a much larger number of more than 560 different forged IPs.

Revealing these injection patterns for different groups of censored domains is crucial for developing an effective strategy to detect and circumvent the GFW's DNS censorship (§2.10). Especially, knowing whether a censored domain belongs to one of the static groups (Groups 0 to 9) or the dynamic group (Group 10) is necessary to avoid misclassifying consistent forged responses as "legitimate" (§2.11).

2.10 Censorship Leakage and Detection

The GFW's bidirectional DNS filtering behavior has been reported as the cause of poisoned DNS responses being cached by public DNS resolvers outside China, when DNS resolution paths unavoidably have to transit via China's network [133, 229]. However, in this section, we show that DNS poisoning against many domains whose authoritative name servers are located in China is another primary reason why poisoned DNS records have tainted many public DNS resolvers around the world. We then show how the datasets of censored domains and forged IPs discovered by

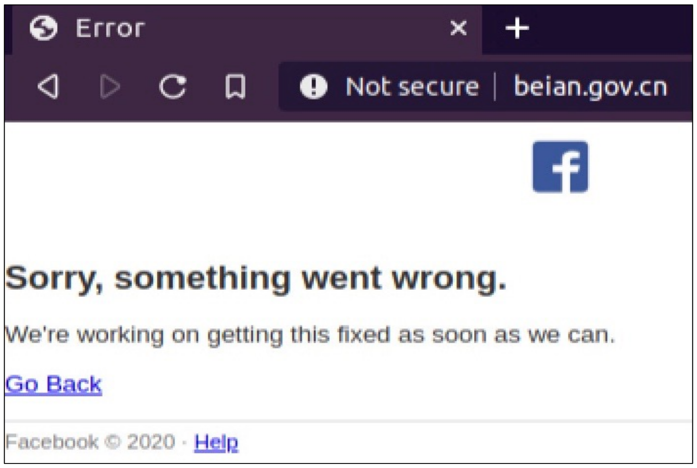
GFWatch can help with detecting and sanitizing poisoned resource records from public DNS resolvers' cache.

2.10.1 Geoblocking of China-based Domains

On August 8th, 2020, GFWatch detected the blockage of `www.beian.gov.cn`, which is managed by the Chinese Ministry of Industry and Information Technology. This service allows website owners to obtain and verify their website's Internet Content Provider (ICP) license, which is obligated to legally operate their site in China. This domain has two authoritative name servers, `dns7.hichina.com` and `dns8.hichina.com`, which are hosted on 16 different IPs. However, checking against the latest MaxMind dataset [174], we find that all of these IPs are located inside China. Consequently, the DNS censorship against this domain by the GFW will cause DNS queries issued from outside China to be poisoned since all resolution paths from outside China will have to cross the GFW.

We initially attributed this blockage to an error or a misconfiguration because previous works have sometimes noticed intermittent failures in the GFW [96, 183]. Furthermore, no prior studies have ever found such a strange blocking behavior—the GFW of China censors a Chinese government website. However, at the time of composing this paper, we are still observing `www.beian.gov.cn` being censored by the GFW, almost half a year since its first detection. Hence, this is a clear case of geoblocking because we can still visit this domain normally from our controlled machines located inside China. To the best of our knowledge, ours is the first academic research to document this geoblocking behavior of the GFW.

Note that this geoblocking is a result of the GFW's DNS censorship, which is not the same as geoblocking enforced at the server side [175]. Geoblocking of China-based websites has been noticed previously but is enforced by their website owners. For instance, political researchers have been using `https://www.tianyancha.com/`



Sorry, something went wrong.

We're working on getting this fixed as soon as we can.

Go Back

Facebook © 2020 · Help

Figure 2.29. Visit to a domain geoblocked by the GFW ends up with an error page from Facebook.

to investigate the ownership of Chinese companies, but since 2019, this website blocks visitors from non-Chinese IPs and shows a clear message for the reason of denying access.

The GFW's blocking of China-based domains using bidirectional DNS filtering in combination with the use of forged IPs owned by non-Chinese entities impacts not only Internet users in China, but also users from around the world. For instance, upon visiting the aforementioned geoblocked domain from a non-censored network outside China, we end up with an error page served from Facebook, as shown in Figure 2.29.

Most ordinary Internet users would not know the underlying reason why their visit to a given China-based domain (e.g., `www.beian.gov.cn`) that is clearly unrelated to Facebook would end up with an error page from Facebook. The fact that the GFW frequently changes the forged IPs used in fake DNS responses (§2.9) would cause even more confusion to the affected users. Depending on which fake IP is injected in the spoofed response, users may encounter a different error page from Figure 2.29. Even more confusing, the visit to this domain from outside China will intermittently succeed because the poisoned responses injected by the GFW sometimes fail to arrive ahead of the legitimate one (§2.11).

At the server side of the forged IPs being used for injecting poisoned responses, their operators would also be puzzled as to why many HTTP requests are sent to their servers, asking for hostnames they do not serve. For the above example, an error log at a Facebook server will show that someone was trying to visit `www.beian.gov.cn` on a Facebook IP, which obviously does not serve any content for that domain, thus the returned error page. As we do not have access to the error logs of Facebook and other organizations whose IPs are used for injecting poisoned DNS responses by the GFW, we cannot quantify the actual cost (e.g., the overhead of serving unsolicited connections, error pages) of such an abusive DNS redirection behavior. However, given the large number of more than 311K censored domains discovered (§2.8) and only a small pool of forged IPs being used (§2.9), we believe that the GFW's injection policy would cost these affected organizations a non-negligible overhead on their servers. Past reports have shown that this abusive design of the GFW can lead to resource exhaustion attacks on specific IPs, making them inaccessible [71, 120, 143].

To estimate the extent to which the above geoblocking and overblocking policies have impacted the global Internet, we analyze the location of authoritative name servers of 138.7K base censored domains and 41K innocuously blocked domains, using the MaxMind dataset [174]. As shown in Figure 2.30, 38% (53K) of the base censored domains and 21.6% (8.8K) of the innocuous censored domains have at least one authoritative name server in China. In other words, there is always a non-zero chance that DNS resolution for these 61.8K domains from outside China will be poisoned, causing their visitors to potentially end up with an error page similar to the above case. On the other hand, 19.4% (26.9K) of base censored domains and 12.5% (5.1K) innocuously blocked domains have all of their authoritative name servers in China,

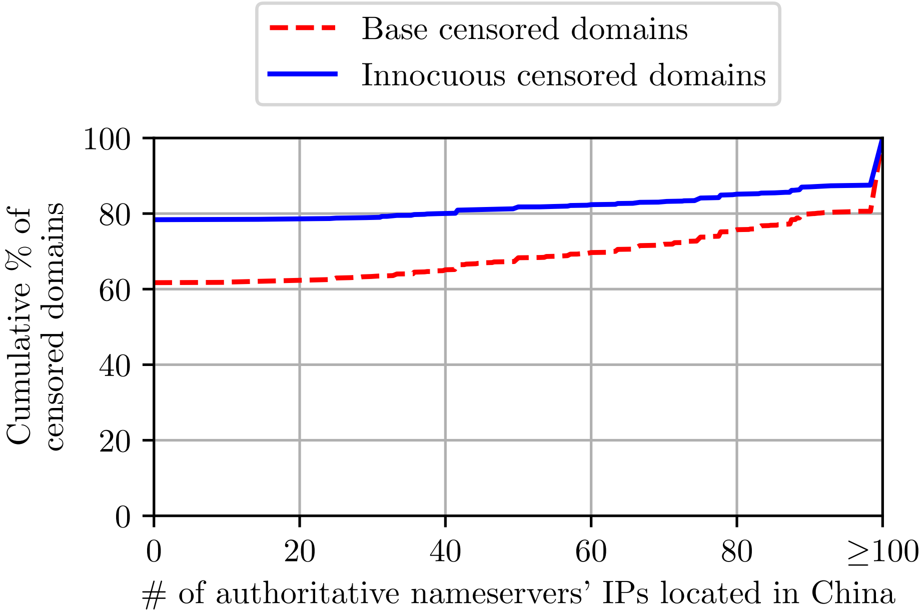


Figure 2.30. CDF of the number of authoritative name servers located inside China as a percentage of 138.7K base censored domains and 41K innocuously blocked domains.

meaning that the resolutions for these 32K domains from outside China will always cross the GFW, thus being poisoned.

2.10.2 Detection

A common operational mechanism of DNS censorship is that the censor takes advantage of the time-honored property of UDP-based DNS resolution to inject poisoned responses, racing against the legitimate response. Depending on the censored domain being queried, the GFW can even emit up to three responses. This behavior of injecting multiple fake responses was first reported recently in [183]. For the completeness of our investigation, we have also identified the three different injectors based on the data collected by GFWatch.

It was first reported by [183] that the GFW comprises multiple injectors that are responsible for DNS poisoning. Depending on the domain being queried (e.g., `google.sm`), multiple forged responses can be triggered simultaneously to increase the chance of successfully poisoning censored clients if one of the injectors is overloaded, and make detection and circumvention non-trivial. From the data collected by GFWatch, we have confirmed the same injection behavior. More specifically, there are three injectors, which can be differentiated by the *"DNS Authoritative Answer"* flag in the DNS header and the *"do not fragment"* flag in the IP header. Injector 1 has the *"DNS Authoritative Answer"* bit set to **1**, Injector 2 has the *"DNS Authoritative Answer"* bit set to **0** and *"do not fragment"* bit set to **1**, whereas Injector 3 has the *"DNS Authoritative Answer"* bit set to **0** and *"do not fragment"* bit set to **0**.

Based on these fingerprints, we then cluster 311K censored domains into three groups with respect to the three injectors. Figure 2.31 depicts the number of censored domains observed over time for each injector. Injector 2 is responsible for 99% of the censored domains, whereas Injectors 3 and 1 are responsible for only 64% and less than 1% (2K) of censored domains, respectively. Note that all domains censored by Injector 3 are also censored by Injector 2, while there are 1.7K domains censored only by Injector 1, but not other injectors.

From the GFW's perspective, the injection of multiple fake responses not only increases the chance of successfully poisoning a censored client but also makes it more costly and challenging to detect and circumvent its DNS censorship [86]. However, based on the pool of forged IPs and their injection patterns that we have revealed in §2.9, detecting DNS censorship by the GFW can be done effectively by checking the returned IP address against the pool of forged IPs discovered by GFWatch. Although this strategy may not detect all poisoned responses due to some rare forged IPs that GFWatch might have not observed in the long tail, from the analysis of injection frequency in §2.9.3, which we have also verified its consistency across different network locations (§2.9.2), we are confident that this detection technique can identify more than 99% of the poisoned responses.

We next employ this detection technique to expose poisoned resource records that have tainted public DNS resolvers around the world. In particular, once a censored

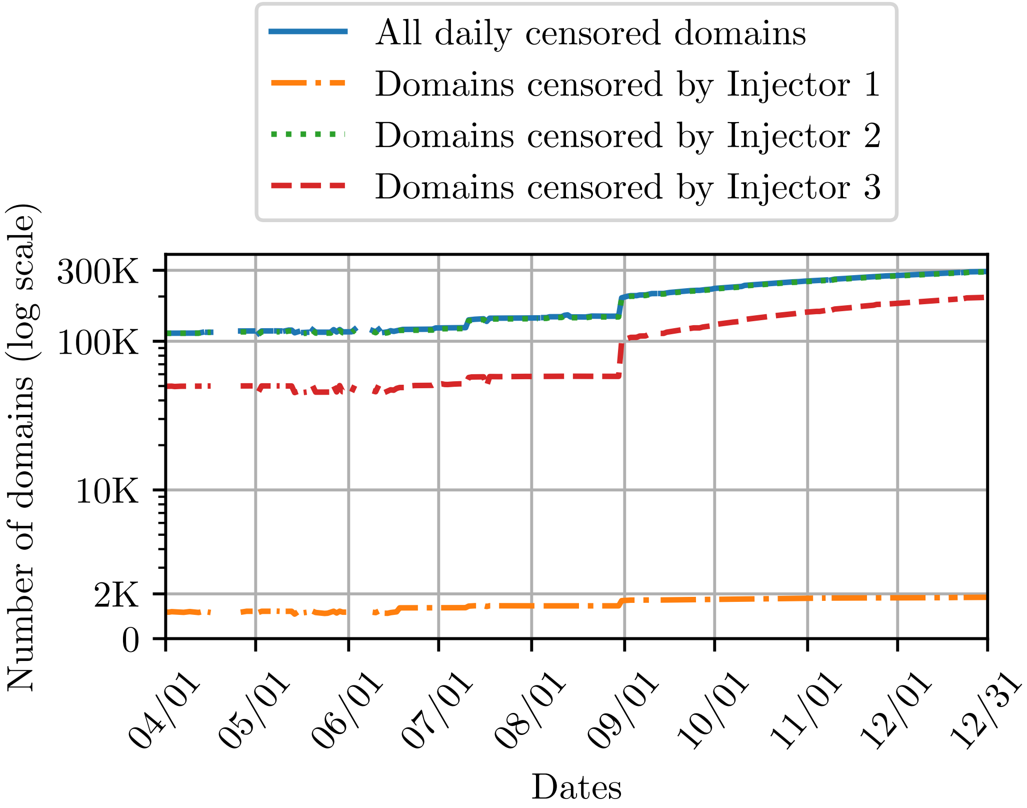


Figure 2.31. Number of censored domains per injector.

domain is detected by GFWatch, we query them against popular DNS resolvers and examine if its response matches any injection pattern we have revealed in §2.9. Table 2.12 shows the top ten resolvers that have been polluted with the highest number of censored domains. In total, we find 77K censored domains whose poisoned resource records have polluted the cache of all popular public DNS resolvers that we examined. Of these censored domains, 61K are base censored domains. This result aligns well with our earlier speculation in §2.10.1.

This finding shows the widespread impact of the bidirectional blocking behavior of the GFW, necessitating the operators of these public DNS resolvers to have an effective and efficient mechanism to prevent these poisoned resource records from polluting their cache, to assure the quality of their DNS service. Furthermore, the 61K base censored domains whose DNS queries from outside China are censored is likely

Table 2.12. Top ten public DNS resolvers with the highest number of censored domains whose poisoned resource records have polluted their cache.

# Domains	Resolver	# Domains	Resolver
74,715	Google	63,295	OpenDNS
71,560	Cloudflare	62,825	Comcast
65,567	OpenNIC	56,913	CleanBrowsing
65,538	FreeDNS	56,628	Level3
64,521	Yandex	55,795	Verisign

the reason why many censored domains are classified as *"newly observed domain"* or *"not rated"* in §2.8.2. This is because FortiGuard's crawlers, which are likely located outside China, probably could not obtain the correct IPs of these domains, thus failing to fetch and classify them.

2.11 Circumvention

We now show how insights gained from analyzing the censored domains (§2.8) and forged IPs discovered by GFWatch over time (§2.9) can assist us in developing strategies to effectively and efficiently circumvent GFW's DNS censorship.

2.11.1 Strategy

The GFW's bidirectional DNS filtering not only impacts in-China users but also prevents users outside China from obtaining legitimate resources records of geographically restricted domains based in China (§2.10.1). Therefore, an effective DNS censorship evasion strategy would benefit not only (1) users inside China who need to access censored domains hosted outside China, but also (2) users outside China who need access to geoblocked domains based in China. Both (1) and (2) also include open DNS resolvers located at both sides of the GFW that want to prevent poisoned responses from polluting their DNS cache.

Since the GFW operates as an on-path injector and does not alter the legitimate response from the actual DNS resolver chosen by a client, a circumvention strategy for the client is to not quickly accept any returned responses when querying a censored domain. Instead, the client should wait for an adjustable amount of time for all responses to arrive, as suggested in [86]. Upon receiving more than one IPv6 answer, the client can filter out the bogus ones that belong to the Teredo subnet `2001::/32`. Furthermore, for IPv4 answers, the client can check them against the injection patterns and forged IPv4 addresses discovered in §2.9.

In our circumvention strategy, for each censored domain we need at least a trustworthy resolver that possesses its genuine resource record(s). Popular open resolvers (e.g., `8.8.8.8`, `1.1.1.1`) are often considered as trustworthy sources when it comes to censorship evasion. However, we have shown that the vast majority of public DNS resolvers have been polluted with poisoned resource records (§2.10.2). Therefore, we opt not to use them in this case, especially for obtaining the legitimate resource records of geoblocked domains based in China. The only remaining source that is immune to the GFW's poisoned responses and has a given censored domain's genuine resource record(s) is its authoritative name servers. This information is available in the zone files.

We send DNS queries for 138.7K base censored domains and 41K innocuous domains to their authoritative name servers from our controlled machines located at both sides of the GFW. We then expect to observe both censored and non-censored resolutions at two sides of the GFW as a result of this experiment. More specifically, from our US machine, resolutions for domains whose authoritative name servers are located outside China will not be censored as their queries will not cross the GFW, whereas resolutions for domains whose authoritative name servers are located inside China are expected to be censored. On the contrary, resolutions from our China machine towards authoritative name servers located inside China will not be censored, while those queries sent to authoritative name servers outside China will.

2.11.2 Evaluation

To evaluate the effectiveness of our method, we apply the proposed circumvention strategy to filter out poisoned responses for those censored resolutions and retain their "legitimate" responses, which we then compare with actual legitimate responses returned from non-censored resolutions conducted at the other side of the GFW. We find that our circumvention strategy is highly effective, with an accuracy rate of 99.8%. That is, 99.8% of responses classified as "legitimate" match the actual legitimate responses obtained from non-censored resolutions. From a total of 1,007,002,451 resolutions that the GFW poisons, 1,005,444,476 responses classified as "legitimate" by our strategy contain the same resource records (i.e., same IPs, CNAMEs, or IPs under the same AS for domains hosted on Content Delivery Networks) with those observed from non-censored resolutions. As discussed in §2.9.3, there are a small number of cases that we could not classify due to the invisibility of those rarely injected forged IPs in the long tail that GFWatch did not observe. This finding highlights the importance of having an up-to-date and continuous view into the pool of forged IPs for effectively circumventing the GFW's DNS censorship.

To further assist in future adoptions of our strategy so that it will not significantly downgrade the normal performance of other UDP-based DNS resolutions for non-censored domains, we analyze the hold-on duration, which the client should wait *only* when resolving a censored domain, instead of holding on for every resolution.

Figure 2.32 shows the cumulative distribution of the delta time between the first forged response and the legitimate one. The (red) dash line is the CDF of the delta time measured at our China machine, and the (blue) solid line is the CDF of this delta time measured at our US machine. On the x-axis, a positive value means a poisoned response arrives before the legitimate one. In contrast, a negative value indicates that the legitimate response has arrived ahead of the fake ones.

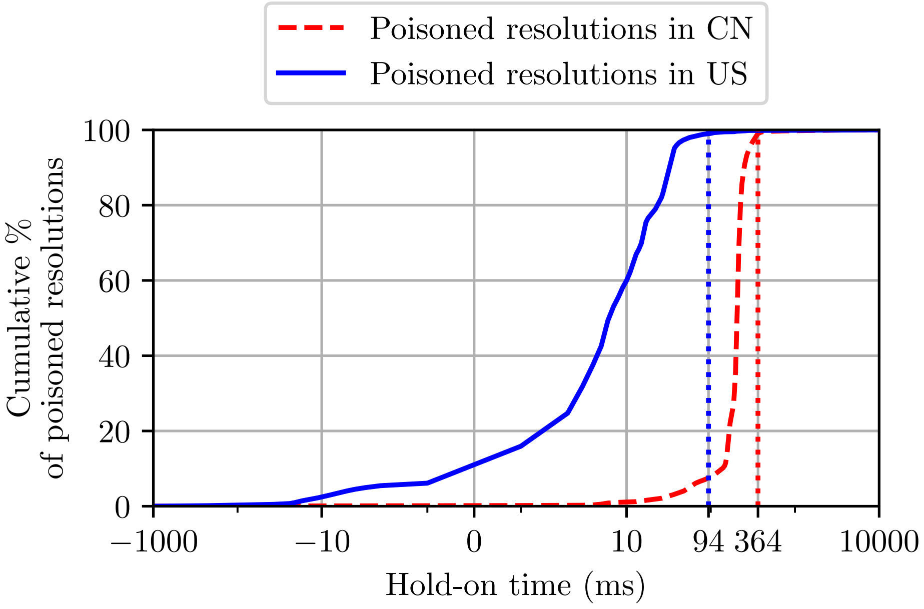


Figure 2.32. CDF of delta time between forged and legitimate responses measured from CN and US controlled machines.

As shown in the figure, the GFW can successfully poison more than 99.9% of all resolutions that carry censored domains, performed from our China machine towards authoritative name servers located outside China. 99% of poisoned responses hit our machine within 364ms ahead of the legitimate ones. Although this delta time may vary, depending on the relative distance between the client and the GFW, for any client whose network location is close to ours, this is the amount of extra time they should wait when resolving a censored domain from inside China. In other words, upon receiving a DNS response after querying a censored domain, the client should wait, at most, an extra 364ms for the legitimate one to arrive. Users at different locations can heuristically probe known censored domains to estimate the hold-on duration that is representative for their location.

From the GFW's perspective, forged responses should ideally arrive at the client before the legitimate one. From our US machine, we find that this is not always true. Due to the unreliable and stateless nature of UDP packets that might get lost or delayed when transferred between two distant locations, and perhaps poisoning users outside China is not the primary design goal of the GFW, 11% of the poisoned responses arrive at our US machine after the legitimate ones. Nevertheless, the remaining 89% of fake responses still hit our machine within 94ms ahead of the legitimate ones. This result again highlights the importance of having a representative dataset of forged IPs used by the GFW to effectively circumvent its DNS censorship. Especially when fake responses arrive later, our dataset of forged IPs is necessary to avoid misclassifying the legitimate ones arriving ahead as "poisoned".

2.11.3 Analysis of True Resource Records

Now that we have successfully obtained the legitimate resource records of the 138.7K base censored domains and 41K innocuously blocked domains, we next analyze them to better understand the impact of blocking these domains. As shown in Table 2.13, 120K (86.8%) base censored domains have either an IPv4, IPv6, or CNAME resource record. In other words, the remaining 18.7K (13.2%) of the base censored domains that currently do not have any resource records, indicating their inactivity. This is also one of the reasons why we observe a large number of domains classified as *"newly observed domain"* and *"not rated"* categories in §2.8.2.

For the innocuously blocked domains, the actual impact of GFW's overblocking may not be as severe because only 25.6K (62.5%) of them have at least one resource record. While the presence of resource records can be a sign of (in)activeness for a given domain, it does not guarantee that a domain is actively hosting any contents or services since a resource record can also be used for redirecting visitors to a domain-parking site. Therefore, the total number of domains with resource records shown in Table 2.13 should be viewed as an upper bound of the actual number of domains that are actively hosting any content or service. As part of our future work, we plan to visit all of these domains using their true resource records and further investigate the contents hosted on them.

Table 2.13. Breakdown of true resource records of base censored domains and innocuously blocked domains.

	Base censored domains		Innocuously blocked domains	
# of domains by NS location	≥1 CN NS	Non-CN NS	≥1 CN NS	Non-CN NS
	53.1K (38.3%)	85.6K (61.7%)	8.9K (21.6%)	32.1K (78.4%)
IPv4	29K (21.1%)	69.5K (50%)	6K (14.7%)	17.8K (43.5%)
IPv6	1.3K (1%)	28K (20.2%)	0.1K (0.3%)	2.8K (7%)
CNAMEs	**31K (22.3%)**	3.6K (2.6%)	**2.9K (7.1%)**	0.5K (1.3%)
# of domains with RR(s)	120K (86.8%)		25.6K (62.5%)	

Another focal point of Table 2.13 is the significantly high number of CNAME resource records of both base censored domains and innocuously blocked domains that have at least one authoritative name server located in China, compared to domains whose authoritative name servers are located outside China. As far as we are aware, this is because of a common workaround that is widely suggested and used by domain owners who want to serve their websites to users at both sides of the GFW since these CNAMEs are not filtered by the GFW.

2.12 Discussion

In this section, we discuss the limitations of our study and provide suggestions for involving parties that are impacted by the GFW's DNS censorship.

2.12.1 Limitations

In order to compare our analysis on the categories of censored domains with prior studies, we choose to use a common classification service provided by FortiGuard [6]. However, we discovered that the GFW's overblocking and geoblocking policy could have already impacted this service (§2.10.2). Moreover, Vallina et al. [238] have shown that different classification services could result in different views of the domains being categorized. We thus tried to obtain additional classification services from two other vendors, namely, McAfee and VirusTotal. However, we were told by McAfee [3] that they only provide the service for business customers, and VirusTotal [5] did not respond to our requests.

Similar to other studies in remote censorship measurement [205, 230, 240], packets sent from our measurement infrastructure may get blocked or discriminated by the GFW. However, over the course of more than nine months operating GFWatch, we did not experience any disruptions caused by such discriminative behaviors, as is evident by the consistency observed between the data collected by GFWatch and across different network locations (§2.9.2). Moreover, as part of our outreach activities, we have also received confirmations from local Chinese advocacy groups and owners of censored domains detected by GFWatch when reaching out to these entities to share our findings. Nonetheless, if our measurement machines ever gets blocked, we can always dynamically change their network location.

Finally, we develop GFWatch as a measurement system to expose the GFW's blocking behavior based on DNS censorship. However, this is not the only filtering technique used by the GFW; censorship can also happen at other layers of the network stack, as previously studied [69, 88, 96, 114, 193, 253, 266]. Although prior works have shown that some websites could be unblocked if the actual IP(s) of censored domains can be obtained properly [62, 133], securing DNS resolutions alone may not be enough in some cases because blocking can also happen at the application layer (e.g., SNI-based blocking [62], keyword-based filtering [207]) or even at the IP layer [129, 132], regardless of potential collateral damage [135].

Nonetheless, DNS is one of the most critical protocols on the Internet since almost every online communication starts with a DNS lookup. We believe that continuously monitoring the GFW's filtering policy at this layer is necessary and important to timely inform the public of the erratic changes in China's information controls policies, both from technical and political perspectives. §2.12.1 provides some examples of domains censored due to political motivations.

Politically Motivated Censorship

Internet censorship and large-scale network outages are often politically motivated [116, 215]. From the censored domains discovered by GFWatch, we find numerous governmental websites censored by the GFW, including many sites belonging to the US government, such as `share.america.gov`, `cecc.gov`, and `uscirf.gov`.

During the nine-month measurement period, GFWatch has also spotted several blockages that coincide with political events. For instance, soon after the clash between China and India due to the border dispute in Ladakh [227], on June 18th 2020 GFWatch detected the DNS filtering of several Indian news sites (e.g., `thewire.in`, `newsr.in`). We reached out to the editor of the Wire India to report blockage against their website by the GFW and were told that they were unaware of the blockage since the site was still accessible from China earlier. Another instance is the blockage of `scratch.mit.edu` that took place in August, 2020, due to some content deemed as anti-China hosted on this website, affecting about three million Chinese users [204]. Although this event was reported by the GreatFire project [121] on the 20th and by Chinese users on the 14th [204], GFWatch actually detected the first DNS poisoning instance earlier on August 13th.

These cases highlight the importance of GFWatch's ability to operate in an automated and continuous fashion to obtain a constantly updated view of the GFW to timely inform the public about changes in its blocking policy.

2.12.2 Suggestions

GFW operators. Although the widespread impact of the GFW's DNS filtering policy is clear, as shown throughout this paper, we are not entirely certain whether this censorship policy is intentional or accidental. While prior works have shown intermittent failures of the GFW [96, 183], all geoblocking of China-based domains and overblocking of innocuous domains discovered by GFWatch have lasted over several months. This relatively long enough period of time leads us to believe that the GFW's operators would have clearly known about the global impact of their DNS filtering policy. By exposing these negative impacts on several parties outside China to the public, we hope to send a meaningful message to the GFW's operators so that they can revise their DNS filtering policy to reduce its negative impacts beyond China's borders.

Public DNS resolvers. Poisoned DNS responses have widely polluted all popular public DNS resolvers outside China due to the geoblocking and overblocking of many domains based in China (§2.10). DNSSEC [92] has been introduced to assure the integrity and authenticity of DNS responses for more than two decades to address these problems. However, DNSSEC is not widely adopted because of compatibility problems and technical complications [66, 75, 123]. To this end, public DNS resolvers can use the strategy introduced in §2.11 to prevent poisoned DNS responses spoofed by the GFW from tainting their cache. By waiting for all responses to arrive and comparing the answers with the pool of forged IPs discovered by GFWatch (§2.9), public DNS resolvers can filter out 99% of poisoned responses by the GFW. Note that it is not always necessary to wait for all responses to arrive because the GFW does not censor all domains. As we will make both censored domains and forged IPs publicly available and update them on a daily basis, these datasets can be used to decide whether to wait or not when resolving a given domain. This way, public DNS resolvers would be able to prevent poisoned responses from polluting their cache, assuring the quality of their DNS service while avoiding any downgrades of normal performance when resolving domains that are not censored.

Owners of forged IPs. Legitimate owners of forged IPs may try to avoid hosting critical services on these IPs as their resources may be saturated due to handling unsolicited TCP and HTTP(s) requests, as shown in §2.10.1. Currently, we do not find evidence that the GFW is using these forged IPs as a way to saturate computing resources of the infrastructure behind them since there are more than 1.7K forged IPs in the pool (§2.9.1) and most of them are dynamically injected (§2.9.3). However, a previous report of the Great Cannon [173] has shown that China is willing to weaponize the global Internet to mount resource exhaustion attacks on specific targets. With DNS censorship, the GFW can adjust its injection pattern to concentrate on a handful of forged IPs, resulting in a large amount of requests towards these targeted IPs and thus saturating their computing resources [71, 120, 143].

Domain owners. Using our dataset of censored domains, domain owners can check whether their domain is censored or not, and censored due to intended blocking or overblocking. Unless the GFW's operators revise their blocking rules, future domain owners should try to refrain from registering domains that end with any overblocking patterns discovered in §2.8.1 to avoid them being inadvertently blocked by the GFW.

End users. Despite the large number of censored domains discovered by GFWatch, different Internet users may be interested in different subsets of these censored domains, but not all. As an immediate countermeasure to the GFW's DNS censorship, we will make the legitimate resource records of censored domains obtained in §2.11 publicly available on a daily basis. This way, impacted users can look up and store legitimate resource records for particular censored domains in their system's `hosts` file to bypass the GFW's DNS censorship. Alternatively, a censorship-circumvention component of software can implement the hold-on strategy (§2.11) and gather records based on the client's location. In case the client cannot access the sanitized data published by GFWatch, another client-side strategy is to send two back-to-back queries. Depending on whether a censored domain belongs to the dynamic or static injection groups (§2.9.3), the client can discern which responses are legitimate. Since the majority of censored domains are poisoned with dynamic IPs, the client can classify the legitimate responses, which typically point to the same IP (due to back-to-back queries) or the same AS. This way, the software only needs to know whether its intended domains are poisoned with static or dynamic IPs. To this end, continuous access to GFWatch's data is not necessary for this strategy to work, while fresh records can still be obtained.

2.13 Conclusion

In this work, we develop GFWatch, a large-scale longitudinal measurement platform, to provide a constantly updated view of the GFW's DNS-based blocking behavior and its impact on the global Internet. Over a nine-month period, GFWatch has tested 534M domains and discovered 311K censored domains.

We find that the GFW's DNS censorship has a widespread negative impact on the global Internet, especially the domain name ecosystem. GFWatch has detected more than 77K censored domains whose poisoned resource records have polluted many popular public DNS resolvers, including Google and Cloudflare. Based on insights gained from the data collected by GFWatch, we then propose strategies to effectively detect poisoned responses and evade the GFW's DNS censorship.

As GFWatch continues to operate, our data will not only cast new light on technical observations, but also timely inform the public about changes in the GFW's blocking policy and assist other detection and circumvention efforts.

CHAPTER 3

NETWORK INTERFERENCE - TRAFFIC DIFFERENTIATION

3.1 Background

Net Neutrality is the notion that ISPs should treat all network traffic equally. Traffic differentiation is an example of a net neutrality violation where certain classes of Internet traffic are given better (or worse) performance. ISPs leverage various techniques to manage their network traffic. One of these techniques is bandwidth throttling which artificially limits the bandwidth that can be effectively used by a subscriber or content providers. When this technique is used towards specific applications, it is often considered as a net neutrality violation. ISPs commonly enforce traffic differentiation by deep packet inspection (DPI) middleboxes that are deployed with the purpose of implementing network management tasks such as bandwidth management, protecting users from malicious traffic, performance optimization, and zero-rating.

A well-known technique to detect traffic differentiation is “Record and replay” where network traffic generated by an application is recorded and later replayed between a device and a control server. A series of back-to-back replay pairs are run in which the original replay exposes the original payload to the middleboxes that use deep packet inspection. While the control replay contains the same traffic patterns but the original payload is obscured to evade detection by DPI devices that often rely on keyword matching in their classification.

3.2 Related Work

Traffic differentiation detection Traffic differentiation has been the target of study for over a decade. Originally, BitTorrent was studied by the Glasnost project [85] which manually crafted measurements to simulate BitTorrent and BitTorrent-like packet exchanges, followed by comparing the throughput distributions of exchanges with and without BitTorrent payloads. NetPolice [261] takes a different approach: detecting differentiation in backbone ISPs by analyzing packet loss behavior of several protocols (HTTP, BitTorrent, SMTP, *etc.*). Bonafide [39] is designed to detect differentiation and traffic shaping in the mobile ecosystem, but still relies on manually crafted files to specify protocols to test, supporting six application protocols. DiffProbe [147] focuses on Skype and Vonage, and detects differentiation by comparing latency and packet loss between exposed and control traffic. The Packsen [252] framework uses several statistical methods for detecting differentiation and inferring shaper details. NANO [232] uses passive measurement from users to infer the existence of traffic differentiation.

A limitation of prior work is that they did not generalize beyond a few tested applications, often used simulated traffic instead of traffic generated by real applications, and did not work from mobile devices. However, recent work [163, 178] showed that deployed differentiation policies often target specific applications based on keyword-based deep packet inspection and thus are often not triggered by simulated traffic. Chkdiff [210, 211] and Molavi Kakhki et al. [178] use application-generated traffic, but are not evaluated at scale.

Identifying rate limiting Recent projects focus on identifying rate limiting of Internet traffic via shaping and policing. The ShaperProbe [148] project detects traffic shaping using end-to-end active probing with synthetic traffic, and it identified suspected shaping in multiple ISPs; however, it is not deployable on mobile devices and does not identify specific applications affected by shaping. Flach et al. [104]

quantify traffic policing for YouTube and its impact on video-quality metrics, but this analysis does not generalize to other video providers and requires access to a content provider's servers. Our approach identifies rate limiting for multiple applications without requiring access to content providers' servers.

3.3 A Large-Scale Analysis of Traffic Differentiation Practices

Net neutrality has driven public debate and discussion over the past decade. While jurisdictions in the European Union have laws forbidding violations of net neutrality, the US enacted similar policies in the middle of 2015 only to roll them back in June, 2018. Despite this interest in net neutrality, users, regulators, and legislators have lacked easy-to-use tools to independently audit network providers, and there is no recent study of the current state of net neutrality violations globally.

In this chapter, we use Wehe, an app developed based on a methodology we designed before for detecting traffic differentiation over mobile networks, to conduct such a global, independent audit of network policies impacting net neutrality. We conduct a large-scale study of deployed differentiation policies using 1,045,413 measurements conducted by 126,249 users across 2,735 ISPs and in 183 countries/regions. We identified differentiation in both cellular and WiFi networks, and analyzed 25 ISPs that have differentiation policies deployed and the observed differentiation behaviors. We also investigate the impact of throttling on video content providers and identify how these policies lead to lower quality streaming despite sufficient resources available in the network to improve it.

3.4 Introduction

Net neutrality, or the notion that Internet service providers (ISPs) should give all network traffic equal service[1], has driven active discussions, laws [9], and policies [101]. However, to date there have been few empirical studies of ISPs' traffic management policies that violate net neutrality principles, or their impact on stakeholders such as consumers, content providers, regulators, and legislators. In this work, we fill this gap via a large-scale study of a common form of net neutrality violations: content-based traffic differentiation that limits throughput for specific applications.

A large-scale study of net neutrality violations and their implications is long overdue, given that the most recent large-scale audits of net neutrality came a decade ago and focused on either backbone networks [261] or a single protocol (BitTorrent) [85]. In the intervening decade, the Internet has evolved in two key ways that require a new approach to auditing. First, today's dominant source of Internet traffic is video streaming from content providers, not BitTorrent. Second, users increasingly access the Internet from their mobile devices, often with a spectrum-constrained cellular connection. There is a need to conduct a study of net neutrality violations that takes these changes into account.

We address this need using 1,045,413 measurements conducted by 126,249 users of our app, across 2,735 ISPs in 183 countries/ regions. From this set of raw measurements, we identify 144 ISPs with sufficient tests to confidently identify differentiation. Wehe builds on prior work for detecting traffic differentiation over mobile networks [178], however, while prior work focused on detecting differentiation on a per-device basis, we leverage our large-scale crowd-sourced data to develop more robust differentiation detection techniques. We then apply these techniques to conduct the largest-scale study of content-based differentiation practices to date.

[1] With a notable exception being *reasonable network management*.

The main contributions of this paper are the methods to detect throttling using data from a large user base, analysis of this data, and findings related to detecting fixed-rate throttling and their impact on affected apps . Beyond technical contributions, our findings have been used by a European national telecom regulator, the US FTC and FCC, US Senators, and numerous US state legislators. To complement this study and to help consumers and regulators make more informed decisions, we maintain a public website with updated analysis and data [13]. This website also contains an extended version of this paper with appendices that provide additional details of observed throttling. We now summarize our technical contributions.

Gathering a large dataset of content-based differentiation practices (§3.5) We perform the largest data collection of content-based differentiation practices, comprising more than 1,000,000 tests, which we continue to maintain on an ongoing basis. We adapted prior work [178] to enable such data collection at scale.

Method for reliably detecting fixed-rate throttling from crowdsourced measurements (§3.6) Individual crowdsourced tests are subject to confounding factors such as transient periods of poor network performance. To address this, we develop a method that reliably identifies fixed-rate throttling by leveraging tests from multiple users in the same ISP. We combine Kolmogorov–Smirnov tests, kernel density estimators, and change point detection to identify cases of fixed-rate throttling and delayed throttling. We evaluated the methodology (§3.7) with controlled lab experiments from the 4 largest US cellular ISPs and found the results of using our methodology on crowdsourced data are consistent with lab experiments.

Characterizing differentiation affecting Wehe tests (§3.8) We conduct a multi-dimensional study of deployed differentiation policies measured by Wehe. We find different network providers using different rate limits (e.g., 1.5 Mbps and 4 Mbps) and targeting a different set of apps (e.g., YouTube vs. Netflix). We also find throttling practices that are poorly disclosed, falsely denied (by one ISP), and that change during the course of our study. Importantly, selective throttling policies potentially give advantages to certain content providers but not others, with implications for fair competition among content providers in throttled networks.

Characterizing video streaming implications of throttling (§3.9) We study how throttling in the US impacts video streaming resolution. We study the video resolutions selected by popular video streaming apps that are affected by throttling, and find examples where throttling limits video quality. We also find many cases where video players self-limit video resolution by default, in some cases selecting a lower resolution than throttling allows. Finally, we observe that streaming sessions experience retransmission rates up to 23%, leading to significant wasted network bandwidth that can be addressed through more efficient throttling implementations.

3.5 Data Collection

We now describe the data collected by the Wehe apps (available from the Google Play and iOS App Stores), which detect content-based differentiation between the device and a server under our control. Wehe is available to download from the Google Play and iOS App Stores.

3.5.1 Methodology

Record and replay To test for differentiation, Wehe uses the "record and replay" technique introduced by Molavi Kakhki et al. [178]. We first record the network traffic generated by an application (e.g., streaming a video using the YouTube app), and include this traffic trace in the app. When a user runs a test, Wehe then replays this traffic between the device and an Wehe server. *We emphasize that our tests* do not *contact content providers' servers.* Thus, all network traffic exchanged between the Wehe app and server are identical to what was recorded, with the exception of different IP addresses.

Wehe runs a series of back-to-back replay pairs. In each back-to-back pair, the *original* replay contains the same payloads as recorded (e.g., YouTube traffic). This exposes the original payload to network devices such as those that use deep packet inspection (DPI). The other replay in the back-to-back pair is the *control* replay, which contains the same traffic patterns (packet sizes, timings) but the original payload is obscured to evade detection by DPI devices that often rely on keyword matching in their classification [163, 164]. For the control replay, Wehe inverts the original payload bits, a technique that our prior work [164] found to evade DPI detection. Note that we do not use random bytes because they were found to trigger differentiation in ways that inverted bits do not [164].

Apps tested by Wehe For this study, Wehe uses traces recorded from YouTube, Netflix, Amazon Prime Video, NBC Sports, Vimeo, Spotify, and Skype. We selected the first five apps because video streaming is a common target of traffic differentiation [146, 163]. We include Spotify because some cellular plans indicate rate limits on streaming audio, and Skype because a telephony app may compete with cellular providers' voice services. The traces in Wehe consist of video streaming from the video apps, music streaming on Spotify, and a video call on Skype. Note that the traces are recorded by the Wehe team, and contain no information about the users running the tests. We use the following symbols to represent each app test: ▶ for YouTube, N for Netflix, a for Amazon Prime Video, 🦚 for NBCSports, S for Skype, ● for Spotify and v for Vimeo.

When running Wehe, users can select which apps to test, and a test consists of up to two replay pairs. The Skype test uses UDP, while the others use TCP. Among TCP tests, NBCSports and Spotify use HTTP, and the others use HTTPS. Thus our approach supports both plaintext and encrypted flows. For the tests that use HTTPS, we simply replay the exact same encrypted bytes over TCP connections between the Wehe app and a Wehe server. Note that since Wehe simply replays the trace as it was recorded, Wehe does not incorporate any dynamic behavior (e.g., adaptive bitrate streaming) that the recorded app might incorporate.

We support UDP traffic in our tests, but do not currently use QUIC traces. An open important research challenge is how to emulate QUIC congestion control, given that we cannot trivially distinguish new payload bytes from retransmissions due to header encryption.

Detecting differentiation for each test After replaying traces for an app, Wehe checks for differentiation and displays the result to the user. Wehe uses a Kolmogorov–Smirnov (KS) test [145] to compare the throughput distributions of the original and the control replays of a given application trace. Wehe samples throughput using fixed time intervals, based on the recorded trace duration: if the replay takes t seconds when recorded, each interval is $t/100$ seconds. Because our record and replay approach sends data no faster than it was recorded, we are guaranteed to have at least 100 samples for each test. However, if the test occurs in an environment where there is a bandwidth bottleneck, the replay can take more than t seconds. If so, we continue to sample at the same rate after t seconds, and thus would record more than 100 samples. Similar to Net Police [261], Wehe conducts Jackknife non-parametric resampling to test the validity of the KS statistic. Wehe indicates to the user that there is differentiation only if *both* the KS test is statistically significant (i.e., p-value less than 0.05, and the resampled KS tests lead to the same result 95% of the time) and the difference in average throughputs is significant (i.e., at least a 10% difference in average throughput) [178].

3.5.2 Implementation

Prior work detected differentiation using packet captures recorded at the replay server [178], assuming that packets received at the server (e.g., TCP ACK packets) came directly from the client. However, we found empirically that this is not the case, largely due to transparent TCP proxies that split the end-to-end connection into two TCP connections. In this case, the server cannot observe rate limits imposed only on the client–proxy connection. To address this, Wehe records traces both from the server side and from the client via periodic throughput measurements collected at the application layer (obtaining raw packet traces would require users to root their phones, which we wish to avoid). We use both traces to identify differentiation and the direction that is affected.

Prior work found that three back-to-back tests yielded low false positive and negative rates for differentiation detection [178]. However, anecdotal reports from Wehe users indicated that the time required to run these tests (16 minutes to test all apps) was a limiting factor in using Wehe. To mitigate this issue, Wehe first analyzes the result of one pair of back-to-back tests for an app. If there is no differentiation detected, then Wehe does not run additional tests for the app. If there is differentiation detected, Wehe runs an additional pair of back-to-back tests and reports differentiation to the user only if it is detected in both tests. The use of only one or two tests might cause higher error rates in results reported to individual app users. In §3.6 we analyze data from *all* tests of the same app in the same ISP across our user base to gain additional statistical confidence in our results.

3.5.3 Confounding factors and limitations

Wehe accounts for the following confounding factors when reporting results to users. First, bandwidth volatility (e.g., due to poor signal strength, cross traffic, etc.) could cause Wehe to incorrectly identify differentiation. To reduce the impact of this, Wehe performs multiple back-to-back tests and reports differentiation to users only when at least two pairs of tests indicate differentiation. This conservative approach may result in false negatives, where Wehe does not report differentiation to the user.

In the next section, we discuss how we aggregate data across our user base to mitigate false negatives and positives due to volatility.

Second, the network may retain history such that one replay test impacts the treatment of the next replay. We instituted random ordering of original and bit-inverted replays, and found no evidence of history affecting our results.

Third, Wehe is subject to the same limitations prior work [178]: it cannot detect differentiation based on IP addresses, peering arrangements, interconnection congestion, traffic volume, or other factors independent of IP payloads. Detecting differentiation based on IP addresses, peering arrangements, and interconnection congestion would seem to require access to content servers (and/ or their IPs)—Wehe alone cannot detect such cases because the paths our measurements follow are potentially different than the ones between clients and content servers.

Though outside the scope of this work, Wehe can be augmented to detect differentiation based on traffic volumes. Specifically, our tests preserve the recorded application's content stream in terms of packet timings and packet sizes, and could trigger differentiation based on those properties. However, both the inverted and original payloads could trigger the same behavior, so we would need to add a second control test (that does not look like any app's traffic volumes) to identify differentiation. Similarly, Wehe could incorporate tests using the real apps under test, in addition to our controlled ones using Wehe, to detect differentiation based on factors other than payload contents. We consider such approaches to be interesting areas for future work.

Last, there is no known API to determine a user's data plan or any differentiation policies on the plan, so we cannot compare Wehe findings with stated policies.

3.5.4 Ethics

Our work involves human subjects, and we took care to follow community best practices when conducting our work. Wehe collects anonymized data from user devices as part of an IRB-approved study. First, as described below, we collect only data that we deemed necessary to characterize differentiation and assess confounding factors. Second, when Wehe is opened by the user for the first time—and before any data is collected—users undergo informed consent via an IRB-approved consent form that specifies the data collected and how it is used. Once users consent, they can initiate tests; if the user does not consent, the app closes immediately. Third, data collection occurs only when users initiate tests, and users can opt out of data collection (and request deletion of data) at any time. Our data-collection and management process has been deemed GDPR-compliant.

3.5.5 Dataset

The data generated by Wehe tests includes throughput samples, as well as the following for each back-to-back test: (1) the server timestamp at the beginning of the test, (2) the first three octets (/24) of the client's IP address, (3) the client's mobile carrier as reported by the operating system, (4) the client's operating system and phone model, (5) the network connection type (WiFi or cellular), and (6) the coarse-grained GPS location (collected with user permission). We describe the reason for collecting each of these items below.

The timestamp allows us to identify trends over time. The carrier name allows us to identify the cellular provider for tests on cellular networks. The client's anonymized IP address information and network type allow us to identify the ISP being tested for WiFi connections[2], and to identify whether there are subnet-level differences in detected differentiation.

[2] Using the "OrgName" field from whois queries to regional Internet registries.

The coarse-grained GPS location (10 km precision) allows us to identify regional differences in ISPs' policies (e.g., in response to state-level net neutrality regulations in the US). The Wehe app first requests the geolocation of the user via the operating system's precise GPS location feature, the Wehe server then geo-codes the geolocation (i.e., looking up the city/state/country) and stores only the truncated geolocation (i.e., with 10 km precision). Users can choose not to share their GPS locations without limiting app functionality. In 15% of tests, the users opted out of location sharing.

The OS and phone model allow us to distinguish whether ISPs discriminate against these factors, or to what extent OSes and phone models might bias the results.

Summary of dataset We summarize our dataset in Table 3.1. Between Jan. 18, 2018 and Jan. 24, 2019, 59,326 iOS users and 66,923 Android users installed Wehe and ran at least one test.

In total, Wehe conducted 1,045,413 tests. We plot the distribution of tests over time in Figure 3.2 (note the log scale on the *y*-axis). We observe a peak of 77,000 tests on January 19, 2018, when a news article raised awareness of the app [12]. There were three other press events that raised awareness of the app; we still observe several hundred tests per day. Wehe users come from at least 183 countries based on geolocation.

Like any crowdsourced dataset, ours is subject to several biases that may impact the generality of our findings. We cannot control when, where, or why users run our tests, and thus we do not have uniform or complete coverage of any ISP or app tested. Figure 3.1 shows the distribution of test locations, where the intensity of the color for each country reflects the number of tests completed in the country. More than 60% of our tests come from the US, most likely due to the recent changes in net neutrality rules combined with US-centric press articles. The phone models used in our tests skew toward higher-end devices, Table 3.2 shows the top phone models and OSes for

Table 3.1. Overview of Wehe data analyzed in this paper.

Replay	Users (%)	Cellular Tests	WiFi Tests
YouTube	106,813 (85%)	97,009	149,850
Netflix	83,369 (66%)	66,320	112,473
Amazon	77,212 (61%)	61,851	102,529
Spotify	65,644 (52%)	43,306	90,963
Skype	60,658 (48%)	37,589	72,250
Vimeo	49,701 (39%)	33,538	67,333
NBC Sports	49,605 (39%)	38,701	71,701
Total	126,249	378,314	667,099

Table 3.2. Summary of Wehe users' phone models. There is a bias toward newer phones and OSes, with devices capable of displaying content in HD.

	iOS		**Android**	
Users		59,326		66,923
Top five OS versions	iOS 11.2.2	15%	Android 7.0	17%
	iOS 11.2.5	7%	Android 8.0.0	9%
	iOS 12.1	5%	Android 8.1.0	. 8%
	iOS 11.4.1	4%	Android 7.1.1	5%
	iOS 11.2.6	3%	Android 6.0.1	4%
Top five phone models	iPhone X	19%	Pixel 2 XL	2.2%
	iPhone 7	14%	Samsung S8	1.9%
	iPhone 6s	12%	Pixel XL	1.8%
	iPhone 7 Plus	11%	Samsung S8+	1.8%
	iPhone 6	7%	Pixel	1.7%

users in the Wehe dataset. A large fraction of our US tests come from large cellular providers, meaning lower-cost providers (e.g., MVNOs) are under-represented.

Despite these biases, our analysis covers 2,735 ISPs[3] in 183 countries, and identifies differentiation in 30 ISPs in 7 countries. We believe this to be the largest study of content-based differentiation practices.

3.6 Detecting Differentiation

We now describe our methodology for identifying and characterizing differentiation using aggregate data collected from multiple users and tests. Specifically, we focus on how we detect fixed-rate bandwidth limits, which we refer to as *throttling*. This is by far the most common type of differentiation that we observed, and the rest of the paper focuses exclusively on fixed-rate throttling.

[3] We noticed that some ISPs used multiple "OrgNames" (e.g., Bouygues and BouyguesTelecom); thus, some ISPs may be counted multiple times.

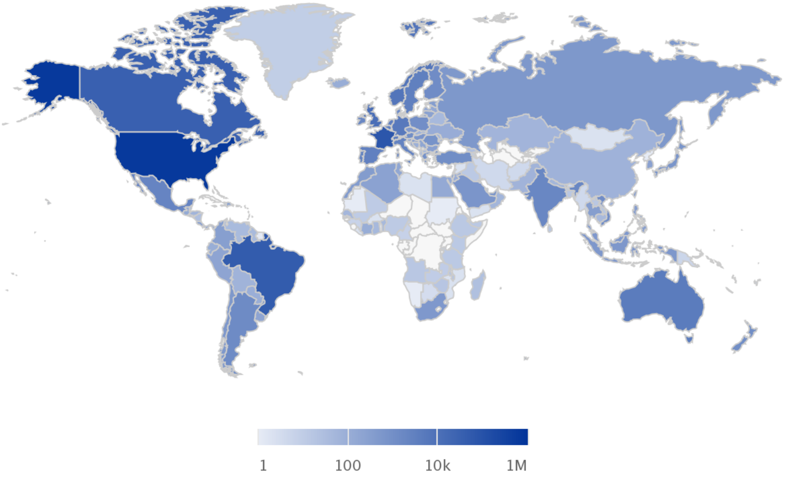


Figure 3.1. Number of tests per country (log scale). Note that 15% of our tests do not have GPS data (e.g., if the user did not provide permission to collect GPS locations), and we excluded them from any geolocation-based analysis.

Our approach relies on the following steps. Similar to prior work, we use the KS test statistic to detect differentiation by comparing throughput distributions for a collection of original replays to those from control replays [178] (§3.6.1) . For replays where differentiation is detected, we detect one or more throttling rates using kernel density estimation (KDE), under the assumption that throughput samples from clients throttled at the same rate will cluster around this value (§3.6.2).

Using this approach to detect throttling rates works well if an entire replay is throttled; however, we find in practice that certain devices enforce fixed-rate throttling only after a burst of packets pass unthrottled, as previously reported by Flach et al [104]. We use change point detection on throughput timeseries data to identify delayed throttling periods (e.g., if they are based on time or number of bytes) and omit unthrottled samples when determining the throttling rate (§3.6.3).

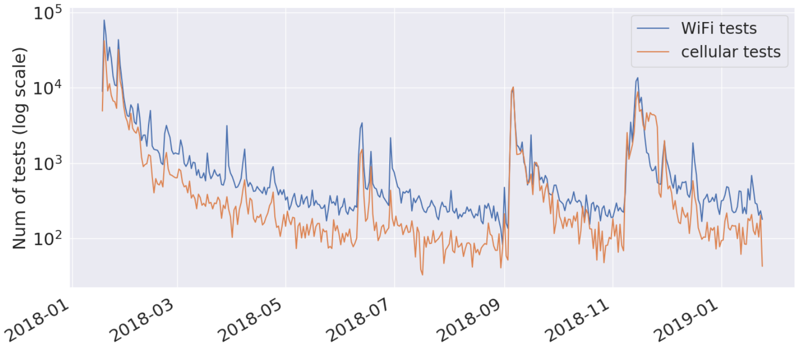


Figure 3.2. Number of Wehe tests per day (log scale).

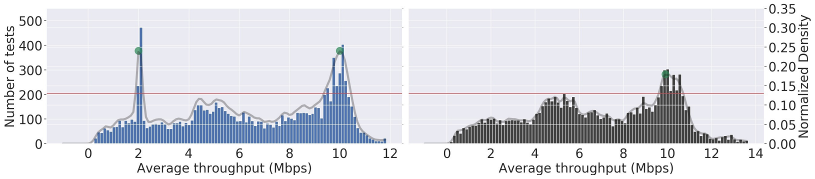


Figure 3.3. Identification of throttling rate. The x-axis is the average throughput, and the y-axes are a histogram of tests (bars) and probability density function (PDF, gray curve) of average throughputs for all YouTube original replays (left) and all YouTube bit-inverted replays (right) from all tests in Sprint network. The horizontal line is the density threshold for detecting potential throttling rates, with green dots are the values above the threshold. We remove values that appear in both original and bit-inverted, leaving 2.0 Mbps as the detected throttling rate.

3.6.1 Identifying differentiation

When identifying differentiation using crowdsourced data, we group tests according to the ISP and the app being tested (e.g., YouTube, Netflix, etc.), which we refer to as an ISP-app pair. We use *all* tests for a given ISP-app pair, where each test consists of one original replay and one bit-inverted replay regardless of whether throttling was detected individually. We focus on ISPs with enough tests to apply the detection methodology; namely, we conservatively require 100 total tests or 10 tests where Wehe identified differentiation.[4] In total, 144 ISPs meet the criteria.

[4] These threshold were picked because they avoided false positives for detecting differentiation.

Our null hypothesis is that there is no differentiation for an ISP–app pair. If this is the case, the distribution of throughput samples observed for original and bit-inverted replays should be similar. To test this, we form two distributions: $\boldsymbol{O}$ is the collection of all throughput samples for all original replays for the ISP-app pair and $\boldsymbol{I}$ is the collection of all throughput samples for all bit-inverted replays for the ISP-app pair. Note that the number of samples in $\boldsymbol{O}$ and $\boldsymbol{I}$ are identical by construction (we include only complete pairs of back-to-back replays).

We then test whether $\boldsymbol{O}$ and $\boldsymbol{I}$ are drawn from different distributions by using the Jackknife re-sampling KS Test described earlier. Specifically, we reject the null hypothesis if the KS-Test indicates different distributions with a $\boldsymbol{p}$-value is 0.05 or less, and the random subsamples of the distribution yield the same result 95% or more of the time.

By aggregating large numbers of tests, we can mitigate the impact of confounding factors such as (random) network dynamics, which should affect both distributions roughly equally given the large number of samples we examine. If we detect differentiation for an ISP-app pair, we next determine whether there is fixed-rate throttling for the pair.

3.6.2 Inferring throttling rates

The technique we use to detect fixed-rate throttling for an ISP–app pair is based on the hypothesis that when an ISP deploys content-specific fixed-rate throttling, this policy affects multiple users (e.g., those with the same data plan). If this occurs, we expect that multiple tests would be throttled in the same way, and thus the distribution of average throughputs for these tests would be centered at the throttling rate instead of being randomly distributed across the range of available bandwidth for a network.

To detect when average throughputs group around a given rate, we use kernel density estimation (KDE), which estimates the probability density function (PDF) of random variables (in our case, throughput). The intuition behind using KDE is that if the random variable (throughput) contains many samples at or near a certain value, the value should have a probability density that is relatively large. Thus, fixed-rate throttling should lead to relatively large probability densities at or near the throttling rate when using KDE. Note that KDE analysis may yield a PDF that has multiple local maxima, meaning the approach can be used to detect multiple throttling rates (or access technology limits).

There are two key challenges for using KDE effectively to identify fixed-rate throttling. First, we must determine what thresholds to use for identifying local maxima in the PDF that correspond to fixed-rate throttling. Second, we must eliminate confounding factors such as rate limits that are not based on the content of network traffic.

Setting thresholds for detection For the first challenge, we use the following heuristic. We assume that at least some fraction f of the total throughput averages, n, for an ISP-app pair are at the throttling rate, and f represents our detection threshold (i.e., we can detect fixed rate throttling affecting at least $f * n$ tests). We then use an *approximation* that the remaining (i.e., unthrottled) samples are randomly distributed across the available bandwidth for the ISP.[5] Finally, we generate data according to this model, run KDE (using a Gaussian kernel with a bandwidth of 0.1), determine the density for the f throttled samples and use that as our detection threshold t.

More specifically, for each ISP–app pair we find the number of replays n and the average throughput range $[x, y]$. We then construct a distribution consisting of $(1 - f) * n$ data points with values uniformly distributed between x and y, and $f * n$

[5] This is not true in practice, but serves as a useful first-order approximation to identify throughput values of interest.

data points with the value $(y-x)/2$. We run KDE on this distribution, and set our detection threshold t to the density value at $(y-x)/2$ (containing a fraction f of the values). We evaluated the methodology with f=0.02 in §3.7, and we found no false positives or negatives.

Eliminating confounding factors The heuristic above identifies characteristic throughput values containing more samples than would be expected from a uniformly random distribution; however, not all such values are due to fixed-rate throttling. For example, an ISP may impose rate limits on *all* traffic for a device (e.g., due to usage or access-technology limits). Importantly, such behavior should impact *both* the original replays and the bit-inverted replays.

To eliminate such cases, we first remove from consideration any average throughput values that have high density in *both* the original and bit-inverted distributions. Next, we include only throughput values with high density and that correspond to throttling rates observed by Wehe tests that indicated differentiation to the user. For this, we run the same KDE analysis described above, but only on tests where the Wehe app identified differentiation.

As an example of this approach, the left plot in Figure 3.3 shows a histogram of average throughput values for all YouTube *original replays* over Sprint, and the estimated PDF (grey curve) from running KDE. The horizontal line indicates our detection threshold, t, which identifies high-density values near 2 Mbps and 10 Mbps. The right figure plots the same, but for the *bit-inverted replays*; note that both original and bit-inverted distributions have above-threshold density values at 10 Mbps, indicating that this throughput value is not due to content-based differentiation. Finally, we confirm that tests where the Wehe app indicated differentiation exhibited throttling at 2 Mbps using KDE analysis, and conclude that 2 Mbps is the throttling rate for this ISP-app pair.

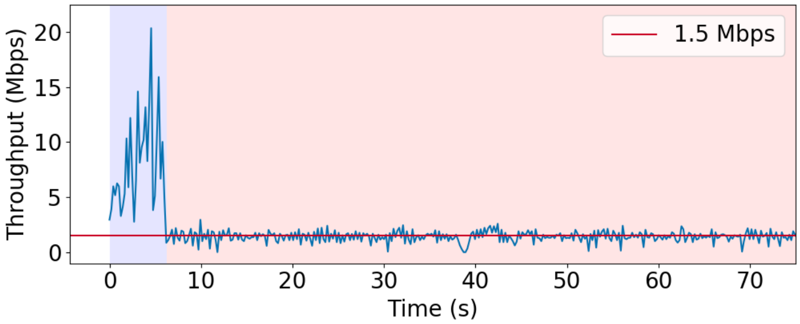


Figure 3.4. Throughput over time for a Netflix test over T-Mobile, showing delayed throttling. Note that the first few seconds of the transfer include rates up to 20 Mbps, after which they drop to 1.5 Mbps (horizontal line).

3.6.3 Accounting for delayed throttling

The methods described so far in this section assume that if fixed-rate throttling occurs, it affects the entirety of a Wehe test experiencing throttling. In the case of T-Mobile, we found empirically that this assumption was violated because they engage in *delayed throttling*, previously reported by Flach et al. [104]. Figure 3.4 shows a timeseries of throughput for a Netflix replay that is subject to this policy: initially the transfer achieves throughput up to 20 Mbps; afterward, the transfer drop to 1.5 Mbps (horizontal line).

Previous work found that delayed throttling was implemented by limiting the number of bytes that are unthrottled, and identified the behavior using the number of bytes that are transferred before the first packet is dropped [104]. In our work, we seek to avoid assumptions about whether such delayed throttling is based on bytes or time, and to use techniques that are insensitive to packet drops caused by reasons other than delayed throttling. Instead, we assume that a detectable delayed throttling session will have at least one phase change, and that all tests for an ISP-app pair affected by delayed throttling will experience the same delay (i.e., number of seconds or bytes). Thus, to detect delayed throttling for an ISP-app pair, we use change point detection (to identify the phase change) and KDE to identify whether the change occurs after a number of seconds or bytes.

Our null hypothesis is that there is no delayed throttling. If this were true, a phase change could be caused by reasons such as bandwidth volatility, and we would expect that the delay would be randomly distributed. To test this hypothesis, we investigate only tests for an ISP-app pair with exactly one phase change, and determine the distribution of delays.

To detect phase changes, we use the PELT algorithm [153] and filter out any tests that do not have exactly one change point. We tuned the detection algorithm so that it would detect change points from tests where we replayed Netflix on T-Mobile's network using our lab devices. To determine whether the change point indicates statistically significant throughput on either side of the boundary, we use a KS test to compare the distributions of throughput before and after the change point. If they are different, we add the change point time and bytes to the list of change points for the ISP-app pair.

After gathering lists of change points, we use KDE[6] to determine whether the change points for the ISP-app pair are randomly distributed or instead cluster together around a time or number of bytes. If there is a relatively large density value at a given number of bytes or time, then we reject the null hypothesis and flag the ISP-app pair as experiencing delayed throttling, according to bytes or time, whichever has the largest density value. As an example, Figure 3.5 shows the distribution and estimated PDF of delayed throttling *bytes* for Netflix on T-Mobile, where most of the change points are detected around 7 MB.[7]

If delayed throttling is detected, we filter out throughput samples during the delay and detect the throttling rate as described in the previous section.

[6] With an empirically derived threshold density of 0.1.

[7] The change point times have substantially lower density.

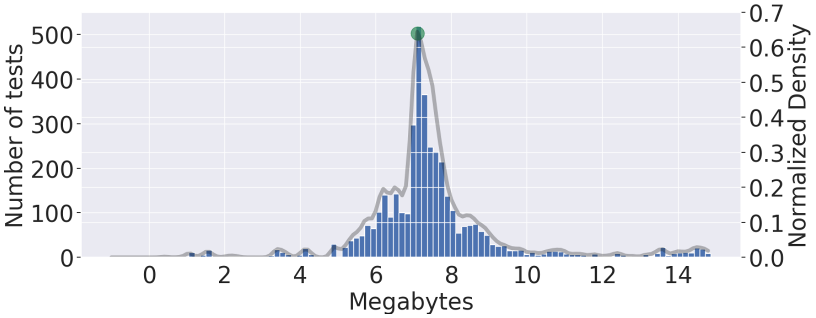


Figure 3.5. Detecting delayed throttling bytes for Netflix in T-Mobile. For each change point (in bytes) on the *x*-axis, the figure shows a histogram and estimated PDF generated from KDE. The green dot (at 7 MB) indicates the detected number bytes before throttling begins.

3.6.4 Limitations and Caveats

The methodology for detecting fixed-rate throttling presented in this paper is subject to the following limitations.

Record/replay limitations The recorded traffic that we use for an app in Wehe's replay tests may not always match the traffic generated by the app. For example, if a video provider switches from HTTP to HTTPS, our tests would be out of date until we create a new recording. Likewise, a throttling device may update its rules for detecting traffic before we deploy a new recording, and this could lead to false negatives. We periodically check for changes to apps that we test in Wehe, e.g., we updated our recordings in mid-January, 2019 after Amazon Prime Video changed from using HTTPS to HTTP.

Detection limits We can find evidence of fixed-rate throttling only when we have sufficient tests (and a sufficient fraction of tests being throttled to the same rate) from an ISP to obtain statistical significance. We detected differentiation for 39 ISPs, but we see no evidence of fixed-rate throttling for 9 of them. Specifically, for these 9 cases we found differences between original and bit-inverted average throughputs, but we did not detect fixed-rate throttling after running KDE. We do not know the root causes for these cases.

3.7 Evaluation of Detection Method

We now evaluate our detection method using controlled experiments from the four largest US cellular providers. Ideally, we would compare our detection results with ground-truth information from each ISP in our study, but gaining access to each network in our crowdsourced data would be infeasible. Further, even if we had this information, we could not control for confounding factors such as varying network conditions, the user's data plan or usage history.

Instead, we validate that our detection methodology produces findings that are *consistent* with controlled experiments performed in our lab. For the largest four US carriers, we do find consistent results—our lab tests indicate content-based differentiation and fixed-rate throttling that matches results produced by our analysis of data from Wehe users.

3.7.1 Lab experiment setup

We purchased SIM cards from AT&T, Sprint, T-Mobile and Verizon. We intentionally purchased prepaid plans that mention indicators of throttling practices, such as "video streaming at 480p" or "video optimized streaming." Note that none of the disclosures indicated which video providers are targeted for throttling, nor how the targeting is done. We conducted lab experiments in Jan. 2018, May 2018 and Jan. 2019 for AT&T, T-Mobile and Verizon, and the tests for Sprint only in January, 2019 due to difficulty acquiring a prepaid SIM.

For each experiment, we ran each of the 7 Wehe tests on each SIM card 10 times. We include two sets of tests for Vimeo (with two different domains) and Amazon Prime Video (one using HTTPS and one using HTTP) in Jan. 2019 to reflect the change in how the service delivered video that month.

Since the data plan disclosures did not indicate which video services were throttled, we do not have ground truth for which Wehe tests should be affected. Instead, our

Table 3.3. Comparison of crowdsourced and lab results on four US mobile carriers in Jan. 2018, May 2018, and Jan. 2019. There is an additional column (') for new Amazon and Vimeo traces recorded in Jan. 2019. A shaded box indicate throttling was detected using crowdsourced data; otherwise throttling was not detected. A ✓ indicates that result from crowdsourced data matched the lab experiment results, and "-" means we do not have a lab experiment.

					S				v				
	01/18	05/18	01/19	01/18	05/18	01/19	01/18	05/18	01/19	01/19′	01/18	05/18	01/19
AT&T	▨ ✓	▨ ✓	▨ ✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Verizon	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
T-Mobile	▨ ✓	▨ ✓	▨ ✓	✓	✓	✓	✓	✓	▨ ✓	✓	✓	✓	✓
Sprint	-	✓	✓	▨ -	-	✓	-	-	✓	✓	-	-	✓

		▶			N				a	
	01/18	05/18	01/19	01/18	05/18	01/19	01/18	05/18	01/19	01/19′
AT&T	▨ ✓	▨ ✓	▨ ✓	▨ ✓	▨ ✓	▨ ✓	✓	✓	✓	▨ ✓
Verizon	▨ ✓	▨ ✓	▨ ✓	▨ ✓	▨ ✓	▨ ✓	▨ ✓	▨ ✓	▨ ✓	▨ ✓
T-Mobile	▨ ✓	▨ ✓	▨ ✓	▨ ✓	▨ ✓	▨ ✓	▨ ✓	▨ ✓	▨ ✓	▨ ✓
Sprint	▨ -	-	▨ ✓	▨ -	-	▨ ✓	▨ -	-	▨ ✓	▨ ✓

hypothesis is that if our lab tests are affected by content-based differentiation, then we should be able to detect exactly which content triggers throttling. We use the "binary randomization" method [163] for identifying content that triggers DPI classification rules used in throttling deployments.

3.7.2 Comparison with Wehe data

To compare the lab findings with crowdsourced Wehe data, we build subsets of Wehe data, one each from Jan., 2018 and May 2018, and two from Jan. 2019 to reflect updated recordings released that month. We then use the methodology from the previous section to detect fixed rate-throttling and compare our findings with those from lab experiments.

Table 3.3 presents a summary of findings, showing that our lab tests and crowdsourced data are consistent. There are at least three columns for each ISP-app pair, representing tests from Jan., 2018, May 2018 and Jan., 2019. There is an additional column for Amazon and Vimeo where we separate out the tests based on whether they were done using older (the third column) or newer traces (the fourth column). A shaded cell indicates that our method detected differentiation using crowdsourced tests for that ISP-app pair from that specific month, while a white cell means that we did not. A ✓shows that the result from Wehe data matches the lab experiment for an ISP-app pair during that month, and a "-" indicates cases where we have no lab experiments (January/May 2018 for Sprint).

Table 3.3 shows that all cases of throttling in lab experiments were also detected in Wehe tests. We could not verify consistency for all Wehe crowdsourced findings; namely, our tests indicate throttling of Skype video in the first nine months of 2018, but we did not have a Sprint SIM for lab tests then.

3.8 Characterizing differentiation

We now present our findings from all Wehe tests in our dataset. In this section, we focus on cases where throttling is detected for at least one ISP-app pair. Table 3.4 summarizes the results. While the majority of *tests* come from WiFi networks, the majority of *detected differentiation* occurs in cellular networks. We discuss our findings in more detail below.

3.8.1 Identified differentiation

We identified 30 ISPs in 7 countries that throttle at least one Wehe test. Nearly all cases of detected throttling affect video streaming services, with YouTube being throttled the most often (25 cases), and Vimeo being throttled the least (3 cases).

Our methodology did not detect any ISP throttling of Spotify tests in our data, and detected throttling of Skype video tests only in Sprint (§3.8.3), Boost Mobile (which is owned by Sprint), and the United Arab Emirates (UAE) on both WiFi and cellular connection. In the UAE, the "throttling" is to zero (i.e., Skype tests are blocked), reportedly because Skype provides an unlicensed VoIP service in the country.

The most common detected throttling rate is 1.5 Mbps (12 cases). These rates typically correspond to ISPs that disclose data plans offering low-resolution video streaming, a topic we investigate in §3.9. Besides blocking, the lowest throttling rate detected is 0.5 Mbps from Boost Mobile, and the highest detected throttling rate is 6 Mbps from Start Communications, a regional ISP based in Ontario, Canada.

Throttling via WiFi In the vast majority of ISPs tested via WiFi, our methodology did not detect throttling. The exceptions were the UAE blocking Skype (1 instance), and five other providers in North America. At least three of these five (Viasat, Hughes and NextLink) are satellite providers, and are likely more bandwidth constrained. While we cannot confirm the type of network for VianetTV and Start Communications, Vianet's website indicates that they offer residential plans that carry Internet traffic over cellular connections. Thus, the majority of detected throttling over WiFi occurred in networks that carry traffic over long-range wireless networks.

Throttling over cellular connections Most cellular throttling comes from providers in the US. We detected differentiation in *nearly every* major US cellular ISP, and we found all these throttling practices started before June 2018 (i.e., when the FCC rolled back restrictions on throttling [10]). While the number of detected cases in the US might be due to the bias in our dataset, it is in part due to the regulatory regime. For example, we do not detect throttling from cellular ISPs in France, where we have a large sample size, and where the practice is illegal. One notable exception is Google Fi, which did not throttle any of our tests.

3.8.2 Variations in detected throttling

Not all tests for each ISP-app pair in Table 3.4 are throttled. We now investigate several potential root causes for the behavior.

Policy changes over time One explanation for non-uniform throttling of Wehe tests is that throttling policies changed for ISP-app pairs during our study. We test this by comparing the detected throttling rates and fraction of throttled tests over time. The number of tests per ISP-app pair varies considerably over time, so we use

Table 3.4. ISPs where we detect differentiation, the throttling rates, apps affected, and the number of Wehe tests in the ISP.

Country	ISP	Throttled Apps	Rate(s)	Tests
WiFi network				
Canada	Start Comms.	a, 🦚	6 Mbps	126
Canada	ViaNetTV	▶, N	1 Mbps	45
UAE	Etisalat	S	0 Mbps	23
US	Hughes Net. Sys.	▶, N	1 Mbps	81
US	NextLink	▶, N, a, 🦚, v	4 Mbps	72
US	ViaSat	▶, N	1 Mbps	112
Cellular network				
Canada	Rogers	▶, N	1.5 Mbps	4479
Canada	SaskTel	N	1.5/3 Mbps	61
Chile	Entel	▶	1.5 Mbps	30
Germany	Telekom DE	▶, a	1.5 Mbps	178
Israel	HOTmobile	▶	1.5 Mbps	23
UAE	Etisalat	S	0 Mbps	73
UAE	du	S	0 Mbps	44
US	AT&T	▶, N, 🦚	1.5 Mbps	46,013
US	BoostMobile	▶, N, a, S	0.5/2 Mbps	792
US	Cellcom	▶, N, a	4 Mbps	97
US	Cricket	▶, a	1.5 Mbps	1,224
US	CSpire	▶, N	1 Mbps	41
US	FamilyMobile	▶, N, a, 🦚	1.5 Mbps	106
US	GCI	▶, N, a, 🦚, v	1/2 Mbps	153
US	Iowa/iWireless	▶, N, a, 🦚	1.5/3 Mbps	76
US	MetroPCS	▶, N, a, 🦚	1.5 Mbps	2,135
US	Sprint	▶, N, a, S	2 Mbps	35,295
US	T-Mobile	▶, N, a, 🦚, v	1.5 Mbps (delayed)	39,820
US	Tracfone Wireless	▶, a	2 Mbps	410
US	Verizon	▶, N, a	2/4 Mbps	69,016
US	Visible	▶, N	2 Mbps	52
US	XfinityMobile	▶, N, a	2 Mbps	131
UK	giffgaff	▶, N	1 Mbps	58
UK	O2	▶, N	1.5 Mbps	210

the following approach to test sample sizes with sufficient power to draw conclusions about policy changes. For each ISP-app pair, we divide tests into periods, each with a minimum of 100 total tests and 10 throttled tests. If the same rate(s) and a similar fraction of throttled tests are observed in all periods, we conclude there is no policy change over time. We consider only two periods that meet this criteria: periods of one month, or periods of 6 months. The latter divides data into periods before and after the US rolled back net neutrality protections in June, 2018.

The monthly analysis covers AT&T, Sprint, T-Mobile, Verizon, and MetroPCS, and the biannual analysis covers BoostMobile, cricket, O2, and Tracfone Wireless. For the vast majority of cases, we see the same throttling rate and similar fractions of throttled tests during the study period, indicating that most policies are stable over the course of one year, and that the throttling policies were in place even before new FCC rules permitted them in June, 2018.

For T-Mobile, we detected that Vimeo tests are throttled only after Nov., 2018, and a small fraction of YouTube tests were throttled at 2 Mbps (instead of 1. 5 Mbps) only in Jan. 2019. For Sprint, we found that Skype video tests ceased to be throttled after Oct., 2018. We detected no changes in other policies.

Time-of-day We now investigate the role time-of-day plays in throttling, in light of claims that throttling is necessary to prevent overloading the network during times of heavy usage [14]. If this were true, we expect to see higher incidence of throttling during a cellular network's busy hours (e.g., 8am to midnight) compared to overnight (e.g., midnight to 8am). We test the hypothesis by grouping tests into day and night using busy hours identified in prior work [244], and checking whether the fractions of the throttled tests are different (e.g., more tests are being throttled during the day).

Specifically, for each ISP-app pair we denote the fraction of throttled tests as f, number of total daytime tests D (d of which are throttled), and N total nighttime tests (n of which are throttled). If there is no time-of-day effect, the number of throttled tests should be $D * f$ during the day and $N * f$ during the night. We run a chi-squared test comparing the actual number of throttled tests (n and d) with the expected number of throttled tests ($D{*}f$ and $N{*}f$); if the p-value is less than 0.05, we conclude there is a time-of-day effect.

Out of the 77 ISP-app pairs we detected throttling, 71 of them include tests both during the day and night; of these we find no evidence of time-of-day effect for 60 cases. Of the remaining 11, four have fewer than 30 tests and thus limited statistical power, and we manually investigated the remaining 7 ISP-app pairs. For these cases,

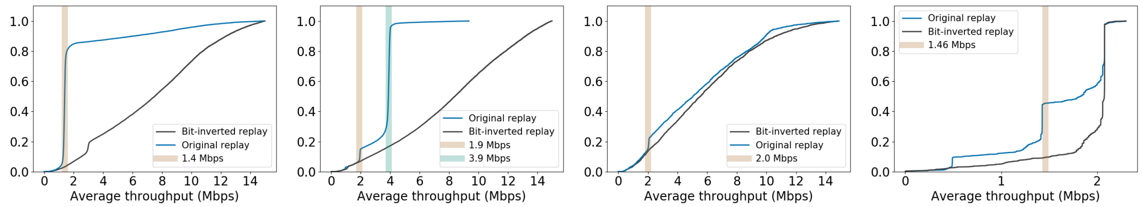
we found the opposite result as our hypothesis: the fraction of tests throttled during busy periods decreased compared to non-busy periods. This could be due to a different set of users with different throttling policies, or it could be due to a lack of sufficient available bandwidth to detect throttling. Our data supports the latter explanation, because during busy hours we see a larger fraction of original and bit-inverted tests with throughput lower than the throttling rate.

Geographical differences We investigated whether there are geographic differences in throttling practices, e.g., one region is affected more than another. This could be due to factors such as state-level net neutrality laws or a regional deployment of throttling (e.g., affecting a subset of a provider's packet gateways). We focus on the US-based ISPs (where we have the most samples), and conduct a state-level regional analysis. Our finding is that there are differences in throttling experienced by Wehe users in each state, but these variations are not persistent and are consistent with random noise.

Mobile OSes We consider whether the mobile OS plays a role in whether a given client will be throttled or not. We analyzed the fraction of tests for an ISP-app pair affected by differentiation for iOS and Android, and found that the top four US cellular providers have similar throttling rates for both mobile OSes.

IP prefixes We next consider whether throttling affects only certain IP prefixes assigned to clients. We first grouped the tests according to the routable IP prefix that the client's IP address belongs to, then determined the fraction of throttled tests for each prefix. If differentiation is implemented on a per-prefix basis we would expect a bimodal trend with prefixes having either no cases of throttling, or nearly all tests experiencing throttling. However, this is not what we observe.

For each ISP-app pair, we calculated the fractions of throttled tests for each IP prefix, and then checked the standard deviation of the fractions, if the distribution is bimodal,we would expect a high standard deviation. In more than 87% of the cases, we observe a standard deviation of less than 0.2; we manually checked the remaining cases and did not see a bimodal trend.



(a) YouTube tests on AT&T (all data)

(b) YouTube tests on Verizon (all data)

(c) YouTube tests on Sprint (January 2018)

(d) Skype tests on Sprint (January 2018)

Figure 3.6. CDF of average throughputs from YouTube ((a)–(c)) and Skype (d) tests. For AT&T, the detected throttling rate is 1.5 Mbps in 3.6(a), for Verizon there are two detected rates (2 Mbps and 4 Mbps) in 3.6(b). In Sprint, we detect throttling of a small portion (4.8%) of the original replays throttled to 2 Mbps in 3.6(c). For Skype tests on Sprint in 3.6(d), the detected throttling rate is 1.5 Mbps.

Other possible explanations Variations in throttling could be due to ISPs offering different service plans and features, only some of which include throttling. When visiting websites for several ISPs where we detected throttling, we found options to subscribe to (more expensive) plans that did not limit video streaming (e.g., ViaSat's Platinum 100 plan), and/or features to disable throttling (e.g., AT&T's Stream Saver). Because our dataset does not include plan information, we cannot quantify the impact of these factors.

3.8.3 Case studies

We now present several notable findings using our methodology on crowdsourced Wehe data from the top four US carriers. A common case is presented in Figure 3.6(a), depicting CDF of average throughput for original and bit-inverted YouTube replays. The vast majority of original samples cluster around 1.5 Mbps (the detected throttling rate) while the bit-inverted replays are (mostly) evenly distributed across the throughput range.

Multiple throttling rates for the same ISP-app pair KDE analysis revealed that Verizon has two throttling rates, one at 4 Mbps (the majority of throttled tests) and the other at 2 Mbps. We show this using a CDF of average throughputs in

Figure 3.6(b). We believe this is due to different plans offered by Verizon; e.g., in Dec. 2018 their "go unlimited" plan included "DVD-Quality Streaming (480p)" while their "beyond unlimited" plan allowed "HD-Quality Streaming (720p)" [11].

Small fraction of tests affected by throttling Our methodology identifies throttling in Sprint, despite a small percent (4.8%) of tests being affected. To demonstrate this visually, we plot a CDF of average throughput samples for original and bit-inverted replays in Figure 3.6(c). There is an inflection point at 2 Mbps, the detected throttling rate, which we also detected in lab experiments using a prepaid SIM. We suspect the reason for such a small fraction of tests being affected is that throttling happens on uncommon data plans, such as the prepaid one we used for lab experiments.

Different policies for different video-streaming apps As discussed in §3.6.3, T-Mobile implements delayed throttling based on bytes. Interestingly, we find that not all throttled video streaming services are treated equally under this policy. We detected 7 MB of delayed throttling for Netflix and NBC Sports, and 6 MB for Amazon Prime. YouTube does not get any delayed throttling in our dataset—they are throttled from the start.

Skype tests in Sprint We did not detect throttling of Skype video calls in our lab experiments on a Sprint SIM; however, our methodology found evidence of Skype video throttling from Wehe crowdsourced data. Figure 3.6(d) shows a CDF of average throughputs for original and bit-inverted Skype video tests, with detected fixed-rate throttling at 1.5 Mbps. When focusing on Jan., 2018 data, we find that the Jackknife KS test used to detect differentiation has a *p*-value of $8 * 10^{-94}$ with 100% accept ratio—strong evidence of throttling.

Interestingly, Wehe tests identified such differentiation until Sep. 2018, but the tests no longer indicated differentiation afterward. One explanation for this behavior is that Sprint no longer throttles based on content in our Skype video tests. When asked (in Oct., 2018) to comment on our findings regarding Skype and other tests indicating throttling, a press spokesperson from Sprint replied: “Sprint does not single out Skype or any individual content provider in this way”. Our lab tests in 2019 corroborate the claim about Skype (but does not speak to the early 2018 findings); however, our lab tests also identify that Sprint *does* single out content providers via content-based throttling.

3.9 Impact on Video Streaming

While Wehe enables us to observe differentiation at scale, it does not provide details about the video resolution for a given app (e.g., which video resolutions are selected by a video streaming app). As described in Section 3.5, Wehe simply replays the payloads recorded from video streaming, and does not adapt bitrates dynamically. To address this, we need additional experiments to help us understand how streaming apps (with adaptive bitrate) behave when being throttled.

This section describes how we conduct these additional measurements by instrumenting video streaming apps to determine how throttling impacts the video resolution selected by each player. We focus on this metric because it is the one most often cited in ISPs’ throttling disclosures (e.g., “video streams up to 480p”), but to date has received little attention from auditing measurements. We first describe the data collected in Section 3.9.1. We then discuss the impact of throttling and app’s data usage setting on streaming in Section 3.9.2. Finally, we identify root causes for observed behavior using sequence-time diagrams in Section 3.9.3.

3.9.1 Measuring video resolution

Experiment environment We analyze Netflix, YouTube and Amazon on prepaid plans from AT&T[8], T-Mobile, Verizon, and Sprint between Jan. 14 and Jan. 25,

[8] The AT&T SIM had Stream Saver [8], which throttles video traffic, enabled by default.

2019. We present the impact of throttling on video quality and throughput over each ISP, and compare each result with tests (1) over WiFi, (2) when connected via an encrypted VPN on the same cellular connection, and (3) when disabling any data-saving by the apps (i.e., enabling streaming at the maximum rate according to the app). The WiFi network is not throttled and VPN tunnels evade any content-based throttling. In each network setting, we use each app to stream video for two minutes.[9] We repeat each ISP-app experiment five times and present summary results.

Video streaming In each video streaming session, we stream the same video and let the client app determine what video resolution to use. While the bitrate selection code is unavailable to us, we expect that the video streaming session is influenced by factors such as encoded bitrates, network conditions, access technology, and data usage settings. We discuss the factors that we vary in our tests.

To vary whether a video streaming session is affected by content-based throttling, we stream video with and without an encrypted VPN tunnel. For access technology, we run tests over both cellular and WiFi connections. For the WiFi network, we confirm that neither the network nor the app (the app does not attempt to save data on WiFi) are the bottleneck. Finally, for the data usage settings, we note that Netflix and Amazon provide an option in their app to let users control how much data they want to use over cellular connections. Amazon has three settings: Good (0.6GB/hour), Better (1.8GB/hour) and Best (5.8GB/hour). Netflix also provides three settings Automatic, Save Data and Max data which allow 4 hours, 6 hours and 20 minutes per GB. For these apps we test both their default[10] and most data-intensive settings (which we refer to as "Max data").

[9] We used iPhones (6S and 8) and a Nexus 6 and obtained similar results from all three.

[10] "Automatic" for Netflix and "Good" data usage on Amazon.

Video quality and throughput There are no publicly known APIs to collect video quality information from mobile apps. Our approach is to monitor video quality during playback using app features that print video resolution on the screen, then use optical character recognition (OCR) to extract them for YouTube and Netflix. Amazon does not expose video resolution information, so we obtain this data from the HTTP GET request and the manifest file. We calculate the throughput based on packet capture data.

3.9.2 Impact of throttling

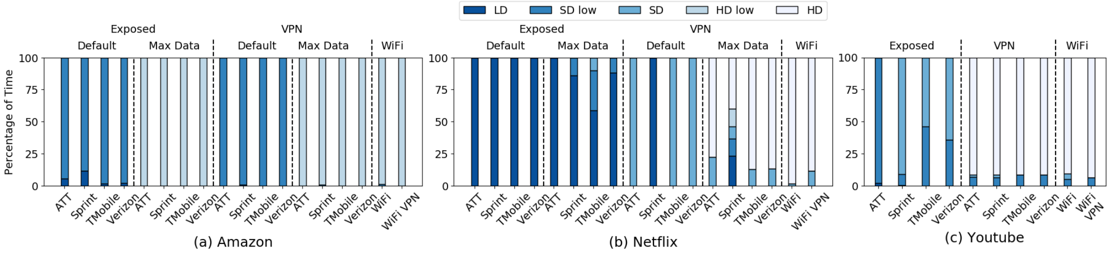
We begin by analyzing the impact of throttling on video streaming resolution and throughput.

Throttling decreases playback resolution Figure 3.7 shows the percentage of time the video is streamed at each resolution. Table 3.5 shows the resolution mapping and the total pixel count of the various resolutions that were streamed on Amazon Prime Video, Netflix and YouTube during our video resolution experiments. Each subfigure plots the results for a different video streaming service, and each plot groups our results according to whether the test exposes the original packet payloads ("exposed") or uses a VPN to conceal them. As expected, in tests where packet payload is exposed for Netflix and YouTube, the playback resolution is lower than the cases where the VPN conceals the payload. This result holds even when we turn off any data saving mechanisms ("Max data" and "VPN Max data"). The exceptions are Amazon and Sprint on Netflix. We discuss the Amazon case below, but do not have a root cause for Sprint/Netflix.

Cellular networks can support higher throughputs Figure 3.8 presents the throughput observed while streaming under the default app settings for exposed (solid) and tunneled traffic (dashed). We confirmed there was sufficient cellular bandwidth (using Speedtest) of at least 20 Mbps in all tests. This shows that cellular net-

Table 3.5. Resolution mapping used in Figure 3.7.

Labels	Resolutions	Total pixel count
LD	384x216	82,944
	426x240	102,240
	512x213	109,056
	480x270	129,600
	652x272	177,344
SD low	608x342	207,936
	710x296	210,160
	640x360	230,400
SD	640x480	307,200
	768x432	331,776
	720x480	345,600
	854x480	409,920
HD low	960x540	518,400
	1152x480	552,960
	1280x533	682,240
HD	1280x720	921,600



(a) Amazon

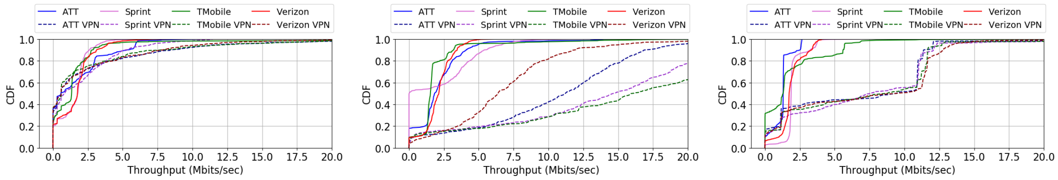
(b) Netflix

(c) Youtube

Figure 3.7. Stacked histogram for each video streaming service, showing the percentage of time the video is streamed at each resolution (LD, SD, and HD are low, standard, and high definition). The precise resolutions are in Table 3.5.

works support much higher throughput than the throttling rate (as indicated by the larger average throughputs for VPN curves of Netflix and YouTube). The exception is Amazon, discussed below.

Apps default to limiting their streaming rates We find that Amazon and Netflix, by default, use a lower video resolution than the network can support, with or without the VPN (Figure 3.7). When compared with "Max data," nearly all of the tests using the default data usage setting select video resolutions that were below 480p (SD), with Netflix picking a resolution as low as 384x216 (LD) and Amazon picking 710x296 ("SD low"). These are substantially lower than the phone screen resolution (1334x750). When we disable the default behavior and allow the apps to stream at



(a) Amazon (b) Netflix (c) YouTube

Figure 3.8. CDF of throughput for each video streaming service (with low data usage settings) and each carrier.

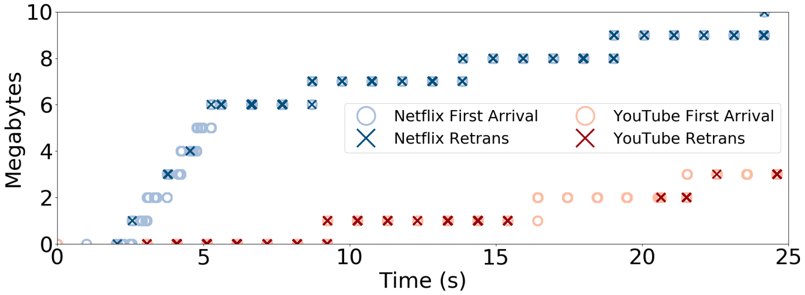


Figure 3.9. Bytes over time when streaming Netflix (red) and YouTube (blue) on T-Mobile. Netflix experiences delayed throttling, but not YouTube.

their highest achievable rate, video streaming services are able to achieve significantly higher resolutions—indicating that, except for Netflix on Sprint, the cellular networks tested have sufficient bandwidth to support HD video.

Amazon over VPN connections is a special case. Unlike others, the throughput does not increase while using a VPN because Amazon's default data usage settings restricts the app to only use 0.6GB per hour, or an average of 1.6 Mbps both with and without the VPN. When we disable the default throughput limitations (not shown) Amazon has throughputs of 2 Mbps when the packet payloads are exposed and throughput of 4.5 Mbps over VPN. Note that the reason Amazon does not appear to be limited by 1.5 Mbps throttling on AT&T is because AT&T throttles each TCP connection to 1.5 Mbps individually, and Amazon uses multiple TCP connections.

In summary, throttling limits maximum video resolution, but apps' default settings and available resolutions also play a significant role.

3.9.3 Transport-layer impact of throttling

We now investigate how throttling impacts video streaming at the transport layer. We explore this impact in Figure 3.9 by considering the bytes transferred over time for each video stream. Each figure is annotated with the initial transmission of a packet (circles) as well as retransmission events (×). We collect packet captures for this analysis from a (non-rooted) iPhone via standard developer tools for iOS [15], and we use the definition of "TCP retransmission" in Wireshark [16].

Transparent proxies and the transport layer We observe AT&T and Verizon implementing TCP terminating proxies in their networks with drastically different results for the transport layer. Though separate analysis with Wehe, we identified that AT&T uses a transparent TCP proxy to split the connection, buffer packets from the server, and pace packets between the proxy and mobile device, at a rate of 1.5 Mbps. This buffering and pacing of packets results in throttling that does not incur high rates of retransmissions.

In contrast to AT&T, the *retransmission rate is 23%* when streaming Netflix on Verizon, the highest among the other carriers and high by any standard.

We conducted additional experiments to investigate the root cause of this behavior. Namely, we used Wehe tests in lab experiments and observed the same high retransmission rates at the client; however, the server traces indicated little-to-no retransmission. Thus, we believe that Verizon implements a transparent TCP proxy like AT&T; however, unlike AT&T, Verizon's proxy does not pace packets, instead sending them faster than the throttling device allows (and thus leading to high packet loss). Interestingly, there is minimal impact of the high retransmission rate on video streaming, likely because the video streaming buffer absorbs any transient disruptions to packet transfers.

Policies can differ between applications Figure 3.9 shows the bytes over time when streaming a video on Netflix and YouTube over T-Mobile's network. Note that when retransmission and first-arrival markers overlap, there are time gaps of 10s of milliseconds, but not visible on a graph (which is measured in 10s of seconds). In each cluster of points, the retransmissions occur first (and correspond to bytes sent one RTO earlier), then as the retransmitted packets are ACKed, new first transmissions occur 10s of milliseconds later.

We observe that T-Mobile throttles Netflix after 7 MB of data transfer (delayed throttling), while it does not delay throttling for YouTube. While packet loss is zero during the delayed throttling period, immediately afterward the retransmission rate is 26%, eventually reducing to 17%. By comparison, YouTube initially experiences a loss rate of 6.8% and drops to 3% after 70 seconds. In both cases, losses waste substantial bandwidth, but the problem is more acute for cases with delayed throttling due to TCP sending at a high rate and adapting slowly.

3.10 Discussion

This section discusses additional considerations about our findings, their generality, and future work.

Bias towards the US. Most of our data comes from the US, which necessarily biases our findings in a way that likely undercounts differentiation outside the US. In addition, Wehe includes tests for video and music streaming apps, as well as VoIP and videoconference apps, that are popular in the US. However, it is likely that other apps are more popular in other countries, and some of those apps may be throttled. If this is the case, we would underreport the prevalence of throttling in such regions. In the future, we will add tests for more apps that are popular in other regions.

ISPs where throttling was not detected. We gathered sufficient samples to detect differentiation in 144 ISPs, and detected differentiation in 30 of them. This suggests that the majority of ISPs that we studied do not deploy content-based differentiation. Examples include major broadband providers in the US (e.g., Comcast), and all broadband and cellular ISPs in France. Note, however, that some ISPs may throttle using methods other than content-based differentiation (e.g., IP address or monthly data usage) that Wehe cannot detect. Thus, we can only say that we did not detect content-based differentiation, but we cannot tell whether other differentiation occurs.

Ground truth. It is difficult, and in some cases impossible, to find ground truth for every ISP in our study. However, we did validate, via documentation on providers' websites, that throttling policies exist for most US carriers and for several outside the US. That said, there are many ISPs that either do not disclose this information or make it hard to find. There is a clear need for better transparency and more uniform ways of disclosing throttling behavior.

Future of DPI-based differentiation. We identified that the classification rules used by ISPs for throttling rely on plaintext payload contents (e.g., SNI field in TLS handshake). In newer protocols such as TLS 1.3 with encrypted SNI (or QUIC with similar features), such information will no longer be in plaintext—begging the question of how DPI devices will identify traffic for differentiation. We believe that content-based differentiation might still exist even when using such protocols, e.g., by correlating flow IPs with the plaintext names in DNS lookups that they correspond to. Of course this can be addressed by technologies like DNS over HTTPS. Assuming all content is encrypted (even DNS), we envision that classifiers will search for traffic patterns instead of text strings. Because Wehe preserves traffic patterns, we believe our approach will still work.

Complex relationships between content providers, ISPs, and throttling practices. We showed that throttling practices are deployed by many ISPs, and these practices generally worsen performance for content providers in terms of packet loss and decreased video quality. However, we cannot identify the extent to which content providers are (dis) satisfied with such policies. For example, content providers may experience reduced transit costs for throttled video when compared to unthrottled video that uses higher resolution and more bandwidth. It is also possible that ISPs and content providers have entered into agreements to collaboratively control traffic volumes from streaming video. In short, the relationship between content providers, ISPs, and deployed traffic management practices may be more complicated than publicly disclosed. Of course, understanding such relationships is outside of the scope of this work.

3.11 Conclusion

In this work, we conducted a large-scale, one-year study of content-based traffic differentiation policies deployed in operational networks. We developed and evaluated a methodology that combines individually error-prone device measurements to form high-confidence, statistically significant inferences of differentiation practices, and identified differentiation in both cellular and WiFi networks. We found that most throttling targets video streaming, and that there are a wide range of throttling implementations detected in our dataset. In addition, we investigated the impact of throttling on video streaming resolution, finding that while throttling does limit video resolution, it is also the case that default settings in video streaming apps in some cases are the primary reason for low resolution. We are making our code, dataset, and summary of findings publicly available to inform stakeholders and bring empirical data to discussions of net neutrality regulations.

CHAPTER 4
DOMAIN NAME ENCRYPTION

4.1 Background

In this section, we provide an overview of the recent proposals that aim to improve the security and privacy of the DNS and TLS protocols.

4.1.1 DNS over HTTPS and DNS over TLS

The DNS protocol exposes all requests and responses in plaintext, allowing a network observer to monitor or modify a user's network traffic, eavesdrop or tamper with it. For instance, a man-on-the-side attacker can send forged DNS responses to redirect a user to malicious websites, while state-level organizations can also inject forged DNS responses to disrupt connections for censorship purposes.

As an attempt to enhance the security and privacy of the DNS protocol, two new DNS standards, namely, DNS over HTTPS (DoH) and DNS over TLS (DoT) were recently proposed. These technologies aim to ensure the integrity, authenticity, and confidentiality of DNS traffic. By using DoH/DoT, all DNS queries and responses are transmitted over TLS, ensuring their integrity against last-mile adversaries who would otherwise be in a position to launch a man-in-the-middle (MITM) or man-on-the-side (MOTS) attack. The main difference between the two standards is that DoT is essentially DNS over TLS, while DoH is DNS over HTTP over TLS. This allows DoH to combine features of HTTP and DNS. One such feature is the option to "push" DNS data from the server to the client.

4.1.2 SSL/TLS and ESNI

During the TLS handshake [84], the two communicating parties exchange messages to acknowledge and verify the other side using digital certificates and agree on various parameters that will be used to create an encrypted channel. In a client-server model, the client trusts a digital certificate presented by the server as long as it has been signed by a trusted certificate authority.

Ideally, private or sensitive information should be transmitted after the TLS handshake has been completed. This goal can be easily achieved when a server hosts only a single domain. However, name-based virtual hosting, which is an increasingly used approach for enabling multiple domains to be hosted on a single server requires a mechanism for the server to know which domain name a user intends to visit before the TLS handshake completes. This mechanism is necessary since the server needs to present the correct certificate to the client. The server name indication (SNI) extension was introduced in 2003 as a solution to this problem. The SNI extension contains a field with the domain name the client wants to visit so that the server can then present the appropriate certificate. Since this step is conducted before the completion of the TLS handshake, the domain name specified in the SNI field is exposed in plaintext. Consequently, all the privacy risks associated with the traditional design of DNS also apply to the SNI extension.

The encrypted server name indication (ESNI) extension [214] has been proposed as part of the TLS 1.3 to resolve the issue of SNI revealing the domain name visited by a user. Clients that use ESNI encrypt the SNI field towards a given server by first obtaining a server-specific public key through a well-known ESNI DNS record. In September 2018, Cloudflare was among the first providers to announce support for ESNI across its network [200].

4.2 Related Work

The domain name system is one of the core elements of the Internet and plays an essential role for most online services. As a result, it has been (ab)used for many different purposes. In this section, we review prior works that investigate DNS from security and privacy perspectives, and some recent studies that analyze the domain name ecosystem via empirical measurements.

From a security perspective, domain names have been heavily abused for illicit purposes. For instance, domain squatting is one of the most common abuses. It is used to register domains that are similar to those owned by well-known Internet companies. Domain squatting has many variations, including typo-squatting [24, 150, 231], homograph-based squatting [110, 203], homophone-based squatting [185], bit-squatting [184], and combo-squatting [155]. Domain names registered using these squatting techniques can then be used for phishing [192, 203] or distributing malware [28]. To cope with these unwanted domain names, DNS data has been used intensively to create domain name reputation systems to detect abuse [34, 35, 36, 160].

Another major form of DNS abuse is DNS poisoning, in which an on-path observer can easily observe and tamper with DNS responses to redirect users to malicious websites or to censor unwanted content [32, 86, 100, 168, 196, 222]. The exposure of domain names in DNS requests and TLS handshakes (due to SNI) has also been extensively used for traffic filtering and censorship [63, 133].

As mentioned in §4.1, the traditional design of DNS exposes Internet users to severe privacy risks. In addition to on-path observers (discussed in §4.3.2), previous works have also studied the privacy risk associated with centralizing all domain name resolutions to third-party recursive resolvers (e.g., 8.8.8.8, 1.1.1.1) [54, 55, 128, 154, 226, 262]. Zhao et al. [262] propose to add random noise and use private information retrieval to improve privacy by obfuscating DNS queries. These proposals however have turned out to be impractical and insecure under certain circumstances [54], and have not been adopted. Lu et al. [170] propose privacy-preserving DNS (PPDNS), which is based on distributed hash tables and computational private information retrieval. More recently, Hoang et al. [134] propose K-resolver as a mechanism to distribute DoH queries among multiple recursors, thus exposing to each recursor only a part of a user's browsing history. Sharing similar goals with our study, Shulman et al. [226] examined the pitfalls of DNS encryption. By analyzing the co-residence of zone files on name servers, the authors argue that guessing visited domains by destination IP address does not provide a significant advantage. Our findings, however, show that this is only the case for a small number of domains that are co-hosted with an adequate number of other domains.

Hounsel et al. [139] study the effect of DoH/DoT on the performance of domain name resolution and content delivery. The study finds that the resolution time of DoH/DoT is longer than traditional DNS resolution. Of the two new technologies, DoT provides better page load times while DoH at best has the same page load times as DNS. They also find that DoT and DoH perform worse than DNS in networks with sub-optimal performance. Similarly, a recent study by Bottger et al. [44] analyzes the DoH ecosystem and shows that they can obtain more advanced privacy features of DoH with marginal performance degradation in terms of page load times.

There have also been studies that investigated the robustness of the DNS ecosystem through various types of measurements. Ramasubramanian et al. [206] leverage a dataset of almost 600K domains to study their trusted computing base, which is the set of name servers on which a FQDN is hosted. The study shows that a typical fully-qualified domain name (FQDN) depends on 46 servers on average. Dell'Amico et al. [83] use DNS data collected through both active and passive measurements to also investigate the ecosystem of dependencies between websites and other Internet services. Similar to our work, Shue et al. [225] use DNS data collected by both passive and active measurements to study web server co-location and shared DNS infrastructure. However, their measurements were conducted from a single location, while excluding all servers belonging to CDNs. Furthermore, the passive DNS dataset used was collected by capturing network traffic from the authors' institute, therefore facing all potential issues of a passive measurement discussed in §4.3.3.1. More recently, Hoang et al. [135] revisit the results of Shue et al. [225] by conducting a large-scale active DNS measurement study, which reveals that the Web is still centralized to a handful of hosting providers, while IP blocklists cause less collateral damage than previously observed regardless of a high level of website co-location.

4.3 Assessing the Privacy Benefits of Domain Name Encryption

As Internet users have become more savvy about the potential for their Internet communication to be observed, the use of network traffic encryption technologies (e.g., HTTPS/TLS) is on the rise. However, even when encryption is enabled, users leak information about the domains they visit via DNS queries and via the Server Name Indication (SNI) extension of TLS. Two recent proposals to ameliorate this issue are DNS over HTTPS/TLS (DoH/DoT) and Encrypted SNI (ESNI). In this paper we aim to assess the privacy benefits of these proposals by considering the relationship between hostnames and IP addresses, the latter of which are still exposed. We perform DNS queries from nine vantage points around the globe to characterize this relationship. We quantify the privacy gain offered by ESNI for different hosting and CDN providers using two different metrics, the k-anonymity degree due to co-hosting and the dynamics of IP address changes. We find that 20% of the domains studied will not gain any privacy benefit since they have a one-to-one mapping between their hostname and IP address. On the other hand, 30% will gain a significant privacy benefit with a k value greater than 100, since these domains are co-hosted with more than 100 other domains. Domains whose visitors' privacy will meaningfully improve are far less popular, while for popular domains the benefit is not significant. Analyzing the dynamics of IP addresses of long-lived domains, we find that only 7.7% of them change their hosting IP addresses on a daily basis. We conclude by discussing potential approaches for website owners and hosting/CDN providers for maximizing the privacy benefits of ESNI.

4.3.1 Introduction

As users become more aware of the importance of protecting their online communication, the adoption of TLS is increasing [158]. It is indicative that almost 200M fully-qualified domain names (FQDNs) support TLS [95], while Let's Encrypt [19] has issued a billion certificates as of February 27, 2020 [18]. Although TLS significantly improves the confidentiality of Internet traffic, on its own it cannot fully protect user privacy, especially when it comes to monitoring the websites a user visits.

Currently, visited domain names are exposed in both i) DNS requests, which remain unencrypted, and ii) the Server Name Indication (SNI) extension [141] during the TLS handshake. As a result, on-path observers can fully monitor the domain names visited by web users through simple eavesdropping of either DNS requests or TLS handshake traffic. Several recent proposals aim to improve the security and privacy of these two protocols. Specifically, DNS over HTTPS (DoH) [137] and DNS over TLS (DoT) [140] aim to preserve the integrity and confidentiality of DNS resolutions against threats "below the recursive," such as DNS poisoning [86], while Encrypted Server Name Indication (ESNI) [214] aims to prevent "nosy" ISPs and other on-path entities from observing the actual visited domain of a given TLS connection.

In this paper, we quantify the potential improvement to user privacy that a full deployment of DoH/DoT and ESNI would achieve in practice, given that destination IP addresses still remain visible to on-path observers. Although it is straightforward to reveal a user's visited site if the destination IP address hosts only that particular domain, when a given destination IP address serves many domains, an adversary will have to "guess" which one is being visited.

We use two properties to quantify the potential privacy benefit of ESNI, assuming the provider of the DoH/DoT server used is fully trusted (as it can still observe all visited domains): the k-anonymity property and the dynamics of hosting IP addresses. Assuming that k different websites are co-hosted on a given IP address (all using HTTPS with ESNI supported), the privacy of a visitor to one of those sites increases as the number of *k-1* other co-hosted sites increases. In addition, the more dynamic the hosting IP address is for a given site, the higher the privacy benefit of its visitors is, as the mapping between domain and hosting IP address becomes less stable, and thus less predictable.

To quantify these two properties, we conducted active DNS measurements to obtain the IP addresses of an average of 7.5M FQDNs per day drawn from lists of popular websites [26, 172] (§4.3.3). To account for sites served from content delivery networks (CDNs) which may direct users differently based on their location, we performed name resolutions from nine locations around the world: Brazil, Germany, India, Japan, New Zealand, Singapore, United Kingdom, United States, and South Africa. Our measurements were conducted in two months to investigate how much a network observer can learn about the domains visited by a user based solely on the IP address information captured from encrypted traffic.

We find that 20% of the domains studied will not benefit at all from ESNI, due to their stable one-to-one mappings between domain name and hosting IP address. For the rest of the domains, only 30% will gain a significant privacy benefit with a k value greater than 100, which means an adversary can correctly guess these domains with a probability lower than 1%. The rest 50% of the domains can still gain some privacy benefits, but at a lower level (i.e., $2 \leq k \leq 100$). While sophisticated website fingerprinting attacks based on several characteristics of network packets (e.g., timing and size [124, 169, 181, 190, 191, 254]) can be used to predict the visited domains, our study aims to provide a lower bound of what an attacker can achieve.

Moreover, we observe that sites hosted by the top-ten hosting providers with the highest privacy value ($k > 500$) are far less popular (§4.4.1.2). These are often less well-known sites hosted on small hosting providers that tend to co-locate many websites on a single IP or server. In contrast, the vast majority of more popular sites would gain a much lower level of privacy. These sites are often hosted by major providers, including Cloudflare ($k = 16$), Amazon ($3 \leq k \leq 5$), Google ($k = 5$), GoDaddy ($k = 4$), and Akamai ($k = 3$).

In addition, we find that frequently changing IP addresses (at least once a day) are limited to only 7.7% of the domains that we were able to resolve each day of our study. As expected, dominant providers in terms of more dynamic IP addresses include major CDN providers, such as Amazon, Akamai, and Cloudflare (§4.4.2.1).

Finally, we validate and compare our main findings by repeating part of our analysis using two different public DNS datasets (§4.4.3), and provide recommendations for both website owners and hosting/CDN providers on how to maximize the privacy benefit offered by the combination of DoH/DoT and ESNI (§4.4.5). In particular, website owners may want to seek hosting services from—the unfortunately quite few—providers that maximize the ratio between co-hosted domains per IP address, and minimize the duration of domain-to-IP mappings. Hosting providers, on the other hand, can hopefully aid in maximizing the privacy benefits of ESNI by increasing the unpredictability of domain-to-IP mappings.

4.3.2 Threat Model

We assume an idealistic future scenario in which both DoH/DoT and ESNI are *fully* deployed on the Internet. Under this assumption, an on-path observer will only be able to rely on the remaining visible information, i.e., destination IP addresses, to infer the sites being visited by the monitored users. The extent to which this inference can be easily made depends on i) whether other domains are hosted on the same IP address, and ii) the stability of the mapping between a given domain and its IP address(es).

The probability with which an adversary can successfully infer the visited domain can be modeled using the k-anonymity property, with k corresponding to the number of domains co-hosted on the same IP address. The probability of a successful guess is inversely proportional to the value of k, i.e., the larger the k, the more difficult it is for the adversary to make a correct guess, thus providing increased user privacy.

The above threat model is oblivious to distinguishable characteristics among a group of co-hosted websites, such as popularity ranking, site sensitivity, and network traffic patterns. We should thus stress that the situation in practice will be *much more favorable* for the adversary. Even for a server with a high k, it is likely that not all k sites will be equally popular or sensitive. Although the popularity and sensitivity can vary from site to site, depending on who, when, and from where is visiting the site [130], an adversary can still consider the popularity and sensitivity of the particular k sites hosted on a given IP address to make a more educated guess about the actual visited site.

Utilizing the ranking information of all domains studied, we model such an adversarial scenario in §4.4.2 and show that our threat model based on k-anonymity is still valid. In addition, page-specific properties such as the number of connections towards different third-party servers and the number of transferred bytes per connection can be used to derive robust web page *fingerprints* [51, 52, 73, 91, 127, 166, 194, 245], which can improve the accuracy of attribution even further. Although identifying a visited website among all possible websites on the Internet by relying solely on fingerprinting is quite challenging, applying the same fingerprinting approach for attributing a given connection (and subsequent associated connections) to one among a set of k *well-known* websites is a vastly easier problem.

Consequently, an on-path observer could improve the probability of correctly inferring the actual visited website by considering the popularity and sensitivity of the co-hosted domains on the visited IP address, perhaps combined with a form of traffic fingerprinting. Although such a more powerful attack is outside the scope of this work, as we show in the rest of the paper, our results already provide a worrisome insight on how effective an even much less sophisticated attribution strategy would be, given the current state of domain co-hosting.

4.3.3 Methodology

In this section we review existing DNS measurement techniques and highlight the data collection goals of our study. We then describe how we select domains and vantage points to achieve these goals.

4.3.3.1 Existing DNS Measurements

Previous studies use passive measurement to observe DNS traffic on their networks [83, 225, 249]. However, passive data collection can suffer from bias depending on the time, location, and demographics of users within the observed network. Passive data collection can also raise ethical concerns, as data collected over a long period of time can gradually reveal online habits of monitored users.

There are also prior works (by both academia and industry) that conducted large-scale active DNS measurements for several purposes and made their datasets available to the community [159, 208]. However, these datasets have two common issues that make them unsuitable to be used directly in our study. First, all DNS queries are resolved from a single location (country), while we aim to observe localized IPs delivered by CDNs to users in different regions. Second, although these datasets have been used in many other studies, none of the prior measurements are designed to filter out poisoned DNS responses (e.g., as a result of censorship leakage), which can significantly affect the accuracy of the results and negatively impact data analysis if not excluded. We discuss steps taken to sanitize these datasets in §4.4.

4.4 Poisoned Response Sanitization

While processing public DNS datasets from other sources (to which we compare our findings in §4.4.3), we surprisingly discovered thousands of low-ranked or obscure domains seemingly being co-hosted on the same IP addresses that also host very popular websites, such as Facebook and Twitter—which of course was not actually the case. As part of our investigation, we observed that the authoritative servers of most of these domains were located in China (using the MaxMind dataset [174]). We then queried the same domains from outside China using their authoritative servers, and indeed received responses pointing to IP addresses that belong to either Facebook or Twitter. By inspecting network traffic captures taken during these name resolutions, we observed that the initial response containing the wrong (falsified) IP address was followed by another DNS response with the same valid DNS query ID that contained a different (correct) IP address.

We attribute the above observed behavior to DNS-based censorship by the "Great Firewall" (GFW) of China [31, 69, 168, 255, 258, 266], which has also been observed and analyzed by previous studies [32, 100]. Censorship leakage happens due to the GFW's filtering design, which inspects and censors both egress and ingress network traffic. While some censors (e.g., Pakistan, Syria, Iran) forge DNS responses with NXDOMAIN [43, 61, 65, 180] or private addresses [37], making them easier to distinguish, China poisons DNS responses with routable public IP addresses belonging to other non-Chinese organizations [100, 133, 258]. In contrast to the findings of previous works, however, in this case the real hosting IP addresses of the censored domains are located within China, while previous works mostly focus on investigating the

Table 4.1. Most frequently abused subnets in poisoned DNS responses from China.

AS32934 **Facebook**	**AS13414** **Twitter**	**AS36351** **SoftLayer**
31.13.72.0	199.59.148.0	74.86.12.0
31.13.69.0	199.59.149.0	67.228.235.0
31.13.73.0	199.16.156.0	74.86.151.0
31.13.66.0	199.59.150.0	75.126.124.0
69.171.245.0	199.16.158.0	67.228.74.0

blockage of websites that are hosted outside China (e.g., google.com, facebook.com, blogger.com).

To validate our findings, we cross-checked the IP addresses from second (real) DNS responses with the ones obtained by resolving the same domains from locations in China. As the authoritative servers of these domains are also in China, our queries did not cross the GFW, which mostly filters traffic at border ASes [32, 72, 257], and thus were not poisoned.

We follow this verification technique, where we issue additional queries to resolve domains whose authoritative name server is located in China and then detect injected DNS packets to exclude poisoned responses from analyses. In total, we detected more than 21K domains based in China with poisoned responses. Table 4.1 shows the top /24 IP subnets belonging to Facebook, Twitter, and SoftLayer, which are the most frequently observed in poisoned responses. Our observation aligns with recent findings of other censorship measurement studies [133, 182].

4.4.0.1 Our Measurement Goals

Ideally, we would like to derive the mapping between all live domain names and their IP addresses. Unfortunately, this is extremely challenging to achieve in practice because there are more than 351.8 million second-level domain names registered across all top-level domains (TLDs) at the time we compose this paper [241], making it unrealistic to actively resolve all of them with adequate frequency. Furthermore, not all domains host web content, while many of them correspond to spam, phishing [192, 203], malware command and control [28], or parking pages registered during the domain dropcatching process [161], which most users do not normally visit.

As we aim to study the privacy benefits of ESNI, we thus choose to focus on active sites that are legitimately visited by the majority of web users. To derive such a manageable subset of sites, we relied on lists of website rankings, but did not consider only the most popular ones, as this would bias our results. Instead, we expanded our selection to include as many sites as possible, so that we can keep our measurements manageable, but at the same time observe a representative subset of *legitimately visited* domains on the Internet.

4.4.0.2 Domains Tested

There are four top lists that are widely used by the research community: Alexa [26], Majestic [172], Umbrella [237], and Quantcast [202]. However, it is challenging to determine which top list should be chosen, as recent works have shown that each top list has its own issues that may significantly affect analysis results if used without some careful considerations [162, 219, 221]. For instance, Alexa is highly fluctuating, with more than 50% of domain names in the list changing every day, while Majestic is more stable but cannot quickly capture sites that suddenly become popular for only a short period of time. Pochat et al. [162] suggest that researchers should combine these four lists to generate a reliable ranking.

For this study, we generated our own list by aggregating domains ranked by Alexa and Majestic from the most recent 30 days for several reasons. First, these two lists use ranking techniques that are more difficult and costly to manipulate [162]. Second, they have the highest number of domains in common among the four. We exclude domains from Quantcast because it would make our observations biased towards popular sites only in the US [162]. Lastly, we do not use domains from Umbrella

Table 4.2. Breakdown of the five largest TLDs studied.

	Daily	**Total**
TLDs	1,031	1,125
FQDNs	7,556,066	13,597,409
.com	3,835,080	7,026,005
.org	347,993	584,924
.de	264,057	501,597
.net	263,262	442,729
.ru	210,701	346,194

because the list is vulnerable to DNS-based manipulation and also contains many domains that do not host web content [162, 219]. To this end, we studied a total of 13.6M domains. As shown in Table 4.2, our derived list comprises an average of 7.5M popular fully qualified domain names (FQDNs) collected on a daily basis, covering 1,031 TLDs. For the whole experiment duration, we studied a total of 13.6M domains and 1,125 TLDs. Table 4.2 also shows the top five largest TLDs in our dataset, with `.com` being the most dominant, comprising more than 50% of the domains observed.

Data scope. Although this subset of domains corresponds to about 4% of all domains in the TLD zone files, we argue that it is still adequate for the goal of our study, i.e., determining whether the current state of website co-location will allow ESNI to provide a meaningful privacy benefit. Considering only this subset of domains may lead to an under-approximation of the actual k-anonymity offered by a given IP or set of IPs, as some co-hosted domains may not be considered. This means that our results can be viewed as a lower bound of the actual k-anonymity degree for a given visited IP address, which is still a desirable outcome.

As discussed in §4.3.2, the popularity of a website, along with other qualitative characteristics, can be used by an adversary to improve attribution. Indeed, given that the long tail of domains that are left out from our dataset mostly correspond to vastly less popular and even unwanted or dormant domains [231], any increase in k they may contribute would in practice be rather insignificant, as (from an attribution perspective) it is unlikely they will be the ones that most web users would actually visit.

4.4.0.3 Measurement Location and Duration

Due to load balancing and content delivery networks, deriving *all* possible IP addresses for a given popular domain is very challenging. To approximate this domain-to-IP mapping, we performed our own active DNS measurements from several vantage points acquired from providers of Virtual Private Servers (VPS). When choosing measurement locations, we tried to distribute our vantage points so that their geographical distances are maximized from each other. This design decision allows us to capture as many localized IP addresses of CDN-hosted sites as possible. To that end, we run our measurements from nine countries, including Brazil, Germany, India, Japan, New Zealand, Singapore, United Kingdom, United States, and South Africa. Our vantage points span the six most populous continents. From all measurement locations mentioned above, we send DNS queries for approximately 7.5M domains on a daily basis. When issuing DNS queries, we enabled the iterative flag in the queries, bypassing local recursive resolvers to make sure that DNS responses are returned by actual authoritative name servers. The results presented in this work are based on data collected for a period of two months, from February 24th to April 25th, 2019.

4.4.1 Data Analysis

In this section we use two metrics, k-anonymity and the dynamics of hosting IP addresses, to quantify the privacy benefits offered by different hosting and CDN providers. To verify the validity of our k-anonymity model, we also apply Zipf's law on the popularity ranking of domains to account for a more realistic (i.e., more powerful) adversary.

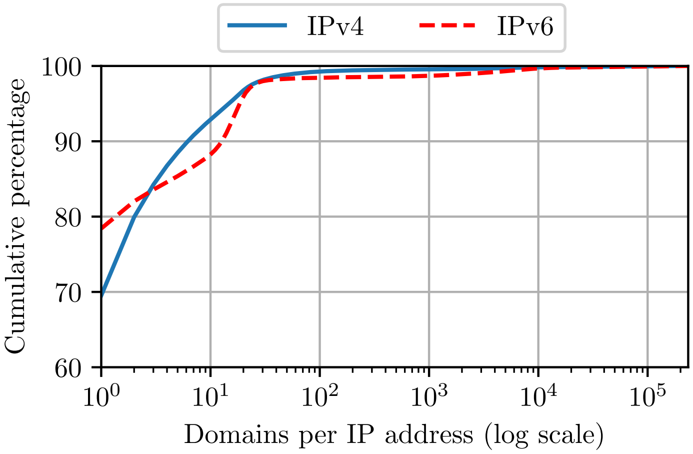


Figure 4.1. Cumulative distribution function (CDF) of the number of domains hosted per IP address, as a percentage of all observed IP addresses. About 70% of all observed IPv4 addresses host only a single domain.

4.4.1.1 Single-hosted vs. Multi-hosted Domains

Over a period of two months, from February 24th to April 25th, 2019, we observed an average of 2.2M and 500K unique IPv4 and IPv6 addresses, respectively, from our daily measurements. Of these IP addresses, 70% of IPv4 and 79% of IPv6 addresses host only a single domain, as shown in Figure 4.1. This means that visitors of the websites hosted on those addresses will not gain any meaningful privacy benefit with ESNI, due to the one-to-one mapping between the domain name and the IP address on which it is hosted. About 95% of both IPv4 and IPv6 addresses host less than 15 domains.

When calculating the percentage of IPv6-supported sites, we find that less than 15% support IPv6. Regardless of the increasing trend [74], the future adoption of IPv6 is still unclear [70]. Since the majority of web traffic is still being carried through IPv4, in the rest of the paper we focus only on IPv4 addresses.

Based on our measurements, we identify three main ways in which a domain may be hosted, in terms of the IP addresses used and the potential privacy benefit due to ESNI, as illustrated in Figure 4.2. In the simplest case, a *single-hosted* domain

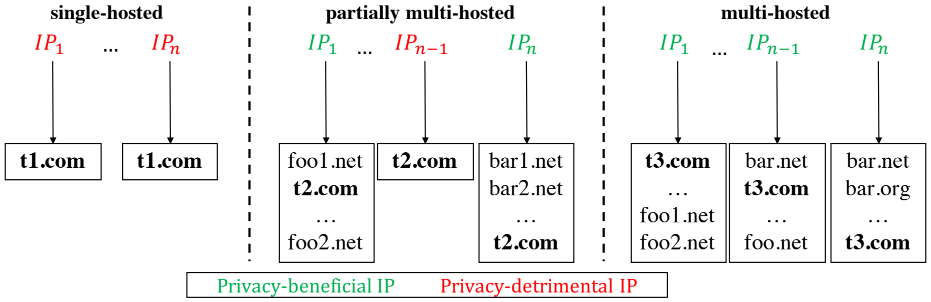


Figure 4.2. Different types of domain hosting according to whether they can benefit from ESNI. Single-hosted domains are exclusively hosted on one or more IP addresses, and thus cannot benefit from ESNI. In contrast, multi-hosted domains are always co-hosted with more domains on a given IP address, and thus can benefit by ESNI.

may be *exclusively hosted* on one or more IP addresses that do not serve any other domain, to which we refer as *privacy-detrimental* IP addresses (Figure 4.2, left). As there is no sharing of the IP address(es) with other domains, an adversary can trivially learn which site is visited based solely on the destination IP address. On the other hand, a *multi-hosted* domain (Figure 4.2, right) may be *co-hosted* on one or more IP addresses that always serve at least one or more other domains, to which we refer as *privacy-beneficial* IP addresses. Since the destination IP address always hosts multiple domains, an adversary can only make a (possibly educated) guess about the actual domain a given user visits, and thus multi-hosted domains always benefit to some extent from ESNI—the more co-hosted domains on a given IP address, the higher the privacy gain offered by ESNI.

Finally, there is a chance that a domain is hosted on a mix of privacy-detrimental and privacy-beneficial IP addresses, which we call *partially multi-hosted* domains (Figure 4.2, middle). In that case, only visitors to the subset of IP addresses that co-host other domains will benefit from ESNI. Based on our measurements, partially multi-hosted domains correspond to only a 0.3% fraction (20K) of all domains (daily average). Single-hosted domains comprise 18.7% (1.4M) and multi-hosted domains comprise 81% (6M) of all domains.

The privacy degree of a partially multi-hosted domain depends on the probability that a visitor gets routed to a privacy-beneficial IP of that domain. In other words, a partially multi-hosted domain will mostly behave as a multi-hosted domain if the majority of its IP addresses are privacy-beneficial. In fact, we find that this is the case for more than 92.5% of the partially multi-hosted domains studied. Based on this fact, and given its extremely small number compared to the other two types, in the rest of our paper we merge partially multi-hosted domains with the actual multi-hosted domains, to simplify the presentation of our results.

Going back to Figure 4.1, based on the above breakdown, we observe that 70% of all IP addresses that host a single domain correspond to 18.7% of all domains, i.e., the single-hosted ones. On the other hand, the 81% of multi-hosted domains are co-hosted on just 30% of the IP addresses observed.

Next, we analyze the popularity distribution of single-hosted and multi-hosted domains to identify any difference in the user population of these two types of domains. Note that we only base our analysis on the ranking information provided by the top lists to comparatively estimate the scale of the user base, and not for absolute ranking purposes. More specifically, we only use the top 100K domains for the analysis in Figure 4.3, since rankings lower than 100K are not statistically significant, as confirmed by both top list providers and recent studies [27, 219]. Figure 4.3 shows that single-hosted and multi-hosted domains exhibit a nearly identical distribution of popularity rankings.

4.4.1.2 Estimating the Privacy Benefit of Multi-hosted Domains

In this section, we focus on the 81% of multi-hosted domains that can benefit from ESNI, and attempt to assess their actual privacy gain. Recall that a website can gain some privacy benefit only if it is co-hosted with other websites, in which case an on-path adversary will not know which among all co-hosted websites is actually being

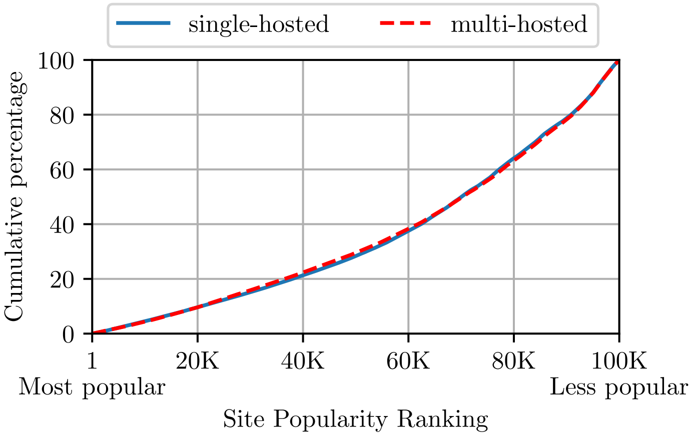


Figure 4.3. CDF of the popularity ranking for single-hosted and multi-hosted domains.

visited. We use k-anonymity to model and quantify the privacy gain of multi-hosted domains.

Going back to Figure 4.2, we can apply this definition in two ways, depending on whether we focus on IP addresses or domains. For a given IP address, its k-anonymity value ("k" for brevity) corresponds to the number of co-hosted domains. For a given multi-hosted domain, its k may be different across the individual IP addresses on which it is hosted, as the number of co-hosted domains on each of those addresses may be different. Consequently, the k value of a multi-hosted domain is calculated as the median k of all its IP addresses.[1] In both cases, the privacy gain increases linearly with k. Based on these definitions, we now explore the privacy gain of domains hosted on different hosting and CDN providers.

Table 4.3 shows the top-ten hosting providers offering the highest median k-anonymity per IP address (i.e., greater than 500). As shown in the third column, the average number of unique IP addresses observed daily for each provider is very low, with half of them hosting all domains under a single IP address. Using the Hurri-

[1] Since most domains have similar k values across their hosting IP addresses, both mean and median can be used in this case.

Table 4.3. Top hosting providers offering the highest median k-anonymity per IP address.

Median k	Organization	Unique IPs	Highest Rank
3,311	AS19574 Corporation Service	2	1,471
2,740	AS15095 Dealer Dot Com	1	80,965
2,690	AS40443 CDK Global	1	68,310
1,338	AS32491 Tucows.com	1	22,931
1,284	AS16844 Entrata	1	96,564
946	AS39570 Loopia AB	6	19,238
824	AS54635 Hillenbrand	1	117,251
705	AS53831 Squarespace	23	386
520	AS12008 NeuStar	2	464
516	AS10668 Lee Enterprises	4	3,211

cane Electric BGP Toolkit, we confirmed that these providers are indeed small, with many of them managing less than 10K IP addresses allocated by regional Internet registries. When looking into the popularity of the websites hosted by these providers, as shown in the last column, the highest ranked website is only at the 386th position, hosted on Squarespace, while more than half of these providers host websites that are well below the top 10K.

Next, we investigate the k-anonymity offered by major providers that dominate the unique IP addresses observed. Table 4.4 lists the top-20 major hosting and CDN providers with more than 5K unique IP addresses observed. Unlike small hosting providers, these major providers are home to more popular sites. Indeed, the most popular sites hosted by these providers are all within the top 10K. In contrast to small providers, however, the median k-anonymity per IP address offered by these providers is quite low, meaning that sites hosted on them will gain a much lower level of privacy. Except from Cloudflare, which has the highest k of 16, all other providers have a single-digit k.

Tables 4.3 and 4.4 represent two ends of the privacy spectrum for multi-hosted domains. On one end, numerous but less popular domains are hosted on providers

Table 4.4. Top hosting providers with highest number of observed IP addresses.

Median k	Organization	Unique IPs	Highest Rank
16	AS13335 Cloudflare, Inc.	64,285	112
5	AS16509 Amazon.com, Inc.	47,786	37
5	AS46606 Unified Layer	27,524	1,265
3	AS16276 OVH SAS	22,598	621
3	AS24940 Hetzner Online GmbH	21,361	61
4	AS26496 GoDaddy.com, LLC	16,415	90
2	AS14061 DigitalOcean, LLC	11,701	685
3	AS14618 Amazon.com, Inc.	11,008	11
6	AS32475 SingleHop LLC	10,771	174
2	AS26347 New Dream Network	10,657	1,419
7	AS15169 Google LLC	9,048	1
3	AS63949 Linode, LLC	8,062	2,175
4	AS8560 1&1 Internet SE	6,898	2,580
3	AS32244 Liquid Web, L.L.C	6,412	1,681
3	AS19551 Incapsula Inc	6,338	1,072
4	AS36351 SoftLayer Technologies	6,005	483
3	AS16625 Akamai Technologies	5,862	13
4	AS34788 Neue Medien Muennich	5,679	7,526
6	AS9371 SAKURA Internet Inc.	5,647	1,550
3	AS8075 Microsoft Corporation	5,360	20

managing a handful of IP addresses, benefiting from high k-anonymity; on the other end, fewer but more popular websites are hosted on providers managing a much larger pool of millions of IP addresses, suffering from low k-anonymity.

To provide an overall view of the whole privacy spectrum, Figure 4.4 shows CDFs of k of all studied domains across nine different regions. As illustrated, k values are almost identical across the nine regions from which we conducted our measurements. While our DNS data shows that there are 471K (CDN-supported) domains served from different IP addresses depending on the resolution location, the k values of these domains remain similar regardless of the DNS resolution origin.

As discussed in §4.3.2, a low (e.g., single-digit) k cannot allow ESNI to offer meaningful privacy, given that i) not all k sites will be equally popular, and ii) website

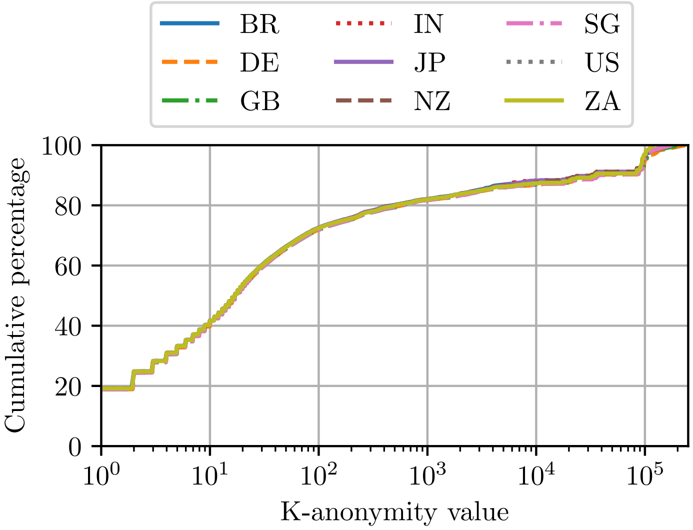


Figure 4.4. CDF of the k-anonymity for all studied domains across nine measurement locations (k=1 corresponds to the 18.7% of single-hosted domains).

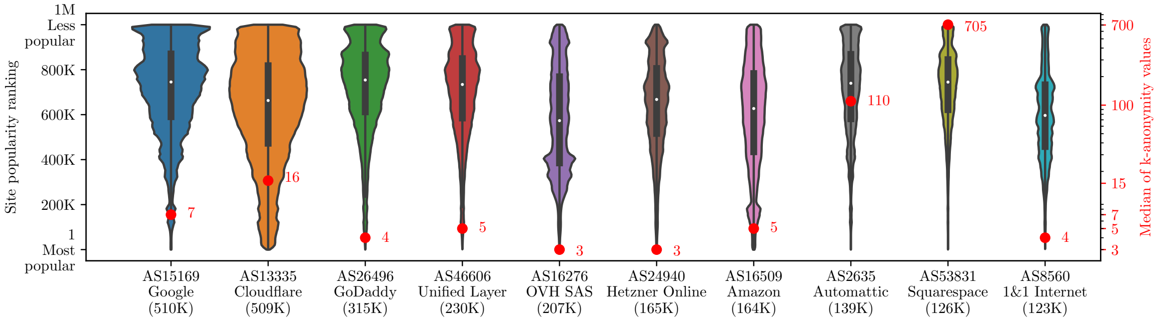


Figure 4.5. Top providers that host most domains.

fingerprinting can be used to improve attribution accuracy even further [51, 166, 191, 194, 245]. Assuming that k needs to be greater than 100 to provide meaningful privacy, since an adversary would correctly guess a domain being visited with a probability less than 1%, then according to Figure 4.4, only about 30% of the sites will benefit from domain name encryption. We conduct a more in-depth analysis of the probability with which an adversary would correctly guess domains being visited based on Zipf's law in §4.4.2.

Finally, we examine the top-10 providers that host the largest number of domains among the ones studied. Although these mostly include some of the providers listed in Table 4.4, two of them are not included on that table, and one (Squarespace) is actually included in Table 4.3. The violin plot of Figure 4.5 depicts the top-ten providers that host most domains. The area of each violin is proportional to the number of domains hosted by that provider, while the shape of each violin illustrates the popularity ranking distribution of hosted websites. The median k of each provider is denoted by the red dot. Google and Cloudflare are the top hosting providers, with more than 500K domains each. Other providers host different numbers of domains, ranging from 315K to 123K.[2] Although hosting fewer domains, both Automattic and Squarespace provide significantly higher privacy with a k of 110 and 705, respectively.

4.4.2 Weighting the Privacy Benefit Based on Domain Popularity

In §4.4.1.2, we used the k-anonymity model to quantify the privacy benefit provided by multi-hosted domains. However, one might consider that the model does not accurately capture a real-world adversary, as not all co-hosted domains are equally popular. Adversaries could base their guess on the probability that a domain is more (or less) likely to be visited, according to the visit frequency of that domain compared to other co-hosted domains. However, it is infeasible for us to obtain the data of domain visit frequencies, since this is only known by the respective hosting providers.

Fortunately, prior studies have shown that the relative visit frequency of domains follows Zipf's law [46, 265]. More specifically, Zipf's law states that the relative probability of a domain (d) being visited is inversely proportional to its popularity

[2] Note that a website may be hosted on more than one provider [135]. In that case, we count the site separately for each hosting provider.

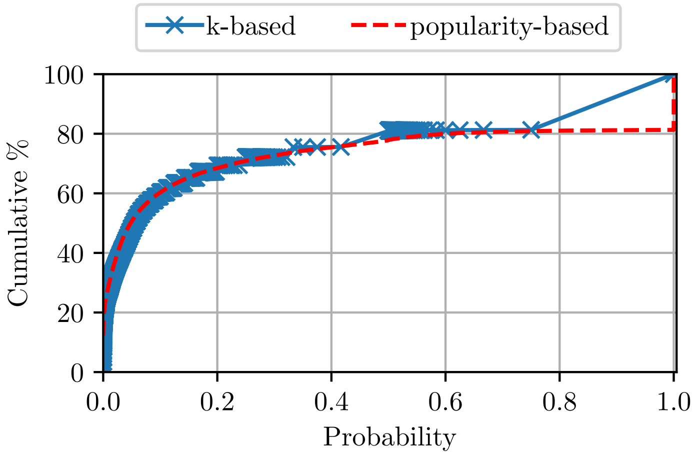


Figure 4.6. CDF of the probability of correctly guessing a visited domain based on the k-anonymity value and popularity ranking information, as percentage of all tested domains.

ranking ($P_d \propto 1/(rank_d)^\alpha$). We thus apply Zipf's law[3] on the popularity ranking of domains to compute the probability with which an adversary can correctly guess that a given domain is being visited.

From a privacy-detrimental IP address, it is straightforward for the adversary to learn the domain being visited as the IP address solely hosts that single domain. However, given a privacy-beneficial IP address that hosts multiple domains, a more realistic adversary would make his guess based on the probability that a domain is more likely to be visited compared to other co-hosted domains. In order to compute this probability, we first obtain the domains $d_1, \ldots, d_n$ that are hosted on a single IP_j and compute their P_d values according to Zipf's law. We define $P_{d_{ij}} = \frac{P_{d_i}}{\sum_{k=1}^{n} P_{d_k}}$ as the probability that domain d_i was visited when IP_j was observed.

For domains that are hosted on multiple IP addresses, the probability is estimated by taking the median of all probabilities that the domain is visited from all IP addresses hosting it. We therefore compute the probability that an adversary can

[3] For simplicity, we present results with $\alpha = 1$, following the strict Zipf's law. However, adjusting the value of α to match previous observations [46] also gave similar results.

correctly guess domain d_i that is hosted on $IP_1, \ldots, IP_m$ as follows:

$$P_i = median(P_{d_{i1}}, \ldots, P_{d_{im}}) \tag{4.1}$$

As shown in Figure 4.6, our k-anonymity model is a close lower bound to the case where the adversary considers the popularity rankings. The figure shows two CDFs of the probability that the adversary can guess which domain is being visited. The continuous (blue) line is computed based on the k-anonymity value of co-hosted domains. Each domain has an equal probability of $1/k$ to be visited. The dashed (red) line is computed by applying the Zipf's law on the domain popularity. We can see that even if adversaries rely on domain popularity rankings to improve the accuracy of their prediction, the highest probability that this guess is correct is similar to the probability estimated by the k-anonymity value.

4.4.2.1 Domain-to-IP Mapping Stability

Besides the degree of co-hosting, the stability of a website's IP address(es) also plays an important role in whether ESNI will provide meaningful privacy benefits. If the IP address of a website changes quite frequently, this will have a positive impact on the privacy offered due to ESNI. Unless adversaries have enough resources to acquire all domain-to-IP mappings of interest at almost real-time, they will no longer be able to use the destination IP address as an accurate predictor of the visited website, because a previously known domain-to-IP mapping may not be valid anymore. On the other hand, mappings that remain stable over the time make it easier for adversaries to monitor the visited websites.

In this section, we examine the stability of domain-to-IP mappings, and how it affects privacy. We are particularly interested in finding how often domain-to-IP mappings change. As discussed in §4.3.3, all top lists of popular sites have their own churn (i.e., new sites appear and old sites disappear from the lists on a daily basis).

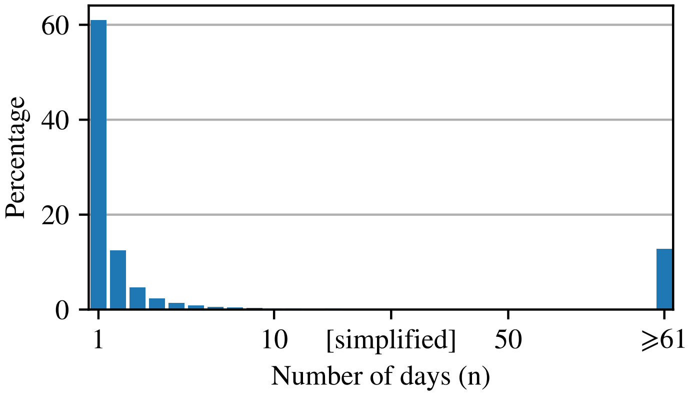


Figure 4.7. Longevity distribution of domain-to-IP mappings as percentage of number of mappings.

To prevent this churn from affecting our analysis, we consider only the subset of domains that were present daily on both top lists (§4.4.0.1) during the whole period of 61 days of our study. This set of domains comprises 2.6M domains, from which we observed a total of 22.7M unique domain-to-IP mappings because a domain may be hosted on hundreds of IP addresses.

Figure 4.7 shows the distribution of the longevity of these mappings in days. More than 80% of the mappings last less than four consecutive days (*short-term mappings*), corresponding to 202K (7.7%) domains served from 400K unique IP addresses. On the other hand, 13% of the mappings remain unchanged for the whole study period (*long-term mappings*), corresponding to 2.4M domains served from 1.1M unique IP addresses. As also shown in Figure 4.7, there are two dominant clusters of domains that either change their hosting IP addresses frequently or do not change at all. This is a favorable result for adversaries, as it implies that they do not have to keep resolving a large number of domains, since most domain-to-IP mappings remain quite stable over a long period.

The popularity distribution of the domains that correspond to these two short-term and long-term mappings is shown in Figure 4.8. While domains with short-term mappings are evenly distributed across the popularity spectrum, domains with long-

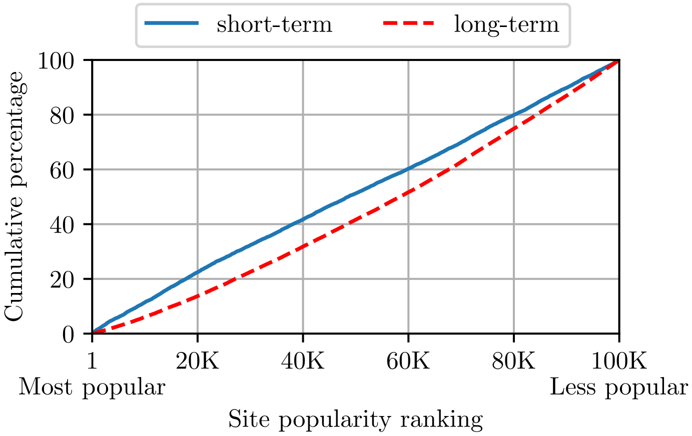


Figure 4.8. CDF of domain popularity for short-term and long-term domain-to-IP mappings.

term mappings slightly lean towards lower popularity rankings. This result can be attributed to the fact that more popular websites are more likely to rotate their IP addresses for load-balancing reasons, while less popular sites are more likely to be served from static IP addresses.

An increased churn of IP addresses also helps ESNI provide better privacy. We thus investigated which providers exhibit the highest churn rate by grouping the IP addresses of short-term mappings according to their ASN. Figure 4.9 shows the top-ten providers with the highest number of IP addresses in short-term mappings (bars). The dots indicate the number of domains hosted on those IP addresses. Although Amazon and Akamai do not top the list of providers that host most domains (Figure 4.5), along with Cloudflare they occupy the top five positions of the providers with the highest number of dynamic IPs. Google uses a relatively small pool of around 5.3K IP addresses, to host more domains (41K) than the other providers.

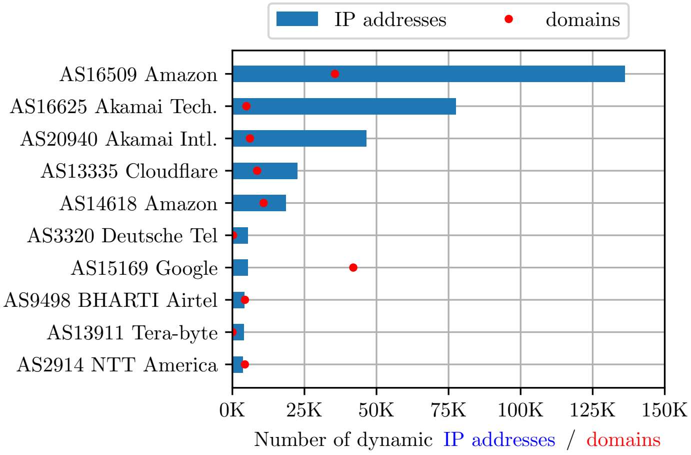


Figure 4.9. Top providers with the highest number of high-churn IP addresses.

4.4.3 Comparison with Other Datasets

In this section, we analyze existing public DNS datasets to examine the impact of i) larger datasets (in terms of number of domains), and ii) more localized vantage points, on the estimation of per-domain k-anonymity.

The Active DNS Project [159] is currently collecting A records of about 300M domains derived from 1.3K zone files on a daily basis. In addition to this effort, Rapid7 [208] also conducts active DNS measurements at a large scale and offers researchers access to its data. Unlike the Active DNS Project, Rapid7 resolves a much larger number of domains (1.8B), but with a lower frequency (domains are resolved only once a week). The dataset includes not only second-level domains from several TLD zone files, but also lower-level domains obtained through web crawling and targeted scanning with Zmap [89].

Different from these datasets, as discussed in §4.4.0.1, our domain name dataset is curated from the global lists of Alexa and Majestic. We also perform measurements from vantage points around the world to observe localized DNS responses from CDNs. In contrast, the above datasets are collected from local vantage points, as their goal is to maximize the number of observed domains, and not to exhaustively resolve all

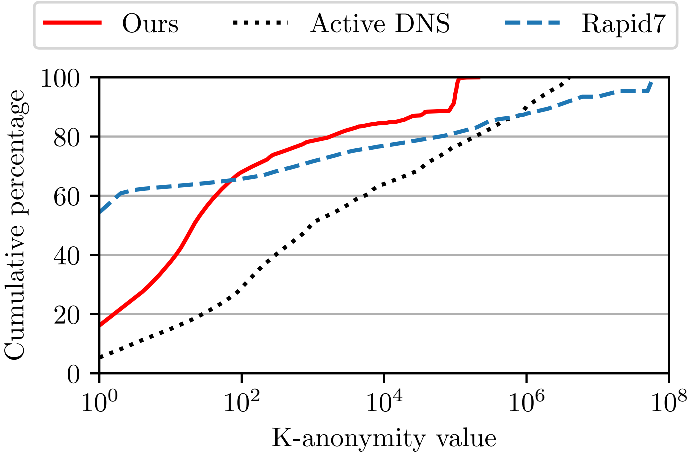


Figure 4.10. CDF of the k-anonymity value per domain as a percentage of *all domains* observed from the Active DNS Project, Rapid7, and our datasets.

potential IP addresses of a domain. In particular, the Active DNS Project is run from Georgia Tech, while Rapid7's data is collected using AWS EC2 nodes in the US. To that end, we used two datasets from the Active DNS Project and Rapid7 collected on March 29th, 2019 for our comparison. We sanitized poisoned records from these datasets, as described in §4.4, before analyzing them.

Figure 4.10 shows the CDF of the k-anonymity value per domain for all the domains in the Active DNS, Rapid7, and our datasets, while Figure 4.11 shows the CDF of the k-anonymity value per domain for only common domains among the three datasets.

In Figure 4.10, when $k = 1$, there is some difference in the percentage of single-hosted domains between the Active DNS Project (5.3%), Rapid7 (54.3%), and our observation (18.7%). As expected, the percentage of single-hosted domains for Rapid7 is the highest because this dataset is the largest (1.8B FQDNs), and includes lower-level FQDNs that may host other services (e.g., email, DNS, SSH) instead of web content. On the other hand, only 5.3% of the domains in the Active DNS dataset are single-hosted, since the dataset contains mostly A records of domains extracted from TLD zone files, instead of many lower-level FQDNs.

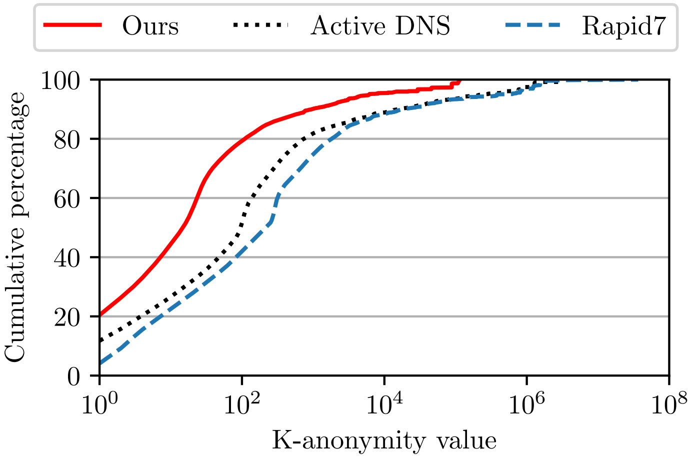


Figure 4.11. CDF of the k-anonymity value per domain as a percentage of *common domains* observed from the Active DNS Project, Rapid7, and our datasets.

When all domains are considered, our observation of single-hosted domains is in between the two (18.7%) because (as mentioned in §4.4.0.1) we derive our domain name dataset from the two global top websites lists (Alexa and Majestic), and include not only second-level domains but also lower-level FQDNs, as long as they are included in the lists and serve web content. This aligns well with the results of Rapid7 and Active DNS, given the inherent over-approximation of the Rapid7 dataset (due to non-web services and highly unpopular or even single-use domains) and the inherent under-approximation of the Active DNS Project dataset (due to only domains extracted from TLD zone files).

When considering only common domains, the percentages of single-hosted domains are 4.1%, 11.7%, and 20.4% for the Rapid7, Active DNS, and our dataset, respectively. In other words, some domains classified as single-hosted in our dataset are actually multi-hosted when considering the larger datasets. This result confirms our hypothesis discussed in §4.4.0.1 that by only considering the two sets of popular websites, we would have missed those less popular, random, or even dormant domains in the long tail. Thus, the result provided by our dataset can be considered as a lower bound value of the actual k-anonymity, since any increase in k provided by less well-known, random, or even dormant domains is less meaningful from the perspective of adversaries whose goal is to reveal a visited website by incorporating the popularity ranking information in their “guess.”

At the right end of the CDFs in both Figures 4.10 and 4.11, the two larger datasets exhibit k-anonymity values higher than ours. The primary reason for this is that these datasets include not only second-level domains, but also third-level and longer FQDNs. The higher k values also comprise the long-tail of domains that are not included in our dataset, including less popular domains, random or single-use FQDNs used for tracking, and malicious domains [231].

4.4.4 Reverse DNS Lookups

The Internet Engineering Task Force (IETF) recommends that it should be possible to conduct a reverse DNS lookup for every given domain [93]. In a forward DNS lookup, a domain name is resolved to an A (IPv4) record, while a reverse DNS lookup sends out an IP address to ask for its associated FQDN. The DNS record storing this information is called a PTR (pointer) record. Unless configured to point to a FQDN by its owner, it is not compulsory to configure PTR records for every IP address.

Although performing reverse DNS lookups seems to be a straightforward way of mapping a given IP address back to its associated FQDN, this can potentially uncover only single-hosted domains. More importantly, not all reverse DNS queries return a (meaningful[4]) domain name because the IETF’s recommendation is only optional, and thus not adopted universally.

We analyzed Rapid7’s reverse DNS dataset, which contains PTR records for the whole public IPv4 space, to see how many IP addresses could be used to reveal the visited destinations under the assumed global ESNI deployment, by just performing

[4] Many PTR records are formatted with dash-separated IP segments. For example, the Amazon EC2 IP address *54.69.253.182* has a PTR record to *ec2-54-69-253-182.us-west-2.compute.amazonaws.com*, which may not actually correspond to a user-facing web service.

a reverse DNS lookup. We find that there are 1.27B unique IP addresses having PTR records. Of these, at least 172M (14%) of them point to a meaningful FQDN (i.e., not in the form of dash-separated IP segments). Within this set of domains, we could find 136K single-hosted domains observed by our dataset. This means about 10% of single-hosted domains have PTR records configured. Under a global ESNI deployment, IP addresses of these domains would be detrimental to the privacy of users who connect to them.

As expected, we also observed more than 3M PTR records in which domain names explicitly indicate through the prefix "mail" that they correspond to email servers. These email servers support PTR record because many providers will not accept messages from other mail servers that do not support reverse lookups [167].

4.4.5 Discussion

4.4.5.1 Recommendations

While the *security* benefits of DoH/DoT against on-path adversaries are clear (e.g., prevention of MiTM or MoTS DNS poisoning attacks), our findings show that Encrypted SNI alone cannot fully address the *privacy* concerns it aims to tackle. More effort and collaboration from all involved parties (i.e., operators of DNS authoritative name servers, website owners, and hosting and CDN providers) are needed. In this section we provide some suggestions for maximizing the privacy benefits of ESNI.

Full Domain Name Confidentiality. In the current designs, plaintext domain names are exposed through two channels: the SNI extension in TLS, and traditional DNS name resolutions—the deployment of DoH/DoT is thus a prerequisite for ESNI. Equivalently, the use of DoH/DoT will not provide any meaningful privacy if domain names are still exposed through the (unencrypted) SNI extension in TLS handshake traffic.

Recently, there is a push for the deployment of DoH/DoT, with major organizations already supporting it (e.g., Google, Cloudflare, Firefox), though this has not been followed by an equivalent effort for the deployment of ESNI. An even more complicated method of securing DNS traffic is DNS-over-HTTPS-over-Tor, which has been already implemented and supported by Cloudflare [220]. Unless the confidentiality of domain names is preserved on both channels (TLS and DNS), neither technology can provide any actual privacy benefit if deployed individually.

Domain Owners. Website owners who want to provide increased privacy to their users can seek hosting providers or CDNs that offer an increased ratio of co-hosted domains per IP address and/or highly dynamic domain-to-IP mappings. In practice, however, this may be challenging. Our results show that unfortunately only a few providers offer a high domain-to-IP ratio, while other more pressing factors (e.g., site popularity) may tilt the decision towards other more important factors, such as latency, bandwidth, or points of presence.

While pointer (PTR) records are often not configured, from a privacy perspective, their operation conflicts with DoH/DoT and ESNI as further discussed in §4.4.4. Consequently, website operators should not configure PTR records unless absolutely necessary (e.g., for email servers). In addition, providers with a higher rotation of IP addresses are more preferable, as this also helps in improving privacy.

Hosting Providers. Hosting and CDN providers are in a more privileged position to achieve meaningful impact in helping improve the potential privacy benefits of ESNI, as they can control the number of co-hosted domains per IP address, and the frequency of IP addresses rotation. Unless website owners prefer otherwise, providers could group more websites under the same IP address (which, understandably, may not be desirable for some websites).

To improve k-anonymity even more meaningfully, providers should cluster websites according to similarities in terms of traffic patterns and popularity ranking, to hinder website fingerprinting attempts. As discussed in §4.4.2.1, more dynamic hosting IP addresses can also help improve visitors' privacy. Currently, the number of websites benefiting from more short-lived domain-to-IP mappings is relatively small. While more frequent IP address changes may complicate operational issues and are certainly more challenging to deploy from a technical perspective (especially for smaller providers), existing load balancing schemes already provide such a capability, which could be tuned to also maximize privacy. In the future, it may be worthwhile to explore more sophisticated schemes that actively attempt to maximize privacy by increasing the "shuffling" rate of co-hosted domains per IP address, to hinder attribution even further, especially when considering more determined adversaries.

4.4.5.2 Impact

The deployment of domain encryption has various advantages and disadvantages from the two—rather conflicting—perspectives of Internet censorship and network visibility.

The existing plaintext exposure of domain names on the wire, as part of DNS requests and TLS handshakes, has enabled the wide use of network traffic filtering and censorship based on domain names [32, 86, 100, 133, 196, 222]. In a future with all domain name information being encrypted, DNS and SNI traffic will no longer be an effective vector to conduct censorship. It is likely that censors will shift to use IP-based blocking, which can be very effective if hosting IP addresses of censored websites are stable and host only a handful of sites or services [88, 131]. However, if providers start adapting according to the above mentioned recommendations, the cost of conducting IP-based blocking will increase, since a censor will have to keep track of which IP address belong to which websites.

More importantly, the collateral damage caused by this type of blocking will also increase dramatically if censored websites are co-hosted with multiple other innocuous websites [135]. Although some previous actions from the side of providers (e.g., hindering domain fronting [102]) have shown that privacy is often given a secondary priority [209], as the collateral damage caused to censors may also impact significantly the providers, the renewed recent focus on privacy as a potential competitive advantage by some providers may encourage the deployment of hosting schemes that will improve the privacy benefits of ESNI.

On the other hand, while providing many security and privacy benefits, domain name encryption can be a "double-edged sword" for network administrators who want to have full visibility and control over domain resolutions in the networks under their responsibility. Until now, the operation of firewalls, intrusion detection systems, and anti-spam or anti-phishing filters has benefited immensely from the domain name information extracted from network traffic, as is evident by the series of works mentioned in §4.2 that employ DNS data to detect domain name abuses and malicious online activities.

Under a full DoH/DoT and ESNI deployment, this visibility will be lost, and systems based on domain reputation [34] and similar technologies will be severely impacted. While many malicious domains often hide themselves by sharing hosting addresses with other innocuous and unpopular websites [231], it will be challenging to detect and block them. A possible solution would be to rely solely on TLS proxying using custom provisioned certificates, in order to gain back the visibility lost by ESNI and DoH/DoT, which is already a common practice used by transparent SSL/TLS proxies. Although this will defeat any privacy benefits of these technologies, this may be an acceptable trade off for corporate networks and other similar environments.

4.4.6 Conclusion

The deployment of encrypted SNI in TLS, combined with DNS over HTTPS/TLS, will definitely provide many security benefits to Internet users. However, as we have shown in this work, a significant effort is still needed in order for these same technologies to provide meaningful privacy benefits. More specifically, while domain name information is encrypted, the IP address information is still visible to any on-path observers and can be used to infer the websites being visited.

Using DNS data collected through active DNS measurements, we studied the degree of co-hosting of the current web, and its implications in relation to ESNI's privacy benefits. Quantifying these benefits for co-hosted websites using k-anonymity, we observed that the majority of popular websites (about half of all domains studied) will gain only a small privacy benefit (k<16). Such a small degree of co-hosting is not enough to withstand determined adversaries that may attempt to perform attribution by considering the popularity or even the traffic patterns of the co-hosted websites on an observed destination IP address. Domains that will obtain a more meaningful privacy benefit (k>500) include only vastly less popular websites mostly hosted by smaller providers, while 20% of the websites, will not gain any benefit at all due to their one-to-one mapping between domain name and hosting IP address.

We hope that our findings will raise awareness about the remaining effort that must be undertaken to ensure a meaningful privacy benefit from the deployment of ESNI. In the meantime, privacy-conscious website owners may seek hosting services offered by providers that exhibit a high ratio of co-hosted domains per IP address, and highly dynamic domain-to-IP mappings.

CHAPTER 5

CONCLUSION

5.1 Summary

In this dissertation, I focused on developing measurement techniques and platforms that study network interference globally. Through different measurement studies on network interference, I shed light on new findings of how network interference happens across the globe longitudinally. In the second part of this thesis, I study how DNS encryption technologies affect users' privacy over the Internet and impact network interference.

In our global Internet censorship study (ICLab), I leveraged commercial VPN vantage points that provide us flexibility and control over measurements, while reducing risks in measuring Internet censorship at a global scale. Collecting data from all levels of the network stack allows us to analyze different types of censorship while minimizing false positives and manual validation.

Our studies on the Great Firewall of China revealed a previously unknown DNS poisoning behavior of the GFW, where I identified three distinct DNS packet injectors. Further, by testing 534M domains over a nine-month period, I find that the GFW's DNS censorship has a widespread negative impact on the global Internet, including poisoned resource records in many popular public DNS resolvers. I also identify different groups of domain names that receive a set of fixed forged IP addresses or CNAME records. Using the insights gained from the data collected by our platform, I propose strategies to detect poisoned responses and evade GFW's DNS censorship effectively.

Our study of traffic differentiation demonstrated that most throttling targets video streaming, and there are a wide range of throttling implementations detected in our dataset. Finally, I find that while throttling does limit video resolution, the default settings in video streaming apps, in some cases, are the primary reason for low resolution.

Our study of the privacy benefits of DNS encryption technologies showed that while the deployment of ESNI and DNS encryption technologies will provide many security benefits to Internet users, a significant effort is needed for these technologies to provide meaningful privacy benefits. This is because the IP address information is still visible to any on-path observers and can be used to infer the websites being visited. Analyzing the co-hosting degree of websites, I observed that the majority of popular websites would gain a small privacy benefit. In contrast, the less popular websites that are primarily hosted by smaller providers obtain a more meaningful privacy benefit.

5.2 Future Work

The research presented in this thesis can be extended in different directions. With respect to Internet censorship measurement, given various active global censorship measurement platforms, one research direction is to study the difference in the list of censorship domains and events that each platform reports. Since each platform leverages different types of vantage points (i.e., volunteers, VPNs, and remote measurements), complimenting these platforms would be beneficial to the community.

I leveraged crowd-sourced data using mobile phone measurements to study content-based traffic differentiation policies deployed in operational networks in the traffic differentiation paper. A future extension of this work would be to study traffic differentiation with the intention of slowing down users' connections. Previous work [264] has reported widespread and persistent slowdowns across more than 400 nodes in

China. Therefore, both commercial and user-based VPN vantage points can be leveraged to measure traffic differentiation globally and longitudinally.

Regarding our work on studying the DNS poisoning behavior of the GFW, it is important to note that network interference can also happen at other layers of the network stack. More specifically, blocking can happen at the application layer (e.g., SNI-based blocking [62], keyword-based filtering [207]) or even at the IP layer [129, 132], regardless of potential collateral damage [135]. A future research direction is to study how the adoption of DNS encryption technologies such as ESNI that encrypts the SNI field would impact the network interference of the GFW.

BIBLIOGRAPHY

[1] The Common Crawl Project. URL https://commoncrawl.org.

[2] ICANN Centralized Zone Data Service. URL https://czds.icann.org.

[3] McAfee: Customer URL Ticketing System. URL https://www.trustedsource.org/?p=mcafee.

[4] Verisign Zone File Service. URL https://www.verisign.com/en_US/channel-resources/domain-registry-products.

[5] Virus Total: URL Scanning Service. URL https://www.virustotal.com/gui/home/url.

[6] FortiGuard Labs Web Filter. URL https://fortiguard.com/webfilter. https://fortiguard.com/webfilter.

[7] Shodan: The search engine for Security. URL https://shodan.io/.

[8] At&t stream saver. https://www.att.com/offers/streamsaver.html, April 2017.

[9] All you need to know about net neutrality rules in the eu. https://berec.europa.eu/eng/netneutrality/, April 2018.

[10] Fcc releases restoring internet freedom order. https://www.fcc.gov/fcc-releases-restoring-internet-freedom-order, January 2018.

[11] Verizon unlimited plans. https://www.verizonwireless.com/plans/unlimited/, December 2018.

[12] Apple is blocking an app that detects net neutrality violations from the app store. https://motherboard.vice.com/en_us/article/j5vn9k/apple-blocking-net-neutrality-app-wehe, April 2018.

[13] Wehe: Check your isp for net neutrality violations. https://dd.meddle.mobi/, April 2018.

[14] At&t plans. https://www.att.com/plans/wireless.html, January 2019.

[15] Recording a packet trace. https://developer.apple.com/documentation/network/recording_a_packet_trace, June 2019.

[16] Wireshark's tcp analysis. https://www.wireshark.org/docs/wsug_html_chunked/ChAdvTCPAnalysis.html, June 2019.

[17] China forcing birth control on Uighurs to suppress population, report says. BBC News, 2020-06-29. URL https://www.bbc.com/news/world-asia-china-53220713.

[18] Josh Aas and Sarah Gran. Let's Encrypt Has Issued a Billion Certificates, 2019. URL https://letsencrypt.org/2020/02/27/one-billion-certs.html.

[19] Josh Aas, Richard Barnes, Benton Case, Zakir Durumeric, Peter Eckersley, Alan Flores-López, J. Alex Halderman, Jacob Hoffman-Andrews, James Kasten, Eric Rescorla, and et al. Let's Encrypt: An Automated Certificate Authority to Encrypt the Entire Web. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, page 2473–2487, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367479. doi: 10.1145/3319535.3363192. URL https://doi.org/10.1145/3319535.3363192.

[20] Nicholas Aase, Jedidiah R. Crandall, Álvaro Díaz, Jeffrey Knockel, Jorge Ocaña Molinero, Jared Saia, Dan Wallach, and Tao Zhu. Whiskey, Weed, and Wukan on the World Wide Web: On Measuring Censors' Resources and Motivations. In *Free and Open Communications on the Internet*. USENIX, 2012.

[21] Gunes Acar, Christian Eubank, Steven Englehardt, Marc Juarez, Arvind Narayanan, and Claudia Diaz. The web never forgets: Persistent tracking mechanisms in the wild. In *Computer and Communications Security*, pages 674–689. ACM, 2014. doi: 10.1145/2660267.2660347.

[22] Giuseppe Aceto, Antonio Montieri, and Antonio Pescapè. Internet Censorship in Italy: A First Look at 3G/4G Networks. In *Cryptology and Network Security*, pages 737–742. Springer, 2016. doi: 10.1007/978-3-319-48965-0_53.

[23] Sadia Afroz, Michael Carl Tschantz, Shaarif Sajid, Shoaib Asif Qazi, Mobin Javed, and Vern Paxson. Exploring Server-side Blocking of Regions. 2018.

[24] Pieter Agten, Wouter Joosen, Frank Piessens, and Nick Nikiforakis. Seven months' worth of mistakes: A longitudinal study of typosquatting abuse. In *Proc. Network and Distributed System Security Symposium (NDSS)*, 2015.

[25] Alexa Internet, Inc. How are Alexa's traffic rankings determined? URL http://www.alexa.com/faqs/?p=134. Accessed 2014.

[26] Alexa Internet, Inc. Top Sites, Accessed 2019. URL https://www.alexa.com/.

[27] Alexa Internet, Inc. How are Alexas's traffic rankings determined?, Accessed 2019. URL https://support.alexa.com/hc/en-us/articles/200449744-How-are-Alexa-s-traffic-rankings-determined-.

[28] Eihal Alowaisheq, Peng Wang, Sumayah Alrwais, Xiaojing Liao, XiaoFeng Wang, Tasneem Alowaisheq, Xianghang Mi, Siyuan Tang, and Baojun Liu. Cracking the Wall of Confinement: Understanding and Analyzing Malicious Domain Take-downs. In *Network and Distributed System Security*. Internet Society, 2019.

[29] Collin Anderson. Dimming the Internet: Detecting Throttling as a Mechanism of Censorship in Iran. 2013.

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

[31] Anonymous. The Collateral Damage of Internet Censorship by DNS Injection. *SIGCOMM Computer Communications Review*, 42(3):21–27, 2012. URL `http://www.sigcomm.org/node/3275`.

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

[33] Anonymous. GFW Archaeology: gfw-looking-glass.sh, March 2020. URL `https://gfw.report/blog/gfw_looking_glass/en/`.

[34] Manos Antonakakis, Roberto Perdisci, David Dagon, Wenke Lee, and Nick Feamster. Building a dynamic reputation system for DNS. In *the 19th USENIX Conference on Security*, pages 18–18, Berkeley, CA, USA, 2010. USENIX Association. ISBN 888-7-6666-5555-4.

[35] Manos Antonakakis, Roberto Perdisci, Wenke Lee, Nikolaos Vasiloglou, II, and David Dagon. Detecting malware domains at the upper DNS hierarchy. In *the 20th USENIX Conference on Security*, pages 27–27, Berkeley, CA, USA, 2011. USENIX Association.

[36] Manos Antonakakis, Roberto Perdisci, Yacin Nadji, Nikolaos Vasiloglou, Saeed Abu-Nimeh, Wenke Lee, and David Dagon. From Throw-away Traffic to Bots: Detecting the Rise of DGA-based Malware. In *the 21st USENIX Conference on Security Symposium*, pages 24–24, Berkeley, CA, USA, 2012. USENIX Association.

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

[38] Derek E. Bambauer, Ronald J. Deibert, J. Palfrey, Rafal Rohozinski, N. Villeneuve, and J. Zittrain. Internet Filtering in China in 2004-2005: A Country Study. 2005.

[39] Vitali Bashko, Nikolay Melnikov, Anuj Sehgal, and Jurgen Schonwalder. Bonafide: A traffic shaping detection tool for mobile networks. In *In Proc. of Integrated Network Management (IM2013)*, 2013.

[40] Berkman Klein Center. Website Inaccessibility Test Lists, 2018. URL `https://github.com/berkmancenter/url-lists`.

[41] Kevin Bock, George Hughey, Xiao Qiang, and Dave Levin. Geneva: Evolving censorship evasion strategies. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, page 2199–2214, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367479. doi: 10.1145/3319535.3363189. URL `https://doi.org/10.1145/3319535.3363189`.

[42] R. Bonica, M. Cotton, B. Haberman, and L. Vegoda. Updates to the Special-Purpose IP Address Registries. RFC 8190, 2017. URL `https://tools.ietf.org/html/rfc8190`.

[43] S. Bortzmeyer and S. Huque. NXDOMAIN: There Really Is Nothing Underneath. RFC 8020, 2016. URL `https://tools.ietf.org/html/rfc8020`.

[44] Timm Böttger, Felix Cuadrado, Gianni Antichi, Eder Leão Fernandes, Gareth Tyson, Ignacio Castro, and Steve Uhlig. An empirical study of the cost of dns-over-https. In *Proceedings of the Internet Measurement Conference*, IMC '19, page 15–21, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450369480. doi: 10.1145/3355369.3355575.

[45] T. Bray. An HTTP Status Code to Report Legal Obstacles. RFC 7725, 2016. URL `https://tools.ietf.org/html/rfc7725`.

[46] L. Breslau, Pei Cao, Li Fan, G. Phillips, and S. Shenker. Web caching and zipf-like distributions: evidence and implications. In *The IEEE Conference on Computer Communications*, volume 1, pages 126–134 vol.1, March 1999.

[47] Martin A. Brown. Pakistan hijacks YouTube. URL `https://dyn.com/blog/pakistan-hijacks-youtube-1/`. 2008, accessed 2018.

[48] Martin A Brown, Doug Madory, Alin Popescu, and Earl Zmijewski. DNS Tampering and Root Servers, 2010.

[49] Sam Burnett and Nick Feamster. Making Sense of Internet Censorship: A New Frontier for Internet Measurement. *SIGCOMM Computer Communications Review*, 43(3):84–89, 2013. doi: 10.1145/2500098.2500111.

[50] Sam Burnett and Nick Feamster. Encore: Lightweight Measurement of Web Censorship with Cross-Origin Requests. In *SIGCOMM*, pages 653–667. ACM, 2015. doi: 10.1145/2785956.2787485.

[51] Xiang Cai, Xin Cheng Zhang, Brijesh Joshi, and Rob Johnson. Touching from a Distance: Website Fingerprinting Attacks and Defenses. In *Proceedings of the ACM Conference on Computer and Communications Security*, 2012.

[52] Xiang Cai, Rishab Nithyanand, Tao Wang, Rob Johnson, and Ian Goldberg. A systematic approach to developing and evaluating website fingerprinting defenses. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, page 227–238, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450329576. doi: 10.1145/2660267.2660362. URL `https://doi.org/10.1145/2660267.2660362`.

[53] CAIDA. AS Classification. URL `http://www.caida.org/data/as-classification/`. Accessed 2017.

[54] S. Castillo-Perez and J. Garcia-Alfaro. Evaluation of two privacy-preserving protocols for the DNS. In *2009 Sixth International Conference on Information Technology: New Generations*, pages 411–416, April 2009. doi: 10.1109/ITNG.2009.195.

[55] Sergio Castillo-Perez and Joaquin Garcia-Alfaro. Anonymous resolution of DNS queries. In *On the Move to Meaningful Internet Systems: OTM 2008*, pages 987–1000, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. ISBN 978-3-540-88873-4.

[56] Censored Planet. Satellite. URL `http://www.censoredplanet.com/projects/satellite`. accessed 2018.

[57] Censored Planet: Satellite and Iris. URL `https://censoredplanet.org/projects/satellite`.

[58] Center for Applied Internet Data Analysis. Inferred AS to Organization Mapping Dataset. Web page, Accessed 2020. URL `https://www.caida.org/data/as-organizations/`.

[59] Center for Applied Internet Data Analysis. Routeviews Prefix to AS mappings Dataset for IPv4 and IPv6 . Web page, Accessed 2020. URL `http://www.caida.org/data/routing/routeviews-prefix2as.xml`.

[60] Abdelberi Chaabane, Terence Chen, Mathieu Cunche, Emiliano De Cristofaro, Arik Friedman, and Mohamed Ali Kaafar. Censorship in the wild: Analyzing Internet filtering in Syria. In *Internet Measurement Conference*. ACM, 2014. URL `http://conferences2.sigcomm.org/imc/2014/papers/p285.pdf`. `http://conferences2.sigcomm.org/imc/2014/papers/p285.pdf`.

[61] Abdelberi Chaabane, Terence Chen, Mathieu Cunche, Emiliano De Cristofaro, Arik Friedman, and Mohamed Ali Kaafar. Censorship in the wild: Analyzing Internet filtering in Syria. In *Internet Measurement Conference*, pages 285–298. ACM, 2014.

[62] Zimo Chai, Amirhossein Ghafari, and Amir Houmansadr. On the importance of encrypted-SNI (ESNI) to censorship circumvention. In *Free and Open Communications on the Internet*. USENIX, 2019. URL `https://www.usenix.org/system/files/foci19-paper_chai_update.pdf`. `https://www.usenix.org/system/files/foci19-paper_chai_update.pdf`.

[63] Zimo Chai, Amirhossein Ghafari, and Amir Houmansadr. On the importance of encrypted-sni (ESNI) to censorship circumvention. In *9th USENIX Workshop on Free and Open Communications on the Internet (FOCI 19)*, Santa Clara, CA, 2019. USENIX Association.

[64] Michael S. Chase and James Mulvenon. You've got dissent!: Chinese dissident use of the internet and beijing's counter-strategies. *Foreign Affairs*, 81:188, 2002.

[65] Taejoong Chung, David Choffnes, and Alan Mislove. Tunneling for transparency: A large-scale analysis of end-to-end violations in the internet. In *Proceedings of the 2016 Internet Measurement Conference*, IMC '16, pages 199–213, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4526-2. doi: 10.1145/2987443.2987455. URL `http://doi.acm.org/10.1145/2987443.2987455`.

[66] Taejoong Chung, Roland van Rijswijk-Deij, Balakrishnan Chandrasekaran, David Choffnes, Dave Levin, Bruce M. Maggs, Alan Mislove, and Christo Wilson. A longitudinal, end-to-end view of the DNSSEC ecosystem. In *26th USENIX Security Symposium*, pages 1307–1322, Vancouver, BC, 2017. USENIX Association. ISBN 978-1-931971-40-9.

[67] Citizen Lab. URL testing lists intended for discovering website censorship, 2014. URL `https://github.com/citizenlab/test-lists`.

[68] Citizen Lab. Collection of censorship blockpages, 2015. URL `https://github.com/citizenlab/blockpages`.

[69] Richard Clayton, Steven J. Murdoch, and Robert N. M. Watson. Ignoring the Great Firewall of China. In *Privacy Enhancing Technologies*, pages 20–35. Springer, 2006. doi: 10.1007/11957454_2.

[70] Lorenzo Colitti, Steinar H. Gunderson, Erik Kline, and Tiziana Refice. Evaluating IPv6 adoption in the internet. In *Passive and Active Measurement*, pages 141–150, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-12334-4.

[71] Craig Hockenberry. Fear China. URL `https://furbo.org/2015/01/22/fear-china`.

[72] Jedidiah R. Crandall, Daniel Zinn, Michael Byrd, Earl Barr, and Rich East. ConceptDoppler: A Weather Tracker for Internet Censorship. In *Computer and Communications Security*, pages 352–365. ACM, 2007. doi: 10.1145/1315245.1315290. URL `http://www.cs.unm.edu/~crandall/concept_doppler_ccs07.pdf`.

[73] Weiqi Cui, Tao Chen, Christian Fields, Julianna Chen, Anthony Sierra, and Eric Chan-Tin. Revisiting assumptions for website fingerprinting attacks. In *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, Asia CCS '19, page 328–339, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367523. doi: 10.1145/3321705.3329802. URL `https://doi.org/10.1145/3321705.3329802`.

[74] Jakub Czyz, Mark Allman, Jing Zhang, Scott Iekel-Johnson, Eric Osterweil, and Michael Bailey. Measuring IPv6 adoption. In *Proceedings of the 2014 ACM Conference on SIGCOMM*, SIGCOMM '14, pages 87–98, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2836-4.

[75] Tianxiang Dai, Haya Shulman, and Michael Waidner. DNSSEC misconfigurations in popular domains. In *Cryptology and Network Security*, pages 651–660. Springer, 2016. ISBN 978-3-319-48965-0.

[76] Alberto Dainotti, Claudio Squarcella, Emile Aben, Kimberly C. Claffy, Marco Chiesa, Michele Russo, and Antonio Pescape. Analysis of Country-Wide Internet Outages Caused by Censorship. *Transactions on Networking*, 22:1964–1977, 2013. ISSN 1063-6692. doi: 10.1109/TNET.2013.2291244. URL `http://www.caida.org/publications/papers/2014/outages_censorship/`.

[77] Jakub Dalek, Bennett Haselton, Helmi Noman, Adam Senft, Masashi Crete-Nishihata, Phillipa Gill, and Ronald J. Deibert. A Method for Identifying and Confirming the Use of URL Filtering Products for Censorship. In *Internet Measurement Conference*, pages 23–30. ACM, 2013. doi: 10.1145/2504730.2504763. URL `http://www3.cs.stonybrook.edu/~phillipa/papers/imc112s-dalek.pdf`.

[78] Jakub Dalek, Robert Deibert, Sarah McKune, Phillipa Gill, Adam Senft, and Naser Noor. Information controls during military operations: The case of Yemen during the 2015 political and armed conflict. Citizen lab, 2015. URL `https://citizenlab.ca/2015/10/information-controls-military-operations-yemen/`.

[79] Alexander Darer, Oliver Farnan, and Joss Wright. FilteredWeb: A Framework for the Automated Search-Based Discovery of Blocked URLs. In *Traffic Measurement and Analysis*, pages 1–9. IEEE, 2017. doi: 10.23919/TMA.2017.8002914.

[80] Ronald Deibert. China's Cyberspace Control Strategy: An Overview and Consideration of Issues for Canadian Policy, 2010.

[81] Ronald Deibert, John Palfrey, Rafal Rohozinski, and Jonathan Zittrain, editors. *Access Controlled: The Shaping of Power, Rights, and Rule in Cyberspace*. MIT Press, 2010.

[82] Ronald J. Deibert. Dark Guests and Great Firewalls: The Internet and Chinese Security Policy. *Journal of Social Issues*, 58:143–159, 2002.

[83] Matteo Dell'Amico, Leyla Bilge, Ashwin Kayyoor, Petros Efstathopoulos, and Pierre-Antoine Vervier. Lean on me: Mining internet service dependencies from large-scale DNS data. In *Proceedings of the 33rd Annual Computer Security Applications Conference*, ACSAC 2017, pages 449–460, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5345-8.

[84] T. Dierks and C. Allen. The TLS protocol version 1.0. RFC 2246, IETF, January 1999. URL `https://tools.ietf.org/html/rfc2246`.

[85] Marcel Dischinger, Massimiliano Marcon, Saikat Guha, Krishna P. Gummadi, Ratul Mahajan, and Stefan Saroiu. Glasnost: Enabling end users to detect traffic differentiation. 2010.

[86] Hai-Xin Duan, Nicholas Weaver, Zengzhi Zhao, Meng Hu, Jinjin Liang, Jian Jiang, Kang Li, and Vern Paxson. Hold-On: Protecting Against On-Path DNS Poisoning. In *the Conference on Securing and Trusting Internet Names*, 2012.

[87] Haixin Duan, Nicholas Weaver, Zongxu Zhao, Meng Hu, Jinjin Liang, Jian Jiang, Kang Li, and Vern Paxson. Hold-On: Protecting against on-path DNS poisoning. In *Securing and Trusting Internet Names*. National Physical Laboratory, 2012. URL `http://conferences.npl.co.uk/satin/papers/satin2012-Duan.pdf`. `http://conferences.npl.co.uk/satin/papers/satin2012-Duan.pdf`.

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

[89] Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. ZMap: Fast Internet-wide Scanning and Its Security Applications. In *Presented as part of the 22nd USENIX Security Symposium (USENIX Security 13)*, pages 605–620, Washington, D.C., 2013. USENIX. ISBN 978-1-931971-03-4. URL `https://www.usenix.org/conference/usenixsecurity13/technical-sessions/paper/durumeric`.

[90] Zakir Durumeric, David Adrian, Ariana Mirian, Michael Bailey, and J. Alex Halderman. A search engine backed by Internet-wide scanning. In *Computer and Communications Security*, pages 542–553. ACM, 2015. doi: 10.1145/2810103.2813703.

[91] K. P. Dyer, S. E. Coull, T. Ristenpart, and T. Shrimpton. Peek-a-Boo, I Still See You: Why Efficient Traffic Analysis Countermeasures Fail. In *Proceedings of the IEEE Symposium on Security & Privacy*, 2012.

[92] D. Eastlake and C. Kaufman. Domain name system security extensions. RFC 2065, IETF, January 1997. URL `https://tools.ietf.org/html/rfc2065`.

[93] H. Eidnes, G. de Groot, and P. Vixie. Classless IN-ADDR.ARPA delegation. RFC 2317, IETF, March 1998. URL `https://www.ietf.org/rfc/rfc2317`.

[94] Mark Emem. Monero Cryptomining Attack Affects Over 200,000 ISP-Grade Routers Globally. *CCN Markets*. URL `https://tinyurl.com/cnn-cryptomining`. Accessed 2018.

[95] Let's Encrypt. Let's Encrypt Stats, 2019. URL `https://letsencrypt.org/stats/`.

[96] Roya Ensafi, Philipp Winter, Abdullah Mueen, and Jedidiah R Crandall. Analyzing the Great Firewall of China over space and time. *PETs '15*.

[97] Roya Ensafi, Jeffrey Knockel, Geoffrey Alexander, and Jedidiah R. Crandall. Detecting Intentional Packet Drops on the Internet via TCP/IP Side Channels. In *Passive and Active Measurement*, pages 109–118. Springer, 2014. doi: 10.1007/978-3-319-04918-2_11.

[98] Roya Ensafi, Philipp Winter, Abdullah Mueen, and Jedidiah R. Crandall. Large-scale Spatiotemporal Characterization of Inconsistencies in the World's Largest Firewall. 2014.

[99] Roya Ensafi, David Fifield, Philipp Winter, Nick Feamster, Nicholas Weaver, and Vern Paxson. Examining How the Great Firewall Discovers Hidden Circumvention Servers. In *Internet Measurement Conference*, pages 445–458. ACM, 2015. doi: 10.1145/2815675.2815690.

[100] Oliver Farnan, Alexander Darer, and Joss Wright. Poisoning the Well: Exploring the Great Firewall's Poisoned DNS Responses. In *Privacy in the Electronic Society*, pages 95–98. ACM, 2016. doi: 10.1145/2994620.2994636.

[101] FCC. Protecting and promoting the open internet. `https://www.federalregister.gov/articles/2015/04/13/2015-07841/protecting-and-promoting-the-open-internet`, April 2015.

[102] David Fifield, Chang Lan, Rod Hynes, Percy Wegmann, and Vern Paxson. Blocking-resistant communication through domain fronting. *Proceedings on Privacy Enhancing Technologies*, 2015(2):46–64, 2015.

[103] Arturo Filasto and Jacob Appelbaum. Ooni: Open observatory of network interference. In *FOCI*, 2012.

[104] Tobias Flach, Pavlos Papageorge, Andreas Terzis, Luis Pedrosa, Yuchung Cheng, Tayeb Karim, Ethan Katz-Bassett, and Ramesh Govindan. An internet-wide analysis of traffic policing. ACM, 2016.

[105] Freedom House. Freedom on the Net 2017, . URL https://freedomhouse.org/report/freedom-net/freedom-net-2017. Accessed 2018.

[106] Freedom House. Country Profiles 2017: Turkey, . URL https://freedomhouse.org/report/freedom-net/2017/turkey. Accessed 2018.

[107] Freedom House. Freedom on the Net 2018: The Rise of Digital Authoritarianism, 2018. URL https://freedomhouse.org/report/freedom-net/2018/rise-digital-authoritarianism.

[108] Christian Fuchs, Kees Boersma, Anders Albrechtslund, and Marisol Sandoval. Internet and Surveillance: The Challenges of Web 2.0 and Social Media. 2011.

[109] V. Fuller and T. Li. Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan. RFC 4632, IETF, August 2006. URL https://tools.ietf.org/html/rfc4632.

[110] Evgeniy Gabrilovich and Alex Gontmakher. The homograph attack. *Commun. ACM*, 45(2):128–, February 2002. ISSN 0001-0782. doi: 10.1145/503124.503156. URL http://doi.acm.org/10.1145/503124.503156.

[111] Genevieve Gebhart and Tadayoshi Kohno. Internet Censorship in Thailand: User Practices and Potential Threats. *EuroSP '17*.

[112] General Data Protection Regulation. General Data Protection Regulation (2016/679). *Official Journal of the European Union*, L 119:1–88, 2016. URL https://gdpr-info.eu/.

[113] Geosurf. Geosurf: Residential and data center proxy network. URL https://www.geosurf.com. Accessed 2018.

[114] Geremie R. Barme And Sang Ye. The Great Firewall of China, 1997-06-01. URL https://www.wired.com/1997/06/china-3/.

[115] gfwrev. 深入理解GFW：DNS污染, November 2009. URL https://gfwrev.blogspot.com/2009/11/gfwdns.html.

[116] Phillipa Gill, Masashi Crete-Nishihata, Jakub Dalek, Sharon Goldberg, Adam Senft, and Greg Wiseman. Characterizing Web Censorship Worldwide: Another Look at the OpenNet Initiative Data. *Transactions on the Web*, 9(1), 2015. doi: 10.1145/2700339.

[117] Phillipa Gill, Masashi Crete-Nishihata, Jakub Dalek, Sharon Goldberg, Adam Senft, and Greg Wiseman. Characterizing Web Censorship Worldwide: Another Look at the OpenNet Initiative Data. *Transactions on the Web*, 9(1), 2015. doi: 10.1145/2700339.

[118] Google. DNS-over-HTTPS, 2018. URL https://developers.google.com/speed/public-dns/docs/dns-over-https. Accessed: October 2018.

[119] Google. DNS-over-TLS, 2019. URL https://developers.google.com/speed/public-dns/docs/dns-over-tls. Accessed: March 2019.

[120] GreatFire Project. GFW Upgrade Fail - Visitors To Blocked Sites Redirected To Porn, . URL https://en.greatfire.org/blog/2015/jan/gfw-upgrade-fail-visitors-blocked-sites-redirected-porn.

[121] GreatFire Project. We Monitor and Challenge Internet Censorship in China, . URL https://greatfire.org.

[122] Olafur Guomundsson. Introducing DNS resolver, 1.1.1.1. https://blog.cloudflare.com/dns-resolver-1-1-1-1, 2018. Online; accessed September 2018.

[123] Shuai Hao, Yubao Zhang, Haining Wang, and Angelos Stavrou. End-Users Get Maneuvered: Empirical Analysis of Redirection Hijacking in Content Delivery Networks. In *27th USENIX Security Symposium*, pages 1129–1145, Baltimore, MD, 2018. USENIX Association. ISBN 978-1-931971-46-1.

[124] Jamie Hayes and George Danezis. k-fingerprinting: A Robust Scalable Website Fingerprinting Technique. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 1187–1203, Austin, TX, August 2016. USENIX Association. ISBN 978-1-931971-32-4. URL https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/hayes.

[125] Sebastian Hellmeier. The Dictator's Digital Toolkit: Explaining Variation in Internet Filtering in Authoritarian Regimes. *Politics & Policy*, 44(6):1158–1191, 2016. doi: 10.1111/polp.12189.

[126] Herdict. Herdict: help spot web blockages. URL https://www.herdict.org/. accessed 2018.

[127] Dominik Herrmann, Rolf Wendolsky, and Hannes Federrath. Website fingerprinting: Attacking popular privacy enhancing technologies with the multinomial naïve-bayes classifier. In *Proceedings of the 2009 ACM Workshop on Cloud Computing Security*, CCSW '09, page 31–42, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605587844. doi: 10.1145/1655008.1655013. URL https://doi.org/10.1145/1655008.1655013.

[128] Dominik Herrmann, Karl-Peter Fuchs, Jens Lindemann, and Hannes Federrath. EncDNS: A lightweight privacy-preserving name resolution service. In *Computer Security - ESORICS 2014*, pages 37–55. Springer, 2014. ISBN 978-3-319-11203-9.

[129] Nguyen Phong Hoang, Arian Akhavan Niaki, Nikita Borisov, Phillipa Gill, and Michalis Polychronakis. Assessing the Privacy Benefits of Domain Name Encryption. In *ACM ASIACCS 2020*. URL `https://arxiv.org/pdf/1911.00563.pdf`.

[130] Nguyen Phong Hoang, Yasuhito Asano, and Masatoshi Yoshikawa. Your neighbors are my spies: Location and other privacy concerns in glbt-focused location-based dating applications. *Transactions on Advanced Communications Technology (TACT)*, 5(3):851–860, May 2016. doi: 10.23919/ICACT.2017.7890236.

[131] Nguyen Phong Hoang, Panagiotis Kintis, Manos Antonakakis, and Michalis Polychronakis. An Empirical Study of the I2P Anonymity Network and Its Censorship Resistance. In *Proceedings of the Internet Measurement Conference 2018*, IMC '18, pages 379–392, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5619-0.

[132] Nguyen Phong Hoang, Sadie Doreen, and Michalis Polychronakis. Measuring I2P censorship at a global scale. In *Free and Open Communications on the Internet*. USENIX, 2019. URL `https://www.usenix.org/system/files/foci19-paper_hoang.pdf`. `https://www.usenix.org/system/files/foci19-paper_hoang.pdf`.

[133] Nguyen Phong Hoang, Sadie Doreen, and Michalis Polychronakis. Measuring I2P Censorship at a Global Scale. In *9th USENIX Workshop on Free and Open Communications on the Internet (FOCI 19)*, Santa Clara, CA, 2019. USENIX Association.

[134] Nguyen Phong Hoang, Ivan Lin, Seyedhamed Ghavamnia, and Michalis Polychronakis. K-resolver: Towards Decentralizing Encrypted DNS Resolution. In *Proceedings of The NDSS Workshop on Measurements, Attacks, and Defenses for the Web 2020*, MADWeb '20. Internet Society, Jan 2020.

[135] Nguyen Phong Hoang, Arian Akhavan Niaki, Michalis Polychronakis, and Phillipa Gill. The web is still small after more than a decade: A revisit study of web co-location. *SIGCOMM Comput. Commun. Rev.*, 2020.

[136] Nguyen Phong Hoang, Arian Akhavan Niaki, Jakub Dalek, Jeffrey Knockel, Pellaeon Lin, Bill Marczak, Masashi Crete-Nishihata, Phillipa Gill, and Michalis Polychronakis. How great is the great firewall? measuring china's DNS censorship. USENIX, 2021. URL `https://www.usenix.org/system/files/sec21-hoang.pdf`. `https://www.usenix.org/system/files/sec21-hoang.pdf`.

[137] P. Hoffman and P. McManus. DNS queries over HTTPS (DoH). RFC 8484, IETF, October 2018. URL `https://tools.ietf.org/html/rfc8484`.

[138] Hola. Hola!VPN: Access any website. URL `https://hola.org/`. Accessed 2018.

[139] Austin Hounsel, Kevin Borgolte, Paul Schmitt, Jordan Holland, and Nick Feamster. Analyzing the Costs (and Benefits) of DNS, DoT, and DoH for the Modern Web. In *Proceedings of the Applied Networking Research Workshop*, ANRW '19, pages 20–22, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-6848-3. doi: 10.1145/3340301.3341129. URL http://doi.acm.org/10.1145/3340301.3341129.

[140] Z. Hu, L. Zhu, J. Heidemann, A. Mankin, D. Wessels, and P. Hoffman. Specification for DNS over transport layer security (TLS). RFC 7858, IETF, May 2016. URL https://tools.ietf.org/html/rfc7858.

[141] Huawei. Transport layer security (TLS) extensions: Server name indication. RFC 6066, IETF, January 2011. URL https://tools.ietf.org/html/rfc6066#section-3.

[142] Christian Huitema. Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs). RFC 4380, IETF, February 2006. URL https://tools.ietf.org/html/rfc4380.

[143] J. Ullrich. Are You Piratebay? thepiratebay.org Resolving to Various Hosts. URL https://isc.sans.edu/forums/diary/Are+You+Piratebay+thepiratebayorg+Resolving+to+Various+Hosts/19175.

[144] Ben Jones, Tzu-Wen Lee, Nick Feamster, and Phillipa Gill. Automated Detection and Fingerprinting of Censorship Block Pages. In *Internet Measurement Conference*, pages 299–304. ACM, 2014. doi: 10.1145/2663716.2663722.

[145] Frank J. Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American Statistical Association*, 46(253), 1951.

[146] Arash Molavi Kakhki, Fangfan Li, David Choffnes, Ethan Katz-Bassett, and Alan Mislove. Bingeon under the microscope: Understanding t-mobiles zerorating implementation. In *Proceedings of the 2016 workshop on QoE-based Analysis and Management of Data Communication Networks*, pages 43–48, 2016.

[147] Partha Kanuparthy and Constantine Dovrolis. Diffprobe: detecting isp service discrimination. IEEE, 2010.

[148] Partha Kanuparthy and Constantine Dovrolis. ShaperProbe: end-to-end detection of ISP traffic shaping using active methods. 2011.

[149] Simon Kenin. Mass MikroTik Router Infection – First we cryptojack Brazil, then we take the World? *SpiderLabs Blog*. URL https://tinyurl.com/mass-mikrotik-router-infection. Accessed 2018.

[150] M. T. Khan, X. Huo, Z. Li, and C. Kanich. Every Second Counts: Quantifying the Negative Externalities of Cybercrime via Typosquatting. In *2015 IEEE Symposium on Security and Privacy*, pages 135–150, May 2015. doi: 10.1109/SP.2015.16.

[151] Mohammad Taha Khan, Joe DeBlasio, Geoffrey M. Voelker, Alex C. Snoeren, Chris Kanich, and Narseo Vallina-Rodriguez. An Empirical Analysis of the Commercial VPN Ecosystem. In *Internet Measurement Conference*, pages 443–456. ACM, 2018. doi: 10.1145/3278532.3278570.

[152] Sheharbano Khattak, Mobin Javed, Syed Ali Khayam, Zartash Afzal Uzmi, and Vern Paxson. A Look at the Consequences of Internet Censorship Through an ISP Lens. In *Internet Measurement Conference*, pages 271–284. ACM, 2014. doi: 10.1145/2663716.2663750.

[153] Rebecca Killick, Paul Fearnhead, and Idris A Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, (500), 2012.

[154] Panagiotis Kintis, Yacin Nadji, David Dagon, Michael Farrell, and Manos Antonakakis. Understanding the Privacy Implications of ECS. In *Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 343–353. Springer, 2016. ISBN 978-3-319-40667-1.

[155] Panagiotis Kintis, Najmeh Miramirkhani, Charles Lever, Yizheng Chen, Rosa Romero-Gómez, Nikolaos Pitropakis, Nick Nikiforakis, and Manos Antonakakis. Hiding in Plain Sight: A Longitudinal Study of Combosquatting Abuse. In *the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 569–586, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4946-8. doi: 10.1145/3133956.3134002. URL `http://doi.acm.org/10.1145/3133956.3134002`.

[156] Jeffrey Knockel, Jedidiah R Crandall, and Jared Saia. Three Researchers, Five Conjectures: An Empirical Analysis of TOM-Skype Censorship and Surveillance. In *Free and Open Communications on the Internet*, Berkeley, CA, 2011. USENIX. URL `http://static.usenix.org/events/foci11/tech/final_files/Knockel.pdf`.

[157] Radhesh Krishnan Konoth, Emanuele Vineti, Veelasha Moonsamy, Martina Lindorfer, Christopher Kruegel, Herbert Bos, and Giovanni Vigna. MineSweeper: An In-depth Look into Drive-by Cryptocurrency Mining and Its Defense. In *Computer and Communications Security*, pages 1714–1730. ACM, 2018. doi: 10.1145/3243734.3243858.

[158] Platon Kotzias, Abbas Razaghpanah, Johanna Amann, Kenneth G. Paterson, Narseo Vallina-Rodriguez, and Juan Caballero. Coming of Age: A Longitudinal Study of TLS Deployment. In *Proceedings of the Internet Measurement Conference 2018*, pages 415–428, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5619-0.

[159] Athanasios Kountouras, Panagiotis Kintis, Chaz Lever, Yizheng Chen, Yacin Nadji, David Dagon, Manos Antonakakis, and Rodney Joffe. Enabling Network Security Through Active DNS Datasets. In *Research in Attacks, Intrusions, and Defenses*, pages 188–208. Springer, 2016. ISBN 978-3-319-45719-2.

[160] Srinivas Krishnan and Fabian Monrose. An Empirical Study of the Performance, Security and Privacy Implications of Domain Name Prefetching. In *Proceedings of the 2011 IEEE/IFIP 41st International Conference on Dependable Systems&Networks*, DSN '11, pages 61–72, Washington, DC, USA, 2011. IEEE Computer Society. ISBN 978-1-4244-9232-9. doi: 10.1109/DSN.2011.5958207. URL `http://dx.doi.org/10.1109/DSN.2011.5958207`.

[161] Tobias Lauinger, Abdelberi Chaabane, Ahmet Salih Buyukkayhan, Kaan Onarlioglu, and William Robertson. Game of Registrars: An Empirical Analysis of Post-Expiration Domain Name Takeovers. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 865–880, Vancouver, BC, 2017. USENIX Association. ISBN 978-1-931971-40-9.

[162] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. In *Network and Distributed System Security Symposium*, 2019. doi: 10.14722/ndss.2019.23386.

[163] Fangfan Li, Arash Molavi Kakhki, David Choffnes, Phillipa Gill, and Alan Mislove. Classifiers unclassified: An efficient approach to revealing IP-traffic classification rules. 2016.

[164] Fangfan Li, Abbas Razaghpanah, Arash Molavi Kakhki, Arian Akhavan Niaki, David Choffnes, Phillipa Gill, and Alan Mislove. liberate, (n): A library for exposing (traffic-classification) rules and avoiding them efficiently. IMC '17, 2017.

[165] Fangfan Li, Arian Akhavan Niaki, David Choffnes, Phillipa Gill, and Alan Mislove. A large-scale analysis of deployed traffic differentiation practices. In *Proceedings of the ACM Special Interest Group on Data Communication*, pages 130–144. 2019.

[166] Marc Liberatore and Brian Neil Levine. Inferring the source of encrypted HTTP connections. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*, 2006.

[167] G. Lindberg. Anti-Spam Recommendations for SMTP MTAs. RFC 2505, IETF, March 1999. URL `https://tools.ietf.org/html/rfc2505`.

[168] Graham Lowe, Patrick Winters, and Michael L. Marcus. The great DNS wall of China. Technical report, New York University, 2007. URL `https://censorbib.nymity.ch/pdf/Lowe2007a.pdf`. `https://censorbib.nymity.ch/pdf/Lowe2007a.pdf`.

[169] Liming Lu, Ee-Chien Chang, and Mun Choon Chan. Website fingerprinting and identification using ordered feature sequences. In Dimitris Gritzalis, Bart Preneel, and Marianthi Theoharidou, editors, *Computer Security – ESORICS 2010*, pages 199–214, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-15497-3.

[170] Y. Lu and G. Tsudik. Towards Plugging Privacy Leaks in the Domain Name System. In *2010 IEEE Tenth International Conference on Peer-to-Peer Computing (P2P)*, pages 1–10, Aug 2010. doi: 10.1109/P2P.2010.5569976.

[171] Luminati. Luminati: largest business proxy service. URL `http://luminati.io`. Accessed 2018.

[172] Majestic. The Majestic Million. Web page, Accessed 2019. URL `https://majestic.com/reports/majestic-million`.

[173] B. Marczak, N. Weaver, J. Dalek, Roya Ensafi, D. Fifield, Sarah McKune, Arn Rey, J. Scott-Railton, Ronald J. Deibert, and V. Paxson. An Analysis of China's Great Cannon. In *USENIX FOCI '15*.

[174] MaxMind. MaxMind GeoLite2 Databases, 2019. URL `https://www.maxmind.com/`.

[175] Allison McDonald, Matthew Bernhard, Luke Valenta, Benjamin VanderSloot, Will Scott, Nick Sullivan, J. Alex Halderman, and Roya Ensafi. 403 Forbidden: A Global View of CDN Geoblocking. In *Internet Measurement Conference*, pages 218–230. ACM, 2018. doi: 10.1145/3278532.3278552.

[176] Xianghang Mi, Ying Liu, Xuan Feng, Xiaojing Liao, Baojun Liu, XiaoFeng Wang, Feng Qian, Zhou Li, Sumayah Alrwais, and Limin Sun. Resident Evil: Understanding Residential IP Proxy as a Dark Service. In *Symposium on Security and Privacy*, pages 170–186. IEEE, 2019. doi: 10.1109/SP.2019.00011. URL `https://mixianghang.github.io/pubs/rpaas.pdf`.

[177] P. Mockapetris. DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION. RFC 1035, IETF, November 1987. URL `https://tools.ietf.org/html/rfc1035`.

[178] Arash Molavi Kakhki, Abbas Razaghpanah, Anke Li, Hyungjoon Koo, Rajesh Golani, David Choffnes, Phillipa Gill, and Alan Mislove. Identifying Traffic Differentiation in Mobile Networks. In *Internet Measurement Conference*, pages 239–251. ACM, 2015. doi: 10.1145/2815675.2815691.

[179] Zubair Nabi. The anatomy of web censorship in Pakistan. In *Free and Open Communications on the Internet*. USENIX, 2013. URL `https://censorbib.nymity.ch/pdf/Nabi2013a.pdf`. `https://censorbib.nymity.ch/pdf/Nabi2013a.pdf`.

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

[181] Milad Nasr, Amir Houmansadr, and Arya Mazumdar. Compressive traffic analysis: A new paradigm for scalable traffic analysis. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, page 2053–2069, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349468. doi: 10.1145/3133956.3134074. URL `https://doi.org/10.1145/3133956.3134074`.

[182] Arian Akhavan Niaki, Shinyoung Cho, Zachary Weinberg, Nguyen Phong Hoang, Abbas Razaghpanah, Nicolas Christin, and Phillipa Gill. Iclab: A global, longitudinal internet censorship measurement platform. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 135–151. IEEE, 2020.

[183] Arian Akhavan Niaki, Nguyen Phong Hoang, Phillipa Gill, Amir Houmansadr, et al. Triplet censors: Demystifying great firewall's {DNS} censorship behavior. In *10th {USENIX} Workshop on Free and Open Communications on the Internet ({FOCI} 20)*, 2020.

[184] Nick Nikiforakis, Steven Van Acker, Wannes Meert, Lieven Desmet, Frank Piessens, and Wouter Joosen. Bitsquatting: Exploiting Bit-flips for Fun, or Profit? In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 989–998, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2035-1. doi: 10.1145/2488388.2488474.

[185] Nick Nikiforakis, Marco Balduzzi, Lieven Desmet, Frank Piessens, and Wouter Joosen. Soundsquatting: Uncovering the Use of Homophones in Domain Squatting. In *Information Security*, pages 291–308. Springer, 2014. ISBN 978-3-319-13257-0.

[186] Aqib Nisar, Aqsa Kashaf, Ihsan Ayyub Qazi, and Zartash Afzal Uzmi. Incentivizing censorship measurements via circumvention. In *SIGCOMM*, pages 533–546. ACM, 2018. doi: 10.1145/3230543.3230568.

[187] D Nobori and Y Shinjo. VPN Gate: A Volunteer-Organized Public VPN Relay System with Blocking Resistance for Bypassing Government Censorship Firewalls. *USENIX NSDI '14*.

[188] OONI. DNS consistency. URL `https://github.com/ooni/spec/blob/master/nettests/ts-002-dns-consistency.md`. Blog post, accessed 2018.

[189] OpenNet Initiative. Country Profiles: South Korea, 2012. URL `https://opennet.net/research/profiles/south-korea`. accessed 2019.

[190] Andriy Panchenko, Lukas Niessen, Andreas Zinnen, and Thomas Engel. Website fingerprinting in onion routing based anonymization networks. In *Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society*, WPES '11, page 103–114, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450310024. doi: 10.1145/2046556.2046570. URL `https://doi.org/10.1145/2046556.2046570`.

[191] Andriy Panchenko, Fabian Lanze, Jan Pennekamp, Thomas Engel, Andreas Zinnen, Martin Henze, and Klaus Wehrle. Website Fingerprinting at Internet Scale. In *Proceedings of NDSS '16*, 2016.

[192] Elkana Pariwono, Daiki Chiba, Mitsuaki Akiyama, and Tatsuya Mori. Don't throw me away: Threats caused by the abandoned internet resources used by android apps. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, ASIACCS '18, pages 147–158, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5576-6. doi: 10.1145/3196494.3196554. URL `http://doi.acm.org/10.1145/3196494.3196554`.

[193] Jong Chun Park and Jedidiah R. Crandall. Empirical study of a national-scale distributed intrusion detection system: Backbone-level filtering of HTML responses in China. In *Distributed Computing Systems*, pages 315–326. IEEE, 2010. doi: 10.1109/ICDCS.2010.46. URL `http://iar.cs.unm.edu/~crandall/icdcs2010.pdf`.

[194] Simran Patil and Nikita Borisov. What can you learn from an ip? In *Proceedings of the Applied Networking Research Workshop*, ANRW '19, pages 45–51, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-6848-3. doi: 10.1145/3340301.3341133. URL `http://doi.acm.org/10.1145/3340301.3341133`.

[195] Paul Pearce, Roya Ensafi, Frank Li, Nick Feamster, and Vern Paxson. Augur: Internet-Wide Detection of Connectivity Disruptions. In *Symposium on Security and Privacy*, pages 427–443. IEEE, 2017. doi: 10.1109/SP.2017.55.

[196] Paul Pearce, Ben Jones, Frank Li, Roya Ensafi, Nick Feamster, Nick Weaver, and Vern Paxson. Global Measurement of DNS Manipulation. In *USENIX Security Symposium*. USENIX, 2017.

[197] Peter Guess. China suddenly blocked an Indonesian newspaper. No one knows why. URL `https://restofworld.org/2021/china-suddenly-blocked-an-indonesian-newspaper-no-one-knows-why/`.

[198] Victor Le Pochat, Tom Van Goethem, and Wouter Joosen. Evaluating the Long-term Effects of Parameters on the Characteristics of the Tranco Top Sites Ranking. In *USENIX CSET '19*.

[199] Jon Postel. Transmission Control Protocol. RFC 793, 1981. URL `https://tools.ietf.org/html/rfc793`.

[200] Matthew Prince. Encrypting SNI: Fixing one of the core internet bugs. https://blog.cloudflare.com/esni/, 2018. Online; accessed September 2018.

[201] Thomas H Ptacek and Timothy N Newsham. Insertion, evasion, and denial of service: Eluding network intrusion detection. Technical report, Secure Networks inc Calgary Alberta, 1998. URL https://users.ece.cmu.edu/~adrian/731-sp04/readings/Ptacek-Newsham-ids98.pdf.

[202] quantcast. Quantcast top websites. Web page, Accessed 2019. URL https://www.quantcast.com/top-sites/.

[203] Florian Quinkert, Tobias Lauinger, William Robertson, Engin Kirda, and Thorsten Holz. It's not what it looks like: Measuring attacks and defensive registrations of homograph domains. In *2019 IEEE Conference on Communications and Network Security (CNS)*, 2019.

[204] R. Liao. China bans Scratch, MIT's programming language for kids, 2020. URL https://techcrunch.com/2020/09/07/scratch-ban-in-china.

[205] Ram Sundara Raman, Adrian Stoll, Jakub Dalek, Reethika Ramesh, Will Scott, and Roya Ensafi. Measuring the deployment of network censorship filters at global scale. In *NDSS '20*.

[206] Venugopalan Ramasubramanian and Emin Gün Sirer. Perils of transitive trust in the domain name system. In *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*, IMC '05, pages 35–35, Berkeley, CA, USA, 2005. USENIX Association. URL http://dl.acm.org/citation.cfm?id=1251086.1251121.

[207] R. Rambert, Z. Weinberg, D. Barradas, and N. Christin. Chinese Wall or Swiss Cheese? Keyword filtering in the Great Firewall of China. In *ACM WWW '21*.

[208] Rapid7. Rapid7: Open data. https://opendata.rapid7.com/, 2019.

[209] Fahmida Y. Rashid. Amazon joins google in shutting down doamin fronting. https://duo.com/decipher/amazon-joins-google-in-shutting-down-domain-fronting, 2018.

[210] Riccardo Ravaioli, Chadi Barakat, and Guillaume Urvoy-Keller. Chkdiff: checking traffic differentiation at internet access. In *Proc. of CoNEXT 2012 Student Workshop*. ACM, 2012.

[211] Riccardo Ravaioli, Guillaume Urvoy-Keller, and Chadi Barakat. Towards a general solution for detecting traffic differentiation at the internet access. In *Proc. of the 23rd International Teletraffic Congress (ITC)*. IEEE, 2015.

[212] Raymond Zhong, Paul Mozur, Jeff Kao, and Aaron Krolik. No 'Negative' News: How China Censored the Coronavirus. The New York Times, 2020-12-19. URL https://www.nytimes.com/2020/12/19/technology/china-coronavirus-censorship.html.

[213] Reporters Without Borders. 2018 world press freedom index. URL https://rsf.org/en/ranking/2018.

[214] E. Rescorla, K. Oku, N. Sullivan, and C. Wood. Encrypted Server Name Indication for TLS 1.3. Internet draft, IETF, March 2019. URL https://tools.ietf.org/html/draft-ietf-tls-esni-03.

[215] Philipp Richter, R. Padmanabhan, N. Spring, A. Berger, and D. Clark. Advancing the Art of Internet Edge Outage Detection. *ACM IMC '18*.

[216] RIPE-NCC. RIPE NCC AS Visibility Tool, Accessed 2020. URL https://stat.ripe.net.

[217] RIPE NCC Staff. RIPE Atlas: A Global Internet Measurement Network. *The Internet Protocol Journal*, 18(3):2–26, 2015. URL http://ipj.dreamhosters.com/wp-content/uploads/2015/10/ipj18.3.pdf.

[218] Prasanto K. Roy. India net neutrality rules could be world's strongest. *BBC News*. URL https://www.bbc.com/news/world-asia-india-42162979. Accessed 2018.

[219] Walter Rweyemamu, Christo Lauinger, Tobiasand Wilson, William Robertson, and Engin Kirda. Clustering and the Weekend Effect: Recommendations for the Use of Top Domain Lists in Security Research. In David Choffnes and Marinho Barcellos, editors, *Passive and Active Measurement*, pages 161–177. Springer International Publishing, 2019. ISBN 978-3-030-15986-3.

[220] Mahrud Sayrafi. Introducing DNS resolver for Tor. https://blog.cloudflare.com/welcome-hidden-resolver/, 2018. Online; accessed September 2018.

[221] Quirin Scheitle, Oliver Hohlfeld, Julien Gamba, Jonas Jelten, Torsten Zimmermann, Stephen D. Strowes, and Narseo Vallina-Rodriguez. A Long Way to the Top: Significance, Structure, and Stability of Internet Top Lists. In *Internet Measurement Conference*, pages 478–493. ACM, 2018. doi: 10.1145/3278532.3278574.

[222] Will Scott, Thomas Anderson, Tadayoshi Kohno, and Arvind Krishnamurthy. Satellite: Joint Analysis of CDNs and Network-Level Interference. In *USENIX Annual Technical Conference*. USENIX, 2016.

[223] Andreas Sfakianakis, Elias Athanasopoulos, and Sotiris Ioannidis. CensMon: A Web Censorship Monitor. In *Free and Open Communications on the Internet*. USENIX, 2011.

[224] Shawn Conaway. The Great Firewall: How China Polices Internet Traffic. Certification Magazine, 2009-09-30. URL http://certmag.com/the-great-firewall-how-china-polices-internet-traffic/.

[225] Craig A. Shue, Andrew J. Kalafut, and Minaxi Gupta. The Web is Smaller Than It Seems. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, IMC '07, pages 123–128, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-908-1. doi: 10.1145/1298306.1298324.

[226] Haya Shulman. Pretty bad privacy: Pitfalls of DNS encryption. In *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, WPES '14, pages 191–200, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-3148-7. doi: 10.1145/2665943.2665959. URL http://doi.acm.org/10.1145/2665943.2665959.

[227] Soutik Biswas. India-China Clash: 20 Indian Troops Killed in Ladakh Fighting. BBC, 2020-06-16. URL https://www.bbc.com/news/world-asia-53061476.

[228] Efe Kerem Sozeri. Inside Turkey's war on Wikipedia. *The Daily Dot*. URL https://www.dailydot.com/layer8/turkey-bans-wikipedia-censorship/. Accessed 2018.

[229] Sparks and Neo and Tank and Sminth and Dozer. The collateral damage of Internet censorship by DNS injection. *SIGCOMM Computer Communication Review*, 42(3):21–27, 2012. URL http://conferences.sigcomm.org/sigcomm/2012/paper/ccr-paper266.pdf.

[230] Ram Sundara Raman, Prerana Shenoy, Katharina Kohls, and Roya Ensafi. Censored planet: An internet-wide, longitudinal censorship observatory. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 49–66, 2020.

[231] Janos Szurdi, Balazs Kocso, Gabor Cseh, Jonathan Spring, Mark Felegyhazi, and Chris Kanich. The long "taile" of typosquatting domain names. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 191–206, San Diego, CA, 2014. USENIX Association. ISBN 978-1-931971-15-7. URL https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/szurdi.

[232] Mukarram Bin Tariq, Murtaza Motiwala, Nick Feamster, and Mostafa Ammar. Detecting network neutrality violations with causal inference. 2009.

[233] Team-Cymru. Team Cymru IP to ASN Mapping Service, Accessed 2020. URL https://team-cymru.com/community-services/ip-asn-mapping/.

[234] Telecom Regulatory Authority of India. Recommendations on Net Neutrality, 2017. URL https://main.trai.gov.in/sites/default/files/Recommendations_NN_2017_11_28.pdf.

[235] Andree Toonk. Turkey Hijacking IP addresses for popular Global DNS providers. URL https://bgpmon.net/turkey-hijacking-ip-addresses-for-popular-global-dns-providers/. 2014, accessed 2018.

[236] Michael Carl Tschantz, Sadia Afroz, Shaarif Sajid, Shoaib Asif Qazi, Mobin Javed, and Vern Paxson. A bestiary of blocking: The motivations and modes behind website unavailability. In *Free and Open Communications on the Internet*. USENIX, 2018.

[237] Cisco Umbrella. Umbrella popularity list. Web page, Accessed 2019. URL `https://s3-us-west-1.amazonaws.com/umbrella-static/index.html`.

[238] Pelayo Vallina, Victor Le Pochat, Álvaro Feal, M. Paraschiv, Julien Gamba, T. Burke, O. Hohlfeld, Juan Tapiador, and N. Vallina-Rodriguez. Mis-shapes, Mistakes, Misfits: An Analysis of Domain Classification Services. *ACM IMC '20*.

[239] Iljitsch van Beijnum. China censorship leaks outside Great Firewall via root server. *Ars Technica*. URL `https://arstechnica.com/tech-policy/2010/03/china-censorship-leaks-outside-great-firewall-via-root-server/`. Accessed 2018.

[240] Ben VanderSloot, Allison McDonald, Scott. Will, J. Alex Halderman, and Roya Ensafi. Quack: Scalable Remote Measurement of Application-Layer Censorship. In *USENIX Security Symposium*. USENIX, 2018.

[241] Verisign. The domain name industry brief. Technical report, Verisign, 2019. URL `https://www.verisign.com/assets/domain-name-report-Q12019.pdf`.

[242] John-Paul Verkamp and Minaxi Gupta. Inferring Mechanics of Web Censorship Around the World. In *Free and Open Communications on the Internet*, 2012.

[243] Matthäus Wander. Measurement survey of server-side DNSSEC adoption. In *Network Traffic Measurement and Analysis*. IEEE, 2017. URL `http://tma.ifip.org/wp-content/uploads/sites/7/2017/06/tma2017_paper58.pdf`.

[244] Huandong Wang, Fengli Xu, Yong Li, Pengyu Zhang, and Depeng Jin. Understanding mobile traffic patterns of large scale cellular towers in urban environment. 2015.

[245] Tao Wang, Xiang Cai, Rishab Nithyanand, Rob Johnson, and Ian Goldberg. Effective Attacks and Provable Defenses for Website Fingerprinting. In *Proceedings of the USENIX Security Symposium*, 2014.

[246] Z. Wang, Y. Cao, Z. Qian, C. Song, and S. Krishnamurthy. Your state is not mine: a closer look at evading stateful internet censorship. In *ACM IMC '17*.

[247] Nicholas Weaver, Robin Sommer, and Vern Paxson. Detecting Forged TCP Reset Packets. In *Network and Distributed System Security*. Internet Society, 2009.

[248] Nicholas Weaver, Christian Kreibich, Martin Dam, and Vern Paxson. Here Be Web Proxies. In *Passive and Active Measurement*, pages 183–192. Springer, 2014. doi: 10.1007/978-3-319-04918-2_18.

[249] Florian Weimer. Passive DNS replication. In *FIRST conference on computer security incident*, page 98, 2005.

[250] Zachary Weinberg, Mahmood Sharif, Janos Szurdi, and Nicolas Christin. Topics of Controversy: An Empirical Analysis of Web Censorship Lists. In *Privacy Enhancing Technologies*, pages 42–61. De Gruyter, 2017. doi: 10.1515/popets-2017-0004.

[251] Zachary Weinberg, Shinyoung Cho, Nicolas Christin, Vyas Sekar, and Phillipa Gill. How to Catch when Proxies Lie: Verifying the Physical Locations of Network Proxies with Active Geolocation. In *Internet Measurement Conference*, pages 203–217. ACM, 2018. doi: 10.1145/3278532.3278551.

[252] Udi Weinsberg, Augustin Soule, and Laurent Massoulie. Inferring traffic shaping and policy parameters using end host measurements. In *Proc. INFOCOM*. IEEE, 2011.

[253] P Winter and S Lindskog. How the Great Firewall of China is Blocking Tor. USENIX FOCI '12.

[254] Charles V Wright, Scott E Coull, and Fabian Monrose. Traffic morphing: An efficient defense against statistical traffic analysis. In *NDSS*, 2009.

[255] Joss Wright. Regional Variation in Chinese Internet Filtering. *Information, Communication & Society*, 17(1):121–141, 2014. doi: 10.1080/1369118X.2013.853818.

[256] Joss Wright, Alexander Darer, and Oliver Farnan. Filterprints: Identifying Localized Usage Anomalies in Censorship Circumvention Tools. 2016.

[257] Xueyang Xu, Z. Morley Mao, and J. Alex Halderman. Internet Censorship in China: Where Does the Filtering Occur? In *Passive and Active Measurement*, volume 6579 of *LNCS*, pages 133–142, Berlin, Heidelberg, 2011. Springer.

[258] Young Xu. Deconstructing the Great Firewall of China. Technical report, Thousand Eyes, 2016. URL `https://blog.thousandeyes.com/deconstructing-great-firewall-china/`.

[259] Tarun Kumar Yadav, Akshat Sinha, Devashish Gosain, Piyush Kumar Sharma, and Sambuddho Chakravarty. Where The Light Gets In: Analyzing Web Censorship Mechanisms in India. In *Internet Measurement Conference*, pages 252–264. ACM, 2018. doi: 10.1145/3278532.3278555.

[260] Boru Yan, Binxing Fang, Bin Li, and Yao Wang. DNS欺骗攻击的检测和防范. 计算机工程, 32(21):130–132, 2006. URL `https://tinyurl.com/web-archive-tomcat`.

[261] Ying Zhang, Z. Morley Mao, and Ming Zhang. Detecting Traffic Differentiation in Backbone ISPs with NetPolice. 2009.

[262] Fangming Zhao, Yoshiaki Hori, and Kouichi Sakurai. Two-servers PIR based DNS query scheme with privacy-preserving. In *Proceedings of the The 2007 International Conference on Intelligent Pervasive Computing*, IPC '07, pages 299–302, Washington, DC, USA, 2007. IEEE Computer Society. ISBN 0-7695-3006-0. doi: 10.1109/IPC.2007.107.

[263] Erkang Zhu, Fatemeh Nargesian, Ken Q. Pu, and Renée Miller. LSH Ensemble: Internet-Scale Domain Search. *Proceedings of the VLDB Endowment*, 9(12): 1185–1196, 2016.

[264] Pengxiong Zhu, Keyu Man, Zhongjie Wang, Zhiyun Qian, Roya Ensafi, J Alex Halderman, and Haixin Duan. Characterizing transnational internet performance and the great bottleneck of china. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 4(1):1–23, 2020.

[265] George Kingsley Zipf. Relative frequency as a determinant of phonetic change. *Harvard studies in classical philology*, 40:1–95, 1929.

[266] Jonathan Zittrain and Benjamin Edelman. Internet filtering in china. *IEEE Internet Computing*, 7(2):70–77, 2003.