

Workgroup: pearg
Internet-Draft: draft-irtf-pearg-censorship-06
Published: 25 May 2022
Intended Status: Informational
Expires: 26 November 2022
Authors: J.L. Hall M.D. Aaron A. Andersdotter
 Internet Society CU Boulder
 B. Jones N. Feamster
 Princeton U Chicago
 M. Knodel
 Center for Democracy & Technology
 A Survey of Worldwide Censorship Techniques

Abstract

This document describes technical mechanisms employed in network censorship that regimes around the world use for blocking or impairing Internet traffic. It aims to make designers, implementers, and users of Internet protocols aware of the properties exploited and mechanisms used for censoring end-user access to information. This document makes no suggestions on individual protocol considerations, and is purely informational, intended as a reference.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 26 November 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

- [1. Introduction](#)
- [2. Terminology](#)
- [3. Technical Prescription](#)
- [4. Technical Identification](#)
 - [4.1. Points of Control](#)
 - [4.2. Application Layer](#)
 - [4.2.1. HTTP Request Header Identification](#)
 - [4.2.2. HTTP Response Header Identification](#)
 - [4.2.3. Transport Layer Security \(TLS\)](#)
 - [4.2.4. Instrumenting Content Distributors](#)
 - [4.2.5. Deep Packet Inspection \(DPI\) Identification](#)
 - [4.3. Transport Layer](#)
 - [4.3.1. Shallow Packet Inspection and Transport Header Identification](#)
 - [4.3.2. Protocol Identification](#)
 - [4.4. Residual Censorship](#)
- [5. Technical Interference](#)
 - [5.1. Application Layer](#)
 - [5.1.1. DNS Interference](#)
 - [5.2. Transport Layer](#)
 - [5.2.1. Performance Degradation](#)
 - [5.2.2. Packet Dropping](#)
 - [5.2.3. RST Packet Injection](#)
 - [5.3. Routing Layer](#)
 - [5.3.1. Network Disconnection](#)
 - [5.3.2. Adversarial Route Announcement](#)
 - [5.4. Multi-layer and Non-layer](#)
 - [5.4.1. Distributed Denial of Service \(DDoS\)](#)
 - [5.4.2. Censorship in Depth](#)
- [6. Non-Technical Interference](#)
 - [6.1. Manual Filtering](#)
 - [6.2. Self-Censorship](#)
 - [6.3. Server Takedown](#)
 - [6.4. Notice and Takedown](#)
 - [6.5. Domain-Name Seizures](#)
- [7. Future work](#)
- [8. Contributors](#)
- [9. Informative References](#)
- [Authors' Addresses](#)

1. Introduction

Censorship is where an entity in a position of power - such as a government, organization, or individual - suppresses communication that it considers objectionable, harmful, sensitive, politically incorrect or inconvenient [[WP-Def-2020](#)]. Although censors that engage in censorship must do so through legal, military, or other means, this document focuses largely on technical mechanisms used to achieve network censorship.

This document describes technical mechanisms that censorship regimes around the world use for blocking or impairing Internet traffic. See [[RFC7754](#)] for a discussion of Internet blocking and filtering in terms of implications for Internet architecture, rather than end-user access to content and services. There is also a growing field of academic study of censorship circumvention (see the review article of [[Tschantz-2016](#)]), results from which we seek to make relevant here for protocol designers and implementers.

Censorship circumvention also impacts the cost of implementation of a censorship measure and we include mentions of tradeoffs in relation to such costs in conjunction with each technical method identified below.

2. Terminology

We describe three elements of Internet censorship: prescription, identification, and interference. The document contains three major sections, each corresponding to one of these elements. Prescription is the process by which censors determine what types of material they should censor, e.g., classifying pornographic websites as undesirable. Identification is the process by which censors classify specific traffic or traffic identifiers to be blocked or impaired, e.g., deciding that webpages containing "sex" in an HTTP Header or that accept traffic through the URL www.sex.example are likely to be undesirable. Interference is the process by which censors intercede in communication and prevents access to censored materials by blocking access or impairing the connection, e.g., implementing a technical solution capable of identifying HTTP headers or URLs and ensuring they are rendered wholly or partially inaccessible.

3. Technical Prescription

Prescription is the process of figuring out what censors would like to block [[Glanville-2008](#)]. Generally, censors aggregate information "to block" in blocklists or use real-time heuristic assessment of content [[Ding-1999](#)]. Some national networks are designed to more naturally serve as points of control [[Leyba-2019](#)]. There are also indications that online censors use probabilistic machine learning

techniques [[Tang-2016](#)]. Indeed, web crawling and machine learning techniques are an active research idea in the effort to identify content deemed as morally or commercially harmful to companies or consumers in some jurisdictions [[SIDN2020](#)].

There are typically a few types of blocklist elements: Keyword, domain name, protocol, or Internet Protocol (IP) address. Keyword and domain name blocking take place at the application level, e.g., HTTP; protocol blocking often occurs using Deep Packet Inspection to identify a forbidden protocol; IP blocking tends to take place using IP addresses in IPv4/IPv6 headers. Some censors also use the presence of certain keywords to enable more aggressive blocklists [[Rambert-2021](#)] or to be more permissive with content [[Knockel-2021](#)].

The mechanisms for building up these blocklists vary. Censors can purchase from private industry "content control" software, such as SmartFilter, which lets censors filter traffic from broad categories they would like to block, such as gambling or pornography [[Knight-2005](#)]. In these cases, these private services attempt to categorize every semi-questionable website as to allow for meta-tag blocking. Similarly, they tune real-time content heuristic systems to map their assessments onto categories of objectionable content.

Countries that are more interested in retaining specific political control typically have ministries or organizations that maintain blocklists. Examples include the Ministry of Industry and Information Technology in China, Ministry of Culture and Islamic Guidance in Iran, and specific to copyright in France [[HADOPI-2020](#)] and across the EU for consumer protection law [[Reda-2017](#)].

4. Technical Identification

4.1. Points of Control

Internet censorship takes place in all parts of the network topology. It may be implemented in the network itself (e.g. local loop or backhaul), on the services side of communication (e.g. web hosts, cloud providers or content delivery networks), in the ancillary services eco-system (e.g. domain name system or certificate authorities) or on the end-client side (e.g. in an end-user device such as a smartphone, laptop or desktop or software executed on such devices). An important aspect of pervasive technical interception is the necessity to rely on software or hardware to intercept the content the censor is interested in. There are various logical and physical points-of-control censors may use

for interception mechanisms, including, though not limited to, the following.

*Internet Backbone: If a censor controls the gateways into a region, they can filter undesirable traffic that is traveling into and out of the region by packet sniffing and port mirroring at the relevant exchange points. Censorship at this point of control is most effective at controlling the flow of information between a region and the rest of the Internet, but is ineffective at identifying content traveling between the users within a region. Some national network designs naturally serve as more effective chokepoints and points of control [[Leyba-2019](#)].

*Internet Service Providers: Internet Service Providers are frequently exploited points of control. They have the benefit of being easily enumerable by a censor - often falling under the jurisdictional or operational control of a censor in an indisputable way - with the additional feature that an ISP can identify the regional and international traffic of all their users. The censor's filtration mechanisms can be placed on an ISP via governmental mandates, ownership, or voluntary/coercive influence.

*Institutions: Private institutions such as corporations, schools, and Internet cafes can use filtration mechanisms. These mechanisms are occasionally at the request of a government censor, but can also be implemented to help achieve institutional goals, such as fostering a particular moral outlook on life by school-children, independent of broader society or government goals.

*Content Distribution Networks (CDNs): CDNs seek to collapse network topology in order to better locate content closer to the service's users. This reduces content transmission latency and improves quality of service. The CDN service's content servers, located "close" to the user in a network-sense, can be powerful points of control for censors, especially if the location of CDN content repositories allow for easier interference.

*Certificate Authorities (CAs) for Public-Key Infrastructures (PKIs): Authorities that issue cryptographically secured resources can be a significant point of control. CAs that issue certificates to domain holders for TLS/HTTPS (the Web PKI) or Regional/Local Internet Registries (RIRs) that issue Route Origination Authorizations (ROAs) to BGP operators can be forced to issue rogue certificates that may allow compromise, i.e., by allowing censorship software to engage in identification and interference where not possible before. CAs may also be forced to revoke certificates. This may lead to adversarial traffic routing

or TLS interception being allowed, or an otherwise rightful origin or destination point of traffic flows being unable to communicate in a secure way.

*Services: Application service providers can be pressured, coerced, or legally required to censor specific content or data flows. Service providers naturally face incentives to maximize their potential customer base and potential service shutdowns or legal liability due to censorship efforts may seem much less attractive than potentially excluding content, users, or uses of their service. Services have increasingly become focal points of censorship discussions, as well as the focus of discussions of moral imperatives to use censorship tools.

*Content sites: On the service side of communications lie many platforms that publish user-generated content require terms of service compliance with all content and user accounts in order to avoid intermediary liability for the web hosts. In aggregate these policies, actions and remedies are known as content moderation. Content moderation happens above the services or application layer, but these mechanisms are built to filter, sort and block content and users thus making them available to censors through direct pressure on the private entity.

*Personal Devices: Censors can mandate censorship software be installed on the device level. This has many disadvantages in terms of scalability, ease-of-circumvention, and operating system requirements. (Of course, if a personal device is treated with censorship software before sale and this software is difficult to reconfigure, this may work in favor of those seeking to control information, say for children, students, customers, or employees.) The emergence of mobile devices exacerbate these feasibility problems. This software can also be mandated by institutional actors acting on non-governmentally mandated moral imperatives.

At all levels of the network hierarchy, the filtration mechanisms used to censor undesirable traffic are essentially the same: a censor either directly identifies undesirable content using the identifiers described below and then uses a blocking or shaping mechanism such as the ones exemplified below to prevent or impair access, or requests that an actor ancillary to the censor, such as a private entity, perform these functions. Identification of undesirable traffic can occur at the application, transport, or network layer of the IP stack. Censors often focus on web traffic, so the relevant protocols tend to be filtered in predictable ways (see [Section 4.2.1](#) and [Section 4.2.2](#)). For example, a subversive image might make it past a keyword filter. However, if later the

image is deemed undesirable, a censor may then blacklist the provider site's IP address.

4.2. Application Layer

The following subsections describe properties and tradeoffs of common ways in which censors filter using application-layer information. Each subsection includes empirical examples describing these common behaviors for further reference.

4.2.1. HTTP Request Header Identification

An HTTP header contains a lot of useful information for traffic identification. Although "host" is the only required field in an HTTP request header (for HTTP/1.1 and later), an HTTP method field is necessary to do anything useful. As such, "method" and "host" are the two fields used most often for ubiquitous censorship. A censor can sniff traffic and identify a specific domain name (host) and usually a page name (GET /page) as well. This identification technique is usually paired with transport header identification (see [Section 4.3.1](#)) for a more robust method.

Tradeoffs: Request Identification is a technically straight-forward identification method that can be easily implemented at the Backbone or ISP level. The hardware needed for this sort of identification is cheap and easy-to-acquire, making it desirable when budget and scope are a concern. HTTPS will encrypt the relevant request and response fields, so pairing with transport identification (see [Section 4.3.1](#)) is necessary for HTTPS filtering. However, some countermeasures can trivially defeat simple forms of HTTP Request Header Identification. For example, two cooperating endpoints - an instrumented web server and client - could encrypt or otherwise obfuscate the "host" header in a request, potentially thwarting techniques that match against "host" header values.

Empirical Examples: Studies exploring censorship mechanisms have found evidence of HTTP header/ URL filtering in many countries, including Bangladesh, Bahrain, China, India, Iran, Malaysia, Pakistan, Russia, Saudi Arabia, South Korea, Thailand, and Turkey [[Verkamp-2012](#)] [[Nabi-2013](#)] [[Aryan-2012](#)]. Commercial technologies such as the McAfee SmartFilter and NetSweeper are often purchased by censors [[Dalek-2013](#)]. These commercial technologies use a combination of HTTP Request Identification and Transport Header Identification to filter specific URLs. Dalek et al. and Jones et al. identified the use of these products in the wild [[Dalek-2013](#)] [[Jones-2014](#)].

4.2.2. HTTP Response Header Identification

While HTTP Request Header Identification relies on the information contained in the HTTP request from client to server, response identification uses information sent in response by the server to client to identify undesirable content.

Tradeoffs: As with HTTP Request Header Identification, the techniques used to identify HTTP traffic are well-known, cheap, and relatively easy to implement. However, they are made useless by HTTPS because HTTPS encrypts the response and its headers.

The response fields are also less helpful for identifying content than request fields, as "Server" could easily be identified using HTTP Request Header identification, and "Via" is rarely relevant. HTTP Response censorship mechanisms normally let the first n packets through while the mirrored traffic is being processed; this may allow some content through and the user may be able to detect that the censor is actively interfering with undesirable content.

Empirical Examples: In 2009, Jong Park et al. at the University of New Mexico demonstrated that the Great Firewall of China (GFW) has used this technique [[Crandall-2010](#)]. However, Jong Park et al. found that the GFW discontinued this practice during the course of the study. Due to the overlap in HTTP response filtering and keyword filtering (see [Section 4.2.4](#)), it is likely that most censors rely on keyword filtering over TCP streams instead of HTTP response filtering.

4.2.3. Transport Layer Security (TLS)

Similar to HTTP, censors have deployed a variety of techniques towards censoring Transport Layer Security (TLS) (and by extension HTTPS). Most of these techniques relate to the Server Name Indication (SNI) field, including censoring SNI, Encrypted SNI, or omitted SNI. Censors can also censor HTTPS content via server certificates. Note that TLS 1.3 acts as a security component of QUIC.

4.2.3.1. Server Name Indication (SNI)

In encrypted connections using TLS, there may be servers that host multiple "virtual servers" at a given network address, and the client will need to specify in the Client Hello message which domain name it seeks to connect to (so that the server can respond with the appropriate TLS certificate) using the Server Name Indication (SNI) TLS extension [[RFC6066](#)]. The Client Hello message is unencrypted for TCP-based TLS. When using QUIC, the Client Hello message is encrypted but its confidentiality is not effectively protected because the initial encryption keys are derived using a value that

is visible on the wire. Since SNI is often sent in the clear (as are the cert fields sent in response), censors and filtering software can use it (and response cert fields) as a basis for blocking, filtering, or impairment by dropping connections to domains that match prohibited content (e.g., bad.foo.example may be censored while good.foo.example is not) [[Shbair-2015](#)]. There are ongoing standardization efforts in the TLS Working Group to encrypt SNI [[I-D.ietf-tls-sni-encryption](#)] [[I-D.ietf-tls-esni](#)] and recent research shows promising results in the use of encrypted SNI in the face of SNI-based filtering [[Chai-2019](#)] in some countries.

Domain fronting has been one popular way to avoid identification by censors [[Fifield-2015](#)]. To avoid identification by censors, applications using domain fronting put a different domain name in the SNI extension than in the Host: header, which is protected by HTTPS. The visible SNI would indicate an unblocked domain, while the blocked domain remains hidden in the encrypted application header. Some encrypted messaging services relied on domain fronting to enable their provision in countries employing SNI-based filtering. These services used the cover provided by domains for which blocking at the domain level would be undesirable to hide their true domain names. However, the companies holding the most popular domains have since reconfigured their software to prevent this practice. It may be possible to achieve similar results using potential future options to encrypt SNI.

Tradeoffs: Some clients do not send the SNI extension (e.g., clients that only support versions of SSL and not TLS), rendering this method ineffective (see [Section 4.2.3.3](#)). In addition, this technique requires deep packet inspection techniques that can be computationally and infrastructurally expensive, especially when applied to QUIC where deep packet inspection requires key extraction and decryption of the Client Hello in order to read the SNI. Improper configuration of an SNI-based block can result in significant overblocking, e.g., when a second-level domain like populardomain.example is inadvertently blocked. In the case of encrypted SNI, pressure to censor may transfer to other points of intervention, such as content and application providers.

Empirical Examples: There are many examples of security firms that offer SNI-based filtering products [[Trustwave-2015](#)] [[Sophos-2015](#)] [[Shbair-2015](#)], and the governments of China, Egypt, Iran, Qatar, South Korea, Turkey, Turkmenistan, and the UAE all do widespread SNI filtering or blocking [[OONI-2018](#)] [[OONI-2019](#)] [[NA-SK-2019](#)] [[CitizenLab-2018](#)] [[Gatlan-2019](#)] [[Chai-2019](#)] [[Grover-2019](#)] [[Singh-2019](#)]. SNI blocking against QUIC traffic has been first observed in Russia in March 2022 [[Elmenhorst-2022](#)].

4.2.3.2. Encrypted SNI (ESNI)

With the data leakage present with the SNI field, a natural response is to encrypt it, which is forthcoming in TLS 1.3 with Encrypted Client Hello (ECH). Prior to ECH, the Encrypted SNI (ESNI) extension is available to prevent the data leakage caused by SNI, which encrypts only the SNI field. Unfortunately, censors can target connections that use the ESNI extension specifically for censorship. This guarantees overblocking for the censor, but can be worth the cost if ESNI is not yet widely deployed within the country. Encrypted Client Hello (ECH) is the emerging standard for protecting the entire TLS Client Hello, but it is not yet widely deployed.

Tradeoffs: The cost to censoring Encrypted SNI (ESNI) is significantly higher than SNI to a censor, as the censor can no longer target censorship to specific domains and guarantees over-blocking. In these cases, the censor uses the over-blocking to discourage the use of ESNI entirely.

Empirical Examples: In 2020, China began censoring all uses of Encrypted ESNI (ESNI) [[Bock-2020b](#)], even for innocuous connections. The censorship mechanism for China's ESNI censorship differs from how China censors SNI-based connections, suggesting that new middleboxes were deployed specifically to target ESNI connections.

4.2.3.3. Omitted-SNI

Researchers have observed that some clients omit the SNI extension entirely. This omitted-SNI approach limits the information available to a censor. Like with ESNI, censors can choose to block connections that omit the SNI, though this too risks over-blocking.

Tradeoffs: The approach of censoring all connections that omit the SNI field is guaranteed to over-block, though connections that omit the SNI field should be relatively rare in the wild.

Empirical Examples: In the past, researchers have observed censors in Russia blocking connections that omit the SNI field [[Bock-2020b](#)].

4.2.3.4. Server Response Certificate

During the TLS handshake after the TLS Client Hello, the server will respond with the TLS certificate. This certificate also contains the domain the client is trying to access, creating another avenue that censors can use to perform censorship. This technique will not work in TLS 1.3, as the certificate will be encrypted.

Tradeoffs: Censoring based on the server certificate requires deep packet inspection techniques that can be more computationally expensive compared to other methods. Additionally, the certificate

is sent later in the TLS Handshake compared to the SNI field, forcing the censor to track the connection for longer.

Empirical Examples: Researchers have observed the Reliance Jio ISP in India using certificate response fields to censor connections [[Satiya-2021](#)].

4.2.4. Instrumenting Content Distributors

Many governments pressure content providers to censor themselves, or provide the legal framework within which content distributors are incentivized to follow the content restriction preferences of agents external to the content distributor [[Boyle-1997](#)]. Due to the extensive reach of such censorship, we define content distributor as any service that provides utility to users, including everything from web sites to locally installed programs. A commonly used method of instrumenting content distributors consists of keyword identification to detect restricted terms on their platform. Governments may provide the terms on such keyword lists. Alternatively, the content provider may be expected to come up with their own list. A different method of instrumenting content distributors consists of requiring a distributor to disassociate with some categories of users. See also [Section 6.4](#).

Tradeoffs: By instrumenting content distributors to identify restricted content or content providers, the censor can gain new information at the cost of political capital with the companies it forces or encourages to participate in censorship. For example, the censor can gain insight about the content of encrypted traffic by coercing web sites to identify restricted content. Coercing content distributors to regulate users, categories of users, content and content providers may encourage users and content providers to exhibit self-censorship, an additional advantage for censors (see [Section 6.2](#)). The tradeoffs for instrumenting content distributors are highly dependent on the content provider and the requested assistance. A typical concern is that the targeted keywords or categories of users are too broad, risk being too broadly applied, or are not subjected to a sufficiently robust legal process prior to their mandatory application (see p. 8 of [[EC-2012](#)]).

Empirical Examples: Researchers discovered keyword identification by content providers on platforms ranging from instant messaging applications [[Senft-2013](#)] to search engines [[Rushe-2015](#)] [[Cheng-2010](#)] [[Whittaker-2013](#)] [[BBC-2013](#)] [[Condliffe-2013](#)]. To demonstrate the prevalence of this type of keyword identification, we look to search engine censorship.

Search engine censorship demonstrates keyword identification by content providers and can be regional or worldwide. Implementation

is occasionally voluntary, but normally it is based on laws and regulations of the country a search engine is operating in. The keyword blocklists are most likely maintained by the search engine provider. China is known to require search engine providers to "voluntarily" maintain search term blocklists to acquire and keep an Internet content provider (ICP) license [Cheng-2010]. It is clear these blocklists are maintained by each search engine provider based on the slight variations in the intercepted searches [Zhu-2011] [Whittaker-2013]. The United Kingdom has been pushing search engines to self-censor with the threat of litigation if they do not do it themselves: Google and Microsoft have agreed to block more than 100,000 queries in U.K. to help combat abuse [BBC-2013] [Condcliffe-2013]. European Union law, as well as US law, requires modification of search engine results in response to either copyright, trademark, data protection or defamation concerns [EC-2012].

Depending on the output, search engine keyword identification may be difficult or easy to detect. In some cases specialized or blank results provide a trivial enumeration mechanism, but more subtle censorship can be difficult to detect. In February 2015, Microsoft's search engine, Bing, was accused of censoring Chinese content outside of China [Rushe-2015] because Bing returned different results for censored terms in Chinese and English. However, it is possible that censorship of the largest base of Chinese search users, China, biased Bing's results so that the more popular results in China (the uncensored results) were also more popular for Chinese speakers outside of China.

Disassociation by content distributors from certain categories of users has happened for instance in Spain, as a result of the conflict between the Catalunyan independence movement and the Spanish legal presumption of a unitary state [Lomas-2019]. E-sport event organizers have also disassociated themselves from top players who expressed political opinions in relation to the 2019 Hong Kong protests [Victor-2019]. See also [Section 5.3.1](#).

4.2.5. Deep Packet Inspection (DPI) Identification

DPI (deep packet inspection) technically is any kind of packet analysis beyond IP address and port number and has become computationally feasible as a component of censorship mechanisms in recent years [Wagner-2009]. Unlike other techniques, DPI reassembles network flows to examine the application "data" section, as opposed to only headers, and is therefore often used for keyword identification. DPI also differs from other identification technologies because it can leverage additional packet and flow characteristics, e.g., packet sizes and timings, when identifying content. To prevent substantial quality of service (QoS) impacts,

DPI normally analyzes a copy of data while the original packets continue to be routed. Typically, the traffic is split using either a mirror switch or fiber splitter, and analyzed on a cluster of machines running Intrusion Detection Systems (IDS) configured for censorship.

Tradeoffs: DPI is one of the most expensive identification mechanisms and can have a large QoS impact [[Porter-2010](#)]. When used as a keyword filter for TCP flows, DPI systems can cause also major overblocking problems. Like other techniques, DPI is less useful against encrypted data, though DPI can leverage unencrypted elements of an encrypted data flow, e.g., the Server Name Indication (SNI) sent in the clear for TLS, or metadata about an encrypted flow, e.g., packet sizes, which differ across video and textual flows, to identify traffic. See [Section 4.2.3.1](#) for more information about SNI-based filtration mechanisms.

Other kinds of information can be inferred by comparing certain unencrypted elements exchanged during TLS handshakes to similar data points from known sources. This practice, called TLS fingerprinting, allows a probabilistic identification of a party's operating system, browser, or application based on a comparison of the specific combinations of TLS version, ciphersuites, compression options, etc. sent in the ClientHello message to similar signatures found in unencrypted traffic [[Husak-2016](#)].

Despite these problems, DPI is the most powerful identification method and is widely used in practice. The Great Firewall of China (GFW), the largest censorship system in the world, uses DPI to identify restricted content over HTTP and DNS and inject TCP RSTs and bad DNS responses, respectively, into connections [[Crandall-2010](#)] [[Clayton-2006](#)] [[Anonymous-2014](#)].

Empirical Examples: Several studies have found evidence of censors using DPI for censoring content and tools. Clayton et al., Crandall et al., Anonymous, and Khattak et al., all explored the GFW [[Crandall-2010](#)] [[Clayton-2006](#)] [[Anonymous-2014](#)]. Khattak et al. even probed the firewall to discover implementation details like how much state it stores [[Khattak-2013](#)]. The Tor project claims that China, Iran, Ethiopia, and others must have used DPI to block the obfs2 protocol [[Wilde-2012](#)]. Malaysia has been accused of using targeted DPI, paired with DDoS, to identify and subsequently attack pro-opposition material [[Wagstaff-2013](#)]. It also seems likely that organizations not so worried about blocking content in real-time could use DPI to sort and categorically search gathered traffic using technologies such as NarusInsight [[Hepting-2011](#)].

4.3. Transport Layer

4.3.1. Shallow Packet Inspection and Transport Header Identification

Of the various shallow packet inspection methods, Transport Header Identification is the most pervasive, reliable, and predictable type of identification. Transport headers contain a few invaluable pieces of information that must be transparent for traffic to be successfully routed: destination and source IP address and port. Destination and Source IP are doubly useful, as not only does it allow a censor to block undesirable content via IP blocklisting, but also allows a censor to identify the IP of the user making the request and the IP address of the destination being visited, which in most cases can be used to infer the domain being visited [[Patil-2019](#)]. Port is useful for allowlisting certain applications.

Combining IP address, port and protocol information found in the transport header, shallow packet inspection can be used by a censor to identify specific TCP or UDP endpoints. UDP endpoint blocking has been observed in the context of QUIC blocking [[Elmenhorst-2021](#)].

Trade-offs: header identification is popular due to its simplicity, availability, and robustness.

Header identification is trivial to implement, but is difficult to implement in backbone or ISP routers at scale, and is therefore typically implemented with DPI. Blocklisting an IP is equivalent to installing a specific route on a router (such as a /32 route for IPv4 addresses and a /128 route for IPv6 addresses). However, due to limited flow table space, this cannot scale beyond a few thousand IPs at most. IP blocking is also relatively crude. It often leads to overblocking and cannot deal with some services like Content Distribution Networks (CDN) that host content at hundreds or thousands of IP addresses. Despite these limitations, IP blocking is extremely effective because the user needs to proxy their traffic through another destination to circumvent this type of identification. In addition, IP blocking is effective against all protocols above IP, e.g. TCP and QUIC.

Port-blocking is generally not useful because many types of content share the same port and it is possible for censored applications to change their port. For example, most HTTP traffic goes over port 80, so the censor cannot differentiate between restricted and allowed web content solely on the basis of port. HTTPS goes over port 443, with similar consequences for the censor except only partial metadata may now be available to the censor. Port allowlisting is occasionally used, where a censor limits communication to approved ports, such as 80 for HTTP traffic and is most effective when used in conjunction with other identification mechanisms. For example, a

censor could block the default HTTPS port, port 443, thereby forcing most users to fall back to HTTP. A counter-example is that port 25 (SMTP) has long been blocked on residential ISPs' networks to reduce the risk for email spam, but in doing so also prohibits residential ISP customers from running their own email servers.

4.3.2. Protocol Identification

Censors sometimes identify entire protocols to be blocked using a variety of traffic characteristics. For example, Iran impairs the performance of HTTPS traffic, a protocol that prevents further analysis, to encourage users to switch to HTTP, a protocol that they can analyze [[Aryan-2012](#)]. A simple protocol identification would be to recognize all TCP traffic over port 443 as HTTPS, but more sophisticated analysis of the statistical properties of payload data and flow behavior, would be more effective, even when port 443 is not used [[Hjelmvik-2010](#)] [[Sandvine-2014](#)].

If censors can detect circumvention tools, they can block them, so censors like China are extremely interested in identifying the protocols for censorship circumvention tools. In recent years, this has devolved into an arms race between censors and circumvention tool developers. As part of this arms race, China developed an extremely effective protocol identification technique that researchers call active probing or active scanning.

In active probing, the censor determines whether hosts are running a circumvention protocol by trying to initiate communication using the circumvention protocol. If the host and the censor successfully negotiate a connection, then the censor conclusively knows that host is running a circumvention tool. China has used active scanning to great effect to block Tor [[Winter-2012](#)].

Trade-offs: Protocol identification necessarily only provides insight into the way information is traveling, and not the information itself.

Protocol identification is useful for detecting and blocking circumvention tools, like Tor, or traffic that is difficult to analyze, like VoIP or SSL, because the censor can assume that this traffic should be blocked. However, this can lead to over-blocking problems when used with popular protocols. These methods are expensive, both computationally and financially, due to the use of statistical analysis, and can be ineffective due to their imprecise nature.

Censors have also used protocol identification in the past in an 'allowlist' filtering capacity, such as by only allowing specific, pre-vetted protocols to be used and blocking any unrecognized

protocols [[Bock-2020](#)]. These protocol filtering approaches can also lead to over-blocking if the allowed lists of protocols is too small or incomplete, but can be cheap to implement, as many standard 'allowed' protocols are simple to identify (such as HTTP).

Empirical Examples: Protocol identification can be easy to detect if it is conducted in real time and only a particular protocol is blocked, but some types of protocol identification, like active scanning, are much more difficult to detect. Protocol identification has been used by Iran to identify and throttle SSH traffic to make it unusable [[Anonymous-2007](#)] and by China to identify and block Tor relays [[Winter-2012](#)]. Protocol identification has also been used for traffic management, such as the 2007 case where Comcast in the United States used RST injection to interrupt BitTorrent Traffic [[Winter-2012](#)]. In 2020, Iran deployed an allowlist protocol filter, which only allowed three protocols to be used (DNS, TLS, and HTTP) on specific ports and censored any connection it could not identify [[Bock-2020](#)]. In 2022, Russia seemed to have used protocol identification to block most HTTP/3 connections [[Elmenhorst-2022](#)].

4.4. Residual Censorship

Another feature of some modern censorship systems is residual censorship, a punitive form of censorship whereby after a censor disrupts a forbidden connection, the censor continues to target subsequent connections, even if they are innocuous [[Bock-2021](#)]. Residual censorship can take many forms and often relies on the methods of technical interference described in the next section.

An important facet of residual censorship is precisely what the censor continues to block after censorship is initially triggered. There are three common options available to an adversary: 2-tuple (client IP, server IP), 3-tuple (client IP, server IP+port), or 4-tuple (client IP+port, server IP+port). Future connections that match the tuple of information the censor records will be disrupted [[Bock-2021](#)].

Residual censorship can sometimes be difficult to identify and can often complicate censorship measurement.

Trade-offs: The impact of residual censorship is to provide users with further discouragement from trying to access forbidden content, though it is not clear how successful it is at accomplishing this.

Empirical Examples: China has used 3-tuple residual censorship in conjunction with their HTTP censorship for years and researchers have reported seeing similar residual censorship for HTTPS. China seems to use a mix of 3-tuple and 4-tuple residual censorship for their censorship of HTTPS with ESNI. Some censors that perform

ensorship via packet dropping often accidentally implement 4-tuple residual censorship, including Iran and Kazakhstan [[Bock-2021](#)].

5. Technical Interference

5.1. Application Layer

5.1.1. DNS Interference

There are a variety of mechanisms that censors can use to block or filter access to content by altering responses from the DNS [[AFNIC-2013](#)] [[ICANN-SSAC-2012](#)], including blocking the response, replying with an error message, or responding with an incorrect address. Note that there are now encrypted transports for DNS queries in DNS-over-HTTPS [[RFC8484](#)] and DNS-over-TLS [[RFC7858](#)] that can mitigate interference with DNS queries between the stub and the resolver.

Responding to a DNS query with an incorrect address can be achieved with on-path interception, off-path cache poisoning, and lying by the nameserver.

"DNS mangling" is a network-level technique of on-path interception where an incorrect IP address is returned in response to a DNS query to a censored destination. An example of this is what some Chinese networks do (we are not aware of any other wide-scale uses of mangling). On those Chinese networks, every DNS request in transit is examined (presumably by network inspection technologies such as DPI) and, if it matches a censored domain, a false response is injected. End users can see this technique in action by simply sending DNS requests to any unused IP address in China (see example below). If it is not a censored name, there will be no response. If it is censored, a forged response will be returned. For example, using the command-line dig utility to query an unused IP address in China of 192.0.2.2 for the name "www.uncensored.example" compared with "www.censored.example" (censored at the time of writing), we get a forged IP address "198.51.100.0" as a response:

```
% dig +short +nodnssec @192.0.2.2 A www.uncensored.example
;; connection timed out; no servers could be reached
```

```
% dig +short +nodnssec @192.0.2.2 A www.censored.example
198.51.100.0
```

DNS cache poisoning happens off-path and refers to a mechanism where a censor interferes with the response sent by an authoritative DNS name server to a recursive resolver by responding more quickly than the authoritative name server can respond with an alternative IP address [[Halley-2008](#)]. Cache poisoning occurs after the requested site's name servers resolve the request and attempt to forward the

true IP back to the requesting device; on the return route the resolved IP is recursively cached by each DNS server that initially forwarded the request. During this caching process if an undesirable keyword is recognized, the resolved IP is "poisoned" and an alternative IP (or NXDOMAIN error) is returned more quickly than the upstream resolver can respond, causing a forged IP address to be cached (and potentially recursively so). The alternative IPs usually direct to a nonsense domain or a warning page. Alternatively, Iranian censorship appears to prevent the communication en-route, preventing a response from ever being sent [[Aryan-2012](#)].

There are also cases of what is colloquially called "DNS lying", where a censor mandates that the DNS responses provided - by an operator of a recursive resolver such as an Internet access provider - be different than what authoritative name server would provide [[Bortzmeyer-2015](#)].

Trade-offs: These forms of DNS interference require the censor to force a user to traverse a controlled DNS hierarchy (or intervening network on which the censor serves as a Active Pervasive Attacker [[RFC7624](#)] to rewrite DNS responses) for the mechanism to be effective. It can be circumvented by using alternative DNS resolvers (such as any of the public DNS resolvers) that may fall outside of the jurisdictional control of the censor, or Virtual Private Network (VPN) technology. DNS mangling and cache poisoning also imply returning an incorrect IP to those attempting to resolve a domain name, but in some cases the destination may be technically accessible; over HTTP, for example, the user may have another method of obtaining the IP address of the desired site and may be able to access it if the site is configured to be the default server listening at this IP address. Target blocking has also been a problem, as occasionally users outside of the censors region will be directed through DNS servers or DNS-rewriting network equipment controlled by a censor, causing the request to fail. The ease of circumvention paired with the large risk of content blocking and target blocking make DNS interference a partial, difficult, and less than ideal censorship mechanism.

Additionally, the above mechanisms rely on DNSSEC not being deployed or DNSSEC validation not being active on the client or recursive resolver (neither of which are hard to imagine given limited deployment of DNSSEC and limited client support for DNSSEC validation). Note that an adversary seeking to merely block resolution can serve a DNSSEC record that doesn't validate correctly, assuming of course that the client/recursive resolver validates.

Previously, techniques were used for e.g. censorship that relied on DNS requests being passed in cleartext over port 53 [[SSAC-109-2020](#)].

With the deployment of encrypted DNS (e.g., DNS-over-HTTPS [[RFC8484](#)]) these requests are now increasingly passed on port 443 with other HTTPS traffic, or in the case of DNS-over-TLS [[RFC7858](#)] no longer passed in the clear (see also [Section 4.3.1](#)).

Empirical Examples: DNS interference, when properly implemented, is easy to identify based on the shortcomings identified above. Turkey relied on DNS interference for its country-wide block of websites such as Twitter and YouTube for almost a week in March of 2014 but the ease of circumvention resulted in an increase in the popularity of Twitter until Turkish ISPs implemented an IP blocklist to achieve the governmental mandate [[Zmijewski-2014](#)]. Ultimately, Turkish ISPs started hijacking all requests to Google and Level 3's international DNS resolvers [[Zmijewski-2014](#)]. DNS interference, when incorrectly implemented, has resulted in some of the largest "censorship disasters". In January 2014, China started directing all requests passing through the Great Fire Wall to a single domain, [dongtaiwang.com](#), due to an improperly configured DNS poisoning attempt; this incident is thought to be the largest Internet-service outage in history [[AFP-2014](#)] [[Anon-SIGCOMM12](#)]. Countries such as China, Iran, Turkey, and the United States have discussed blocking entire TLDs as well, but only Iran has acted by blocking all Israeli (.il) domains [[Albert-2011](#)]. DNS-blocking is commonly deployed in European countries to deal with undesirable content, such as child abuse content (Norway, United Kingdom, Belgium, Denmark, Finland, France, Germany, Ireland, Italy, Malta, the Netherlands, Poland, Spain and Sweden [[Wright-2013](#)] [[Eneman-2010](#)]), online gambling (Belgium, Bulgaria, Czech Republic, Cyprus, Denmark, Estonia, France, Greece, Hungary, Italy, Latvia, Lithuania, Poland, Portugal, Romania, Slovakia, Slovenia, Spain (see Section 6.3.2 of: [[EC-gambling-2012](#)], [[EC-gambling-2019](#)])), copyright infringement (all European Economic Area countries), hate-speech and extremism (France [[Hertel-2015](#)]) and terrorism content (France [[Hertel-2015](#)]).

5.2. Transport Layer

5.2.1. Performance Degradation

While other interference techniques outlined in this section mostly focus on blocking or preventing access to content, it can be an effective censorship strategy in some cases to not entirely block access to a given destination, or service but instead degrade the performance of the relevant network connection. The resulting user experience for a site or service under performance degradation can be so bad that users opt to use a different site, service, or method of communication, or may not engage in communication at all if there are no alternatives. Traffic shaping techniques that rate-limit the bandwidth available to certain types of traffic is one example of a performance degradation.

Trade offs: While implementing a performance degradation will not always eliminate the ability of people to access a desired resource, it may force them to use other means of communication where censorship (or surveillance) is more easily accomplished.

Empirical Examples: Iran has been known to shape the bandwidth available to HTTPS traffic to encourage unencrypted HTTP traffic [[Aryan-2012](#)].

5.2.2. Packet Dropping

Packet dropping is a simple mechanism to prevent undesirable traffic. The censor identifies undesirable traffic and chooses to not properly forward any packets it sees associated with the traversing undesirable traffic instead of following a normal routing protocol. This can be paired with any of the previously described mechanisms so long as the censor knows the user must route traffic through a controlled router.

Trade offs: Packet Dropping is most successful when every traversing packet has transparent information linked to undesirable content, such as a Destination IP. One downside Packet Dropping suffers from is the necessity of blocking all content from otherwise allowable IPs based on a single subversive sub-domain; blogging services and github repositories are good examples. China famously dropped all github packets for three days based on a single repository hosting undesirable content [[Anonymous-2013](#)]. The need to inspect every traversing packet in close to real time also makes Packet Dropping somewhat challenging from a QoS perspective.

Empirical Examples: Packet Dropping is a very common form of technical interference and lends itself to accurate detection given the unique nature of the time-out requests it leaves in its wake. The Great Firewall of China has been observed using packet dropping as one of its primary mechanisms of technical censorship [[Ensafi-2013](#)]. Iran has also used Packet Dropping as the mechanisms for throttling SSH [[Aryan-2012](#)]. These are but two examples of a ubiquitous censorship practice. Notably, packet dropping during the handshake or working connection is the only interference technique observed for QUIC traffic so far, e.g. in India, Iran, Russia and Uganda [[Elmenhorst-2021](#)][[Elmenhorst-2022](#)].

5.2.3. RST Packet Injection

Packet injection, generally, refers to a man-in-the-middle (MITM) network interference technique that spoofs packets in an established traffic stream. RST packets are normally used to let one side of TCP connection know the other side has stopped sending information, and thus the receiver should close the connection. RST Packet Injection

is a specific type of packet injection attack that is used to interrupt an established stream by sending RST packets to both sides of a TCP connection; as each receiver thinks the other has dropped the connection, the session is terminated.

QUIC is not vulnerable to these types of injection attacks once the connection has been setup. While QUIC implements a stateless reset mechanism, such a reset is only accepted by a peer if the packet ends in a previously issued stateless reset token which is hard to guess. During the handshake, QUIC only provides effective protection against off-path attackers but is vulnerable to injection attacks by attackers that have parsed prior packets. (See [[I-D.ietf-quic-transport](#)] for more details.)

Trade-offs: Although ineffective against non-TCP protocols (QUIC, IPSec), RST Packet Injection has a few advantages that make it extremely popular as a technique employed for censorship. RST Packet Injection is an out-of-band interference mechanism, allowing the avoidance of the the QoS bottleneck one can encounter with inline techniques such as Packet Dropping. This out-of-band property allows a censor to inspect a copy of the information, usually mirrored by an optical splitter, making it an ideal pairing for DPI and protocol identification [[Weaver-2009](#)] (this asynchronous version of a MITM is often called a Man-on-the-Side (MOTS)). RST Packet Injection also has the advantage of only requiring one of the two endpoints to accept the spoofed packet for the connection to be interrupted.

The difficult part of RST Packet Injection is spoofing "enough" correct information to ensure one end-point accepts a RST packet as legitimate; this generally implies a correct IP, port, and TCP sequence number. Sequence number is the hardest to get correct, as [[RFC0793](#)] specifies an RST Packet should be in-sequence to be accepted, although the RFC also recommends allowing in-window packets as "good enough". This in-window recommendation is important, as if it is implemented it allows for successful Blind RST Injection attacks [[Netsec-2011](#)]. When in-window sequencing is allowed, it is trivial to conduct a Blind RST Injection: while the term "blind" injection implies the censor doesn't know any sensitive sequencing information about the TCP stream they are injecting into, they can simply enumerate all ~70000 possible windows; this is particularly useful for interrupting encrypted/obfuscated protocols such as SSH or Tor [[Gilad](#)]. Some censorship evasion systems work by trying to confuse the censor into tracking incorrect information, rendering their RST Packet Injection useless [[Khattak-2013](#)], [[Wang-2017](#)], [[Li-2017](#)], [[Bock-2019](#)], [[Wang-2020](#)].

RST Packet Injection relies on a stateful network, making it useless against UDP connections. RST Packet Injection is among the most popular censorship techniques used today given its versatile nature

and effectiveness against all types of TCP traffic. Recent research shows that a TCP RST packet injection attack can even work in the case of an off-path attacker [[Cao-2016](#)].

Empirical Examples: RST Packet Injection, as mentioned above, is most often paired with identification techniques that require splitting, such as DPI or protocol identification. In 2007, Comcast was accused of using RST Packet Injection to interrupt traffic it identified as BitTorrent [[Schoen-2007](#)], this later led to a US Federal Communications Commission ruling against Comcast [[VonLohmann-2008](#)]. China has also been known to use RST Packet Injection for censorship purposes. This interference is especially evident in the interruption of encrypted/obfuscated protocols, such as those used by Tor [[Winter-2012](#)].

5.3. Routing Layer

5.3.1. Network Disconnection

While it is perhaps the crudest of all techniques employed for censorship, there is no more effective way of making sure undesirable information isn't allowed to propagate on the web than by shutting off the network. The network can be logically cut off in a region when a censoring body withdraws all of the Border Gateway Protocol (BGP) prefixes routing through the censor's country.

Trade-offs: The impact to a network disconnection in a region is huge and absolute; the censor pays for absolute control over digital information by losing all the benefits the Internet brings; this rarely a long-term solution for any censor and is normally only used as a last resort in times of substantial unrest.

Empirical Examples: Network Disconnections tend to only happen in times of substantial unrest, largely due to the huge social, political, and economic impact such a move has. One of the first, highly covered occurrences was with the Junta in Myanmar employing Network Disconnection to help Junta forces quash a rebellion in 2007 [[Dobie-2007](#)]. China disconnected the network in the Xinjiang region during unrest in 2009 in an effort to prevent the protests from spreading to other regions [[Heacock-2009](#)]. The Arab Spring saw the the most frequent usage of Network Disconnection, with events in Egypt and Libya in 2011 [[Cowie-2011](#)], and Syria in 2012 [[Thomson-2012](#)]. Russia indicated that it would attempt to disconnect all Russian networks from the global internet in April 2019 as part of a test of the nation's network independence. Reports also indicate that, as part of the test disconnect, Russian telecommunications firms must now route all traffic to state-operated monitoring points [[Cimpanu-2019](#)]. India was the country

that saw the largest number of internet shutdowns per year in 2016 and 2017 [[Dada-2017](#)].

5.3.2. Adversarial Route Announcement

More fine-grained and potentially wide-spread censorship can be achieved with BGP hijacking, which adversarially re-routes BGP IP prefixes incorrectly within a region and beyond. This restricts and effectively censors the correctly known location of information that flows into or out of a jurisdiction and will similarly prevent people from outside your jurisdiction from viewing content generated outside your jurisdiction as the adversarial route announcement propagates. The first can be achieved by an adversarial BGP announcement of incorrect routes that are not intended to leak beyond a jurisdiction, where the latter attacks traffic by deliberately introducing bogus BGP announcements that reach the global internet.

Trade-offs: A global leak of a misrouted website can overwhelm an ISP if the website gets a lot of traffic. It is not a permanent solution because incorrect BGP routes that leak globally can be fixed, though within a jurisdiction only the ISP/IXP is in a position to correct them for local users.

Empirical examples: In 2008 Pakistan Telecom censored Youtube at the request of the Pakistan government by changing its BGP routes for the website. The new routes were announced to the ISP's upstream providers and beyond. The entire Internet began directing Youtube routes to Pakistan Telecom and continued doing so for many hours. In 2018 nearly all Google services and Google cloud customers like Spotify all lost more than one hour of service after it lost control of several million of its IP addresses. Those IP prefixes were being misdirected to China Telecom, a Chinese government-owned ISP [[Google-2018](#)], in a manner similar to the BGP hijacking of US government and military websites by China Telecom in 2010. ISPs in both Russia (2022) and Myanmar (2021) have tried to hijack the same Twitter prefix more than once [[MANRS](#)].

5.4. Multi-layer and Non-layer

5.4.1. Distributed Denial of Service (DDoS)

Distributed Denial of Service attacks are a common attack mechanism used by "hacktivists" and malicious hackers, but censors have used DDoS in the past for a variety of reasons. There is a huge variety of DDoS attacks [[Wikip-DoS](#)], but at a high level two possible impacts tend to occur; a flood attack results in the service being unusable while resources are being spent to flood the service, a

crash attack aims to crash the service so resources can be reallocated elsewhere without "releasing" the service.

Trade-offs: DDoS is an appealing mechanism when a censor would like to prevent all access to undesirable content, instead of only access in their region for a limited period of time, but this is really the only uniquely beneficial feature for DDoS as a technique employed for censorship. The resources required to carry out a successful DDoS against major targets are computationally expensive, usually requiring renting or owning a malicious distributed platform such as a botnet, and imprecise. DDoS is an incredibly crude censorship technique, and appears to largely be used as a timely, easy-to-access mechanism for blocking undesirable content for a limited period of time.

Empirical Examples: In 2012 the U.K.'s GCHQ used DDoS to temporarily shutdown IRC chat rooms frequented by members of Anonymous using the Syn Flood DDoS method; Syn Flood exploits the handshake used by TCP to overload the victim server with so many requests that legitimate traffic becomes slow or impossible [[Schone-2014](#)] [[CERT-2000](#)]. Dissenting opinion websites are frequently victims of DDoS around politically sensitive events in Burma [[Villeneuve-2011](#)]. Controlling parties in Russia [[Kravtsova-2012](#)], Zimbabwe [[Orion-2013](#)], and Malaysia [[Muncaster-2013](#)] have been accused of using DDoS to interrupt opposition support and access during elections. In 2015, China launched a DDoS attack using a true MITM system collocated with the Great Firewall, dubbed "Great Cannon", that was able to inject JavaScript code into web visits to a Chinese search engine that commandeered those user agents to send DDoS traffic to various sites [[Marczak-2015](#)].

5.4.2. Censorship in Depth

Often, censors implement multiple techniques in tandem, creating "censorship in depth". Censorship in depth can take many forms; some censors block the same content through multiple techniques (such as blocking a domain by DNS, IP blocking, and HTTP simultaneously), some deploy parallel systems to improve censorship reliability (such as deploying multiple different censorship systems to block the same domain), and others can use complimentary systems to limit evasion (such as by blocking unwanted protocols entirely, forcing users to use other filtered protocols).

Trade-offs: Censorship in depth can be attractive for censors to deploy, as it offers additional guarantees about censorship: even if someone evades one type of censorship, they may still be blocked by another. The main drawback to this approach is the cost to initial deployment, as it requires the system to deploy multiple censorship systems in tandem.

Empirical Examples: Censorship in depth is present in many large censoring nation states today. Researchers have observed China has deployed significant censorship in depth, often censoring the same resource across multiple protocols [[Chai-2019](#)], [[Bock-2020b](#)] or deploying additional censorship systems to censor the same content and protocol [[Bock-2021b](#)]. Iran also has deployed a complimentary protocol filter to limit which protocols can be used on certain ports, forcing users to rely on protocols their censorship system can filter [[Bock-2020](#)].

6. Non-Technical Interference

6.1. Manual Filtering

As the name implies, sometimes manpower is the easiest way to figure out which content to block. Manual Filtering differs from the common tactic of building up blocklists in that it doesn't necessarily target a specific IP or DNS, but instead removes or flags content. Given the imprecise nature of automatic filtering, manually sorting through content and flagging dissenting websites, blogs, articles and other media for filtration can be an effective technique. This filtration can occur on the Backbone/ISP level - China's army of monitors is a good example [[BBC-2013b](#)] - but more commonly manual filtering occurs on an institutional level. Internet Content Providers such as Google or Weibo, require a business license to operate in China. One of the prerequisites for a business license is an agreement to sign a "voluntary pledge" known as the "Public Pledge on Self-discipline for the Chinese Internet Industry". The failure to "energetically uphold" the pledged values can lead to the ICPs being held liable for the offending content by the Chinese government [[BBC-2013b](#)].

6.2. Self-Censorship

Self-censorship is difficult to document, as it manifests primarily through a lack of undesirable content. Tools which encourage self-censorship are those which may lead a prospective speaker to believe that speaking increases the risk of unfavourable outcomes for the speaker (technical monitoring, identification requirements, etc.). Reporters Without Borders exemplify methods of imposing self-censorship in their annual World Press Freedom Index reports [[RWB2020](#)].

6.3. Server Takedown

As mentioned in passing by [[Murdoch-2011](#)], servers must have a physical location somewhere in the world. If undesirable content is hosted in the censoring country the servers can be physically seized or - in cases where a server is virtualized in a cloud

infrastructure where it may not necessarily have a fixed physical location - the hosting provider can be required to prevent access.

6.4. Notice and Takedown

In many countries, legal mechanisms exist where an individual or other content provider can issue a legal request to a content host that requires the host to take down content. Examples include the systems employed by companies like Google to comply with "Right to be Forgotten" policies in the European Union [[Google-RTBF](#)], intermediary liability rules for electronic platform providers [[EC-2012](#)], or the copyright-oriented notice and takedown regime of the United States Digital Millennium Copyright Act (DMCA) Section 512 [[DMLP-512](#)].

6.5. Domain-Name Seizures

Domain names are catalogued in so-called name-servers operated by legal entities called registries. These registries can be made to cede control over a domain name to someone other than the entity which registered the domain name through a legal procedure grounded in either private contracts or public law. Domain name seizures is increasingly used by both public authorities and private entities to deal with undesired content dissemination [[ICANN2012](#)] [[EFF2017](#)].

7. Future work

In addition to establishing a thorough resource for describing censorship techniques this document implicates critical areas for future work.

Taken as a whole the apparent costs of implementation of censorship techniques indicate a need for better classification of censorship regimes as they evolve and mature and specifying censorship circumvention techniques themselves. Censors maturity refers to the technical maturity required of the censor to perform the specific censorship technique. Future work might classify techniques by essentially how hard a censor must work, including what infrastructure is required, in order to successfully censor content, users or services.

On circumvention, the increase in protocols leveraging encryption is an effective counter-measure against some forms of censorship described in this document, but that thorough research on circumvention and encryption be left for another document. Moreover the censorship circumvention community has developed an area of research on "pluggable transports," which collects, documents and makes agile methods for obfuscating the on-path traffic of censorship circumvention tools such that it appears indistinguishable from other kinds of traffic [[Tor-2020](#)]. Those

methods would benefit from future work in the internet standards community, too.

Lastly the empirical examples demonstrate that censorship techniques can evolve quickly, and experience shows that this document can only be a point-in-time statement. Future work might extend this document with updates and new techniques described using a comparable methodology.

8. Contributors

This document benefited from discussions with and input from David Belson, Stephane Bortzmeyer, Vinicius Fortuna, Gurshabad Grover, Andrew McConachie, Martin Nilsson, Michael Richardson, Patrick Vacek and Chris Wood.

9. Informative References

- [AFNIC-2013] AFNIC, "Report of the AFNIC Scientific Council: Consequences of DNS-based Internet filtering", 2013, <<http://www.afnic.fr/medias/documents/conseilscientifique/SC-consequences-of-DNS-based-Internet-filtering.pdf>>.
- [AFP-2014] AFP, "China Has Massive Internet Breakdown Reportedly Caused By Their Own Censoring Tools", 2014, <<http://www.businessinsider.com/chinas-internet-breakdown-reportedly-caused-by-censoring-tools-2014-1>>.
- [Albert-2011] Albert, K., "DNS Tampering and the new ICANN gTLD Rules", 2011, <<https://opennet.net/blog/2011/06/dns-tampering-and-new-icann-gtld-rules>>.
- [Anon-SIGCOMM12] Anonymous, "The Collateral Damage of Internet Censorship by DNS Injection", 2012, <<http://www.sigcomm.org/sites/default/files/ccr/papers/2012/July/2317307-2317311.pdf>>.
- [Anonymous-2007] Anonymous, "How to Bypass Comcast's Bittorrent Throttling", 2012, <<https://torrentfreak.com/how-to-bypass-comcast-bittorrent-throttling-071021>>.
- [Anonymous-2013] Anonymous, "GitHub blocked in China - how it happened, how to get around it, and where it will take us", 2013, <<https://en.greatfire.org/blog/2013/jan/github-blocked-china-how-it-happened-how-get-around-it-and-where-it-will-take-us>>.
- [Anonymous-2014] Anonymous, "Towards a Comprehensive Picture of the Great Firewall's DNS Censorship", 2014, <<https://>

www.usenix.org/system/files/conference/foci14/foci14-anonymous.pdf>.

[AP-2012] Associated Press, "Sattar Beheshit, Iranian Blogger, Was Beaten In Prison According To Prosecutor", 2012, <http://www.huffingtonpost.com/2012/12/03/sattar-beheshit-iran_n_2233125.html>.

[Aryan-2012] Aryan, S., Aryan, H., and J.A. Halderman, "Internet Censorship in Iran: A First Look", 2012, <<https://jhalderm.com/pub/papers/iran-foci13.pdf>>.

[BBC-2013] BBC News, "Google and Microsoft agree steps to block abuse images", 2013, <<http://www.bbc.com/news/uk-24980765>>.

[BBC-2013b] BBC, "China employs two million microblog monitors state media say", 2013, <<http://www.bbc.com/news/world-asia-china-2439695>>.

[Bentham-1791] Bentham, J., "Panopticon Or the Inspection House", 1791, <https://www.google.com/books/edition/_/Ec4TAAAAQAAJ?hl=en>.

[Bock-2019] Bock, K., Hughey, G., Qiang, X., and D. Levin, "Geneva: Evolving Censorship Evasion Strategies", 2019, <https://geneva.cs.umd.edu/papers/geneva_ccs19.pdf>.

[Bock-2020] Bock, K., Fax, Y., Reese, K., Singh, J., and D. Levin, "Detecting and Evading Censorship-in-Depth: A Case Study of Iran's Protocol Filter", 2020, <<https://geneva.cs.umd.edu/papers/evading-censorship-in-depth.pdf>>.

[Bock-2020b]

Bock, K., iyouport, ., Anonymous, ., Merino, L., Fifield, D., Houmansadr, A., and D. Levin, "Exposing and Circumventing China's Censorship of ESNI", 2020, <<https://geneva.cs.umd.edu/posts/china-censors-esni/esni/>>.

[Bock-2021] Bock, K., Bharadwaj, P., Singh, J., and D. Levin, "Your Censor is My Censor: Weaponizing Censorship Infrastructure for Availability Attacks", 2021, <<https://geneva.cs.umd.edu/papers/woot21-weaponizing-availability.pdf>>.

[Bock-2021b] Bock, K., Naval, G., Reese, K., and D. Levin, "Even Censors Have a Backup: Examining China's Double HTTPS

Censorship Middleboxes", 2021, <<https://geneva.cs.umd.edu/papers/foci21.pdf>>.

[Bortzmeyer-2015] Bortzmeyer, S., "DNS Censorship (DNS Lies) As Seen By RIPE Atlas", 2015, <https://labs.ripe.net/Members/stephane_bortzmeyer/dns-censorship-dns-lies-seen-by-atlas-probes>.

[Boyle-1997] Boyle, J., "Foucault in Cyberspace: Surveillance, Sovereignty, and Hardwired Censors", 1997, <https://scholarship.law.duke.edu/faculty_scholarship/619/>.

[Bristow-2013] Bristow, M., "China's internet 'spin doctors'", 2013, <<http://news.bbc.co.uk/2/hi/asia-pacific/7783640.stm>>.

[Calamur-2013] Calamur, K., "Prominent Egyptian Blogger Arrested", 2013, <<http://www.npr.org/blogs/thetwo-way/2013/11/29/247820503/prominent-egyptian-blogger-arrested>>.

[Cao-2016] Cao, Y., Qian, Z., Wang, Z., Dao, T., Krishnamurthy, S., and L. Marvel, "Off-Path TCP Exploits: Global Rate Limit Considered Dangerous", 2016, <https://www.usenix.org/system/files/conference/usenixsecurity16/sec16_paper_cao.pdf>.

[CERT-2000] CERT, "TCP SYN Flooding and IP Spoofing Attacks", 2000, <<http://www.cert.org/historical/advisories/CA-1996-21.cfm>>.

[Chai-2019] Chai, Z., Ghafari, A., and A. Houmansadr, "On the Importance of Encrypted-SNI (ESNI) to Censorship Circumvention", 2019, <https://www.usenix.org/system/files/foci19-paper_chai_update.pdf>.

[Cheng-2010] Cheng, J., "Google stops Hong Kong auto-redirect as China plays hardball", 2010, <<http://arstechnica.com/tech-policy/2010/06/google-tweaks-china-to-hong-kong-redirect-same-results/>>.

[Cimpanu-2019] Cimpanu, C., "Russia to disconnect from the internet as part of a planned test", 2019, <<https://www.zdnet.com/article/russia-to-disconnect-from-the-internet-as-part-of-a-planned-test/>>.

[CitizenLab-2018] Marczak, B., Dalek, J., McKune, S., Senft, A., Scott-Railton, J., and R. Deibert, "Bad Traffic: Sandvine's PacketLogic Devices Used to Deploy Government Spyware in Turkey and Redirect Egyptian Users to Affiliate Ads?", 2018, <<https://citizenlab.ca/2018/03/>>

[bad-traffic-sandvines-packetlogic-devices-deploy-government-spyware-turkey-syria/](#)>.

[Clayton-2006] Clayton, R., "Ignoring the Great Firewall of China", 2006, <http://link.springer.com/chapter/10.1007/11957454_2>.

[Condliffe-2013] Condliffe, J., "Google Announces Massive New Restrictions on Child Abuse Search Terms", 2013, <<http://gizmodo.com/google-announces-massive-new-restrictions-on-child-abus-1466539163>>.

[Cowie-2011] Cowie, J., "Egypt Leaves the Internet", 2011, <<https://archive.nanog.org/meetings/nanog51/presentations/Tuesday/LT-Cowie-Egypt%20Leaves%20The%20Internet.pdf>>.

[Crandall-2010] Crandall, J., "Empirical Study of a National-Scale Distributed Intrusion Detection System: Backbone-Level Filtering of HTML Responses in China", 2010, <<http://www.cs.unm.edu/~crandall/icdcs2010.pdf>>.

[Dada-2017] Dada, T. and P. Micek, "Launching STOP: the #KeepItOn internet shutdown tracker", 2017, <<https://www.accessnow.org/keepiton-shutdown-tracker/>>.

[Dalek-2013] Dalek, J., "A Method for Identifying and Confirming the Use of URL Filtering Products for Censorship", 2013, <<http://conferences.sigcomm.org/imc/2013/papers/imc112s-dalekA.pdf>>.

[Ding-1999] Ding, C., Chi, C.H., Deng, J., and C.L. Dong, "Centralized Content-Based Web Filtering and Blocking: How Far Can It Go?", 1999, <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.132.3302&rep=rep1&type=pdf>>.

[DMLP-512] Digital Media Law Project, "Protecting Yourself Against Copyright Claims Based on User Content", 2012, <<http://www.dmlp.org/legal-guide/protecting-yourself-against-copyright-claims-based-user-content>>.

[Dobie-2007] Dobie, M., "Junta tightens media screw", 2007, <<http://news.bbc.co.uk/2/hi/asia-pacific/7016238.stm>>.

[EC-2012] European Commission, "Summary of the results of the Public Consultation on the future of electronic commerce in the Internal Market and the implementation of the Directive on electronic commerce (2000/31/EC)", 2012, <https://ec.europa.eu/information_society/newsroom/image/document/2017-4/consultation_summary_report_en_2010_42070.pdf>.

[EC-gambling-2012]

European Commission, "Online gambling in the Internal Market", 2012, <<https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52012SC0345>>.

[EC-gambling-2019]

European Commission, "Evaluation of regulatory tools for enforcing online gambling rules and channeling demand towards controlled offers", 2019, <https://ec.europa.eu/growth/content/evaluation-regulatory-tools-enforcing-online-gambling-rules-and-channelling-demand-towards-1_en>.

[EFF2017]

Malcom, J., Stoltz, M., Rossi, G., and V. Paxson, "Which Internet registries offer the best protection for domain owners?", 2017, <https://www.eff.org/files/2017/08/02/domain_registry_whitepaper.pdf>.

[Ellul-1973]

Ellul, J., "Propaganda: The Formation of Men's Attitudes", 1973, <<https://www.penguinrandomhouse.com/books/46234/propaganda-by-jacques-ellul/>>.

[Elmenhorst-2021]

Elmenhorst, K., Schuetz, B., Basso, S., and N. Aschenbruck, "Web Censorship Measurements of HTTP/3 over QUIC", 2021, <<https://dl.acm.org/doi/pdf/10.1145/3487552.3487836>>.

[Elmenhorst-2022]

Elmenhorst, K., "A Quick Look at QUIC Censorship", 2022, <<https://www.opentech.fund/news/a-quick-look-at-quic/>>.

[Eneman-2010]

Eneman, M., "ISPs filtering of child abusive material: A critical reflection of its effectiveness", 2010, <<https://www.gu.se/forskning/publikation/?publicationId=96592>>.

[Ensafi-2013]

Ensafi, R., "Detecting Intentional Packet Drops on the Internet via TCP/IP Side Channels", 2013, <<http://arxiv.org/pdf/1312.5739v1.pdf>>.

[Fareed-2008]

Fareed, M., "China joins a turf war", 2008, <<http://www.theguardian.com/media/2008/sep/22/chinathemedia.marketingandpr>>.

[Fifield-2015]

Fifield, D., Lan, C., Hynes, R., Wegmann, P., and V. Paxson, "Blocking-resistant communication through domain

fronting", 2015, <https://petsymposium.org/2015/papers/03_Fifield.pdf>.

[Gao-2014] Gao, H., "Tiananmen, Forgotten", 2014, <<http://www.nytimes.com/2014/06/04/opinion/tiananmen-forgotten.html>>.

[Gatlan-2019] Gatlan, S., "South Korea is Censoring the Internet by Snooping on SNI Traffic", 2019, <<https://www.bleepingcomputer.com/news/security/south-korea-is-censoring-the-internet-by-snooping-on-sni-traffic/>>.

[Gilad] Gilad, Y. and A. Herzberg, "Off-Path TCP Injection Attacks", 2014, <<https://doi.org/10.1145/2597173>>.

[Glanville-2008] Glanville, J., "The Big Business of Net Censorship", 2008, <<http://www.theguardian.com/commentisfree/2008/nov/17/censorship-internet>>.

[Google-2018] "Google Cloud Networking Incident #18018", 2018, <<https://status.cloud.google.com/incident/cloud-networking/18018>>.

[Google-RTBF] Google, Inc., "Search removal request under data protection law in Europe", 2015, <https://support.google.com/legal/contact/lr_eudpa?product=websearch>.

[Grover-2019] Grover, G., Singh, K., and E. Hickok, "Reliance Jio is using SNI inspection to block websites", 2019, <<https://cis-india.org/internet-governance/blog/reliance-jio-is-using-sni-inspection-to-block-websites>>.

[Guardian-2014] The Gaurdian, "Chinese blogger jailed under crackdown on 'internet rumours'", 2014, <<http://www.theguardian.com/world/2014/apr/17/chinese-blogger-jailed-crackdown-internet-rumours-qin-zhihui>>.

[HADOPI-2020] Haute Autorité pour la Diffusion des oeuvres et la Protection des Droits sur Internet, "Présentation", 2020, <<https://www.hadopi.fr/en/node/3668>>.

[Halley-2008] Halley, B., "How DNS cache poisoning works", 2014, <<https://www.networkworld.com/article/2277316/tech-primers/tech-primers-how-dns-cache-poisoning-works.html>>.

[Heacock-2009] Heacock, R., "China Shuts Down Internet in Xinjiang Region After Riots", 2009, <<https://opennet.net/blog/2009/07/china-shuts-down-internet-xinjiang-region-after-riots>>.

[Hepting-2011]

Electronic Frontier Foundation, "Hepting vs. AT&T", 2011, <<https://www.eff.org/cases/hepting>>.

[Hertel-2015] Hertel, O., "Comment les autorités peuvent bloquer un

site Internet", 2015, <https://www.sciencesetavenir.fr/high-tech/comment-les-autorites-peuvent-bloquer-un-site-internet_35828>.

[Hjelmvik-2010] Hjelmvik, E., "Breaking and Improving Protocol

Obfuscation", 2010, <https://www.iis.se/docs/hjelmvik_breaking.pdf>.

[Hopkins-2011] Hopkins, C., "Communications Blocked in Libya, Qatari

Blogger Arrested: This Week in Online Tyranny", 2011, <<http://readwrite.com/2011/03/03/communications-blocked-in-libya-this-week-in-onlin>>.

[Husak-2016] Husak, M., Cermak, M., Jirsik, T., and P. Celeda,

"HTTPS traffic analysis and client identification using passive SSL/TLS fingerprinting", 2016, <<https://link.springer.com/article/10.1186/s13635-016-0030-7>>.

[I-D.ietf-quic-transport] Iyengar, J. and M. Thomson, "QUIC: A UDP-

Based Multiplexed and Secure Transport", Work in Progress, Internet-Draft, draft-ietf-quic-transport-34, 14 January 2021, <<https://www.ietf.org/archive/id/draft-ietf-quic-transport-34.txt>>.

[I-D.ietf-tls-esni] Rescorla, E., Oku, K., Sullivan, N., and C. A.

Wood, "TLS Encrypted Client Hello", Work in Progress, Internet-Draft, draft-ietf-tls-esni-14, 13 February 2022, <<https://www.ietf.org/archive/id/draft-ietf-tls-esni-14.txt>>.

[I-D.ietf-tls-sni-encryption] Huitema, C. and E. Rescorla, "Issues

and Requirements for Server Name Identification (SNI) Encryption in TLS", Work in Progress, Internet-Draft, draft-ietf-tls-sni-encryption-09, 28 October 2019, <<https://www.ietf.org/archive/id/draft-ietf-tls-sni-encryption-09.txt>>.

[ICANN-SSAC-2012] ICANN Security and Stability Advisory Committee

(SSAC), "SAC 056: SSAC Advisory on Impacts of Content Blocking via the Domain Name System", 2012, <<https://www.icann.org/en/system/files/files/sac-056-en.pdf>>.

[ICANN2012] ICANN Security and Stability Advisory Committee,

"Guidance for Preparing Domain Name Orders, Seizures &

Takedowns", 2012, <<https://www.icann.org/en/system/files/files/guidance-domain-seizures-07mar12-en.pdf>>.

[Jones-2014] Jones, B., "Automated Detection and Fingerprinting of Censorship Block Pages", 2014, <<http://conferences2.sigcomm.org/imc/2014/papers/p299.pdf>>.

[Khattak-2013] Khattak, S., "Towards Illuminating a Censorship Monitor's Model to Facilitate Evasion", 2013, <<http://0b4af6cdc2f0c5998459-c0245c5c937c5dedcca3f1764ecc9b2f.r43.cf2.rackcdn.com/12389-foci13-khattak.pdf>>.

[Knight-2005] Knight, W., "Iranian net censorship powered by US technology", 2005, <<https://www.newscientist.com/article/dn7589-iranian-net-censorship-powered-by-us-technology/>>.

[Knockel-2021] Knockel, J. and L. Ruan, "Measuring QQMail's automated email censorship in China", 2021, <<https://dl.acm.org/doi/10.1145/3473604.3474560>>.

[Kopel-2013] Kopel, K., "Operation Seizing Our Sites: How the Federal Government is Taking Domain Names Without Prior Notice", 2013, <<http://dx.doi.org/doi:10.15779/Z384Q3M>>.

[Kravtsova-2012] Kravtsova, Y., "Cyberattacks Disrupt Opposition's Election", 2012, <<http://www.themoscowtimes.com/news/article/cyberattacks-disrupt-oppositions-election/470119.html>>.

[Leyba-2019] Leyba, K., Edwards, B., Freeman, C., Crandall, J., and S. Forrest, "Borders and Gateways: Measuring and Analyzing National AS Chokepoints", 2019, <<https://forrest.biodesign.asu.edu/data/publications/2019-compass-chokepoints.pdf>>.

[Li-2017] Li, F., Razaghpanah, A., Kakhki, A., Niaki, A., Choffnes, D., Gill, P., and A. Mislove, "lib•erate, (n) : A library for exposing (traffic-classification) rules and avoiding them efficiently", 2017, <<https://david.choffnes.com/pubs/liberate-imc17.pdf>>.

[Lomas-2019] Lomas, N., "Github removes Tsunami Democràtic's APK after a takedown order from Spain", 2019, <<https://techcrunch.com/2019/10/30/github-removes-tsunami-democratics-apk-after-a-takedown-order-from-spain/>>.

[MANRS] Siddiqui, A., "Lesson Learned: Twitter Shored Up Its Routing Security", 2022, <<https://www.manrs.org/2022/03/lesson-learned-twitter-shored-up-its-routing-security/>>.

[Marczak-2015]

Marczak, B., Weaver, N., Dalek, J., Ensafi, R., Fifield, D., McKune, S., Rey, A., Scott-Railton, J., Deibert, R., and V. Paxson, "An Analysis of China's "Great Cannon"", 2015, <<https://www.usenix.org/system/files/conference/foci15/foci15-paper-marczak.pdf>>.

[Muncaster-2013] Muncaster, P., "Malaysian election sparks web blocking/DDoS claims", 2013, <http://www.theregister.co.uk/2013/05/09/malaysia_fraud_elections_ddos_web_blocking/>.

[Murdoch-2011] Murdoch, S.J. and R. Anderson, "Access Denied: Tools and Technology of Internet Filtering", 2011, <<http://access.opennet.net/wp-content/uploads/2011/12/accessdenied-chapter-3.pdf>>.

[NA-SK-2019] Morgus, R., Sherman, J., and S. Nam, "Analysis: South Korea's New Tool for Filtering Illegal Internet Content", 2019, <<https://www.newamerica.org/cybersecurity-initiative/c2b/c2b-log/analysis-south-koreas-sni-monitoring/>>.

[Nabi-2013] Nabi, Z., "The Anatomy of Web Censorship in Pakistan", 2013, <<http://0b4af6cdc2f0c5998459-c0245c5c937c5dedcca3f1764ecc9b2f.r43.cf2.rackcdn.com/12387-foci13-nabi.pdf>>.

[Netsec-2011] n3t2.3c, "TCP-RST Injection", 2011, <https://nets.ec/TCP-RST_Injection>.

[OONI-2018] Evdokimov, L., "Iran Protests: DPI blocking of Instagram (Part 2)", 2018, <<https://ooni.org/post/2018-iran-protests-pt2/>>.

[OONI-2019] Singh, S., Filastò, A., and M. Xynou, "China is now blocking all language editions of Wikipedia", 2019, <<https://ooni.org/post/2019-china-wikipedia-blocking/>>.

[Orion-2013] Orion, E., "Zimbabwe election hit by hacking and DDoS attacks", 2013, <<http://www.theinquirer.net/inquirer/>>.

news/2287433/zimbabwe-election-hit-by-hacking-and-ddos-attacks>.

[Patil-2019] Patil, S. and N. Borisov, "What Can You Learn from an IP?", 2019, <<https://irtf.org/anrw/2019/anrw2019-final44-acmpaginated.pdf>>.

[Porter-2010] Porter, T., "The Perils of Deep Packet Inspection", 2010, <<http://www.symantec.com/connect/articles/perils-deep-packet-inspection>>.

[Rambert-2021] Rampert, R., Weinberg, Z., Barradas, D., and N. Christin, "Chinese Wall or Swiss Cheese? Keyword filtering in the Great Firewall of China", 2021, <<https://www.andrew.cmu.edu/user/nicolasc/publications/Rambert-WWW21.pdf>>.

[Reda-2017] Reda, J., "New EU law prescribes website blocking in the name of 'consumer protection'", 2017, <<https://juliareda.eu/2017/11/eu-website-blocking/>>.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <<https://www.rfc-editor.org/info/rfc793>>.

[RFC6066] Eastlake 3rd, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, DOI 10.17487/RFC6066, January 2011, <<https://www.rfc-editor.org/info/rfc6066>>.

[RFC7624] Barnes, R., Schneier, B., Jennings, C., Hardie, T., Trammell, B., Huitema, C., and D. Borkmann, "Confidentiality in the Face of Pervasive Surveillance: A Threat Model and Problem Statement", RFC 7624, DOI 10.17487/RFC7624, August 2015, <<https://www.rfc-editor.org/info/rfc7624>>.

[RFC7754] Barnes, R., Cooper, A., Kolkman, O., Thaler, D., and E. Nordmark, "Technical Considerations for Internet Service Blocking and Filtering", RFC 7754, DOI 10.17487/RFC7754, March 2016, <<https://www.rfc-editor.org/info/rfc7754>>.

[RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <<https://www.rfc-editor.org/info/rfc7858>>.

[RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <<https://www.rfc-editor.org/info/rfc8484>>.

[RSF-2005]

Reporters Sans Frontieres, "Technical ways to get around censorship", 2005, <http://archives.rsf.org/print-blogs.php3?id_article=15013>.

[Rushe-2015] Rushe, D., "Bing censoring Chinese language search results for users in the US", 2013, <<http://www.theguardian.com/technology/2014/feb/11/bing-censors-chinese-language-search-results>>.

[RWB2020] Reporters Without Borders, "2020 World Press Freedom Index: Entering a decisive decade for journalism, exacerbated by coronavirus", 2020, <<https://rsf.org/en/2020-world-press-freedom-index-entering-decisive-decade-journalism-exacerbated-coronavirus>>.

[Sandvine-2014] Sandvine, "Technology Showcase on Traffic Classification: Why Measurements and Freeform Policy Matter", 2014, <<https://www.sandvine.com/downloads/general/technology/sandvine-technology-showcases/sandvine-technology-showcase-traffic-classification.pdf>>.

[Satija-2021] Satija, S. and R. Chatterjee, "BlindTLS: Circumventing TLS-based HTTPS censorship", 2021, <<https://sambhav.info/files/blindtls-foci21.pdf>>.

[Schoen-2007] Schoen, S., "EFF tests agree with AP: Comcast is forging packets to interfere with user traffic", 2007, <<https://www.eff.org/deeplinks/2007/10/eff-tests-agree-ap-comcast-forging-packets-to-interfere>>.

[Schone-2014] Schone, M., Esposito, R., Cole, M., and G. Greenwald, "Snowden Docs Show UK Spies Attacked Anonymous, Hackers", 2014, <<http://www.nbcnews.com/feature/edward-snowden-interview/exclusive-snowden-docs-show-uk-spies-attacked-anonymous-hackers-n21361>>.

[Senft-2013] Senft, A., "Asia Chats: Analyzing Information Controls and Privacy in Asian Messaging Applications", 2013, <<https://citizenlab.org/2013/11/asia-chats-analyzing-information-controls-privacy-asian-messaging-applications/>>.

[Shbair-2015] Shbair, W.M., Cholez, T., Goichot, A., and I. Chrismont, "Efficiently Bypassing SNI-based HTTPS Filtering", 2015, <<https://hal.inria.fr/hal-01202712/document>>.

[SIDN2020] Moura, G., "Detecting and Taking Down Fraudulent Webshops at the .nl ccTLD", 2020, <<https://labs.ripe.net/Members/>>.

giovane.moura/detecting-and-taking-down-fraudulent-webshops-at-a-ccld>.

- [Singh-2019] Singh, K., Grover, G., and V. Bansal, "How India Censors the Web", 2019, <<https://arxiv.org/abs/1912.08590>>.
- [Sophos-2015] Sophos, "Understanding Sophos Web Filtering", 2015, <<https://www.sophos.com/en-us/support/knowledgebase/115865.aspx>>.
- [SSAC-109-2020] ICANN Security and Stability Advisory Committee, "SAC109: The Implications of DNS over HTTPS and DNS over TLS", 2020, <<https://www.icann.org/en/system/files/files/sac-109-en.pdf>>.
- [Tang-2016] Tang, C., "In-depth analysis of the Great Firewall of China", 2016, <<https://www.cs.tufts.edu/comp/116/archive/fall2016/ctang.pdf>>.
- [Thomson-2012] Thomson, I., "Syria Cuts off Internet and Mobile Communication", 2012, <http://www.theregister.co.uk/2012/11/29/syria_internet_blackout/>.
- [Tor-2020] The Tor Project, "Tor: Pluggable Transports", 2020, <<https://2019.www.torproject.org/docs/pluggable-transports.html.en>>.
- [Trustwave-2015] Trustwave, "Filter: SNI extension feature and HTTPS blocking", 2015, <https://www3.trustwave.com/software/8e6/hlp/r3000/files/1system_filter.html>.
- [Tschantz-2016] Tschantz, M., Afroz, S., Anonymous, A., and V. Paxson, "SoK: Towards Grounding Censorship Circumvention in Empiricism", 2016, <<https://oaklandsok.github.io/papers/tschantz2016.pdf>>.
- [Verkamp-2012] Verkamp, J.P. and M. Gupta, "Inferring Mechanics of Web Censorship Around the World", 2012, <<https://www.usenix.org/system/files/conference/foci12/foci12-final1.pdf>>.
- [Victor-2019] Victor, D., "Blizzard Sets Off Backlash for Penalizing Hearthstone Gamer in Hong Kong", 2019, <<https://www.nytimes.com/2019/10/09/world/asia/blizzard-hearthstone-hong-kong.html>>.
- [Villeneuve-2011] Villeneuve, N., "Open Access: Chapter 8, Control and Resistance, Attacks on Burmese Opposition Media",

2011, <<http://access.opennet.net/wp-content/uploads/2011/12/accesscontested-chapter-08.pdf>>.

[VonLohmann-2008] VonLohmann, F., "FCC Rules Against Comcast for BitTorrent Blocking", 2008, <<https://www.eff.org/deeplinks/2008/08/fcc-rules-against-comcast-bit-torrent-blocking>>.

[Wagner-2009] Wagner, B., "Deep Packet Inspection and Internet Censorship: International Convergence on an 'Integrated Technology of Control'", 2009, <<http://advocacy.globalvoicesonline.org/wp-content/uploads/2009/06/deeppacketinspectionandinternet-censorship2.pdf>>.

[Wagstaff-2013] Wagstaff, J., "In Malaysia, online election battles take a nasty turn", 2013, <<http://www.reuters.com/article/2013/05/04/uk-malaysia-election-online-idUKBRE94309G20130504>>.

[Wang-2017] Wang, Z., Cao, Y., Qian, Z., Song, C., and S. Krishnamurthy, "Your State is Not Mine: A Closer Look at Evading Stateful Internet Censorship", 2017, <https://www.cs.ucr.edu/~zhiyunq/pub/imc17_censorship_tcp.pdf>.

[Wang-2020] Wang, Z., Zhu, S., Cao, Y., Qian, Z., Song, C., Krishnamurthy, S., Chan, K., and T. Braun, "SYMTCP: Eluding Stateful Deep Packet Inspection with Automated Discrepancy Discovery", 2020, <https://www.cs.ucr.edu/~zhiyunq/pub/ndss20_symtcp.pdf>.

[Weaver-2009] Weaver, N., Sommer, R., and V. Paxson, "Detecting Forged TCP Packets", 2009, <<http://www.icir.org/vern/papers/reset-injection.ndss09.pdf>>.

[Whittaker-2013] Whittaker, Z., "1,168 keywords Skype uses to censor, monitor its Chinese users", 2013, <<http://www.zdnet.com/1168-keywords-skype-uses-to-censor-monitor-its-chinese-users-7000012328/>>.

[Wikip-DoS] Wikipedia, "Denial of Service Attacks", 2016, <https://en.wikipedia.org/w/index.php?title=Denial-of-service_attack&oldid=710558258>.

[Wilde-2012] Wilde, T., "Knock Knock Knockin' on Bridges Doors", 2012, <<https://blog.torproject.org/blog/knock-knock-knockin-bridges-doors>>.

[Winter-2012] Winter, P., "How China is Blocking Tor", 2012, <<http://arxiv.org/pdf/1204.0447v1.pdf>>.

[WP-Def-2020]

Wikipedia contributors, "Censorship", 2020, <<https://en.wikipedia.org/w/index.php?title=Censorship&oldid=943938595>>.

[Wright-2013] Wright, J. and Y. Breindl, "Internet filtering trends

in liberal democracies: French and German regulatory debates", 2013, <<https://policyreview.info/articles/analysis/internet-filtering-trends-liberal-democracies-french-and-german-regulatory-debates>>.

[Zhu-2011] Zhu, T., "An Analysis of Chinese Search Engine

Filtering", 2011, <<http://arxiv.org/ftp/arxiv/papers/1107/1107.3794.pdf>>.

[Zmijewski-2014] Zmijewski, E., "Turkish Internet Censorship Takes a

New Turn", 2014, <<https://blogs.oracle.com/internetintelligence/turkish-internet-censorship-takes-a-new-turn>>.

Authors' Addresses

Joseph Lorenzo Hall
Internet Society

Email: hall@isoc.org

Michael D. Aaron
CU Boulder

Email: michael.drew.aaron@gmail.com

Amelia Andersdotter

Email: amelia.ietf@andersdotter.cc

Ben Jones
Princeton

Email: bj6@cs.princeton.edu

Nick Feamster
U Chicago

Email: feamster@uchicago.edu

Mallory Knodel
Center for Democracy & Technology

Email: mknodel@cdt.org