A Survey of Worldwide Censorship Techniques

draft-hall-censorship-tech-00

Abstract

This document describes the technical mechanisms used by censorship regimes around the world to block or degrade internet traffic. It aims to make designers, implementers, and users of Internet protocols aware of the properties being exploited and mechanisms used to censor end-user access to information. This document makes no suggestions on individual protocol considerations, and is purely informational, intended to be 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 http://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 April 30, 2015.

Copyright Notice

Copyright (c) 2014 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 (http://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 Simplified BSD License text as described in Section 4.e of
the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

1. Introduction

2. Technical Aggregation

Aggregation is the process of figuring out what censors would like to block. Generally, censors aggregate "to block" information in three possible sorts of blacklists: Keyword, Domain Name, or IP. Keyword and Domain Name blocking take place at the application level (e.g. HTTP), whereas IP blocking tends to take place in the TCP/IP header. The mechanisms for building up these blacklists are varied. Many times private industries that sell "content control" software, such as SmartFilter, provide their services to nations which can then pick from broad categories, such as gambling or pornography, that they would like to block [ref-1]. In these cases, the private services embark on an attempt to label every semi-questionable website as to allow for this metatag blocking. Countries that are more interested in retaining specific political control, a desire which requires swift and decisive action, often have ministries or organizations, such as the Ministry of Industry and Information Technology in China or the Ministry of Culture and Islamic Guidance in Iran, which maintain their own blacklists.

3. Technical Identification

3.1. Points of Control

Digital censorship, necessarily, takes place over a network. Network design gives censors a number of different points-of-control where they can identify the content they are interested in filtering. 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. This requirement, the need to have the interception mechanism located somewhere, logically or physically, implicates four general points-of-control:

- 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 sniffing and 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.

- Internet Service Providers: Internet Service Providers are perhaps the most natural point-of-control. They have a benefit of being
easily enumerable by a censor paired with the ability to 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 cyber cafes can put filtration mechanisms in place. These mechanisms are occasionally at the request of a censor, but are more often implemented to help achieve institutional goals, such as to prevent the viewing of pornography on school computers.

- 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. The emergence of mobile devices exacerbate these feasibility problems.

At all levels of the network hierarchy, the filtration mechanisms used to detect undesirable traffic are essentially the same: a censor sniffs transmitting packets and identifies undesirable content, and then uses a blocking or shaping mechanism to prevent or degrade access. Identification of undesirable traffic can occur at the application, transport, or network layer of the IP stack. Censors are almost always concerned with web traffic, so the relevant protocols tend to be filtered in predictable ways. For example, a subversive image would always make it past a keyword filter, but the IP address of the site serving the image may be blacklisted when identified as a provider of undesirable content.

3.2. Application Layer

3.2.1. HTTP Request Header Identification

A HTTP header contains a lot of useful information for traffic identification; although host is the only required field in a HTTP request header, a HTTP method field is necessary to do anything useful. As such, the method and host fields are the two fields used most often for ubiquitous censorship. As a censor, I 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 TCP/IP identification 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 TCP/IP identification is necessary for filtering of HTTPS. Empirical Examples: Empirical examples of pure
HTTP Request Identification are unusually hard to identify due to the lack of distinguishing characteristics. Commercial technologies such as the McAfee SmartFilter and NetSweeper are often purchased by censors [ref-2]. These commercial technologies use a combination of HTTP Request Identification and TCP/IP Header Identification to filter specific URLs. There has not been research conducted to try and identify if only one of these two techniques is being used.

3.2.2. 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. Usually implemented at the Backbone or ISP level, the technique normally relies on mirroring, or duplicating the packets such that one can provide uninterrupted service while inspecting the duplicates for undesirable content, to prevent QoS degradation [ref-3]—the mirrored traffic is identified by relevant response fields (such as Server or Via). Tradeoffs: As with HTTP Request Header Identification, the techniques used to identify HTTP traffic are well-known, cheap, and relatively easy to implement, but is made useless by HTTPS, because the response in HTTPS is encrypted, including 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 can let a page momentarily load before blocking mechanisms kick in; giving the user a very clear indication that the censor is actively interfering with undesirable content. Empirical Examples: pointing to the "smoking-gun" examples in response header identification is difficult for the same reasons identifying requests is difficult. The best targeted evidence comes from a 2010 study conducted by Jong Park at the University of New Mexico. The study strongly indicates HTTP Response Header Identification was being used as a censorship identification technique in China from August 2008-January 2009 [ref-4].

3.2.3. Search Engine Keyword Identification

While technically similar to a HTTP request filter, the pervasiveness of search engines blacklisting search terms warrants its own attention. Search Engine Keyword Identification differentiates itself from other keyword identification techniques by being controlled by the company managing the search engine. Identification can be regional or worldwide. Implementation is occasionally voluntary, but normally is based on laws and regulations of the country a search engine is operating in. The keyword blacklists are
most likely maintained by the search engine provider. Tradeoffs: Search Engine Keyword Identification is an inconvenience as opposed to a hard block. As around half of all web traffic comes from search [ref-5], disrupting the flow of users to undesirable content is an effective method to redirect non-dedicated, curious users to less subversive content. It is also likely an effective method at encouraging self-censorship (see below) around the blocked content.  

Empirical Examples: Search Engine Keyword Identification is one of the easiest mechanisms to detect given the clear indicators, such as a specialized or blank results, paired with a trivial enumeration mechanism. China requires search engine providers to "voluntarily" maintain search term blacklists to acquire/keep an ICP license [ref-6]. It is clear these blacklists are maintained by each search engine provider based on the slight variations in the intercepted searches [ref-7][ref-8]. The United Kingdom has been pushing search engines to self censor with the threat of litigation if they don't do it themselves: Google and Microsoft have agreed to block more than 100,00 queries in U.K. to help combat abuse [ref-9][ref-10].

3.2.4. Deep Packet Inspection (DPI) Identification

Deep Packet Inspection has become computationally feasible as a censorship mechanism in the past 5 years [ref-11]. DPI differs from other filtration techniques in that it examine the application "data" section of traversing packets as opposed to only the header. To prevent substantial QoS impacts, DPI normally works by splitting the traffic, using either a mirror switch or fiber splitter, and analyzing a copy of the traffic. Keyword identification is often times used to flag undesirable content. Tradeoffs: While DPI can be employed across entire networks, it is one of the most expensive technical filtration mechanisms to implement and avoiding a large impact to QoS is difficult [ref-12]. Often times a targeted approach proves itself more feasible. Any encryption on the application level, such as HTTPS, also makes DPI useless as a censorship technique as the content typically analyzed is encrypted in this case. DPI, when paired with a keyword filter, can cause major overblocking problems if used indiscriminately. Empirical Evidence: Identifying deep packet inspection censorship is non-trivial; one must be sure that the undesirable content being filtered isn't being caught by simpler mechanisms before claiming more advanced DPI techniques are being used. The Tor project claims that China, Iran, Ethiopia, and others must be using DPI to block the obsf2 protocol [ref-13]. Malaysia has been accused of using target DPI, paired with DDoS, to identify and subsequently knockout pro-opposition material [ref-14]. 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 [ref-15].
3.3. Transport Layer

3.3.1. TCP/IP Header Identification

TCP/IP Header Identification is the most pervasive, reliable, and predictable type of identification. TCP/IP 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 cto block undesirable content via IP blacklisting, but also allows a censor to identify the IP of the user making the request. Port is useful for whitelisting certain applications or forcing an HTTP proxy for non-technical users. Trade-offs: This method of filtration is popular due to its simplicity, relative cheapness, and wide availability. It is trivial to implement a filtration mechanism at the Backbone, ISP, or Institutional level that compares the IP address of a packet with a blacklist of IP addresses. IP blocking is relatively crude, often leading to overblocking, and one of the simplest to circumvent via VPN or proxy as those either mask transport protocol within a tunnel or reroute data that might have been blocked otherwise. Port blocking is semi-effective at best. A censor can block communication on the default port of an undesirable application (for example uTorrent defaults to 32459), but almost all applications allow the user to change ports. Port whitelisting, where a censor only allow communication on approved ports, such as 80 for HTTP traffic, is more often used. This identification mechanism is often used in conjunction with HTTP Identification. Empirical Examples: TCP/IP Header Identification is pervasive. Some form of TCP/IP Header Identification is used by most, if not all, ISP and backbone censors. Any time an IP blacklist is being used, TCP/IP Header Identification is probably the technique being used to match the request against the blacklist. The examples of TCP/IP Header Identification are too numerous to enumerate in any meaningful way.

3.3.2. Protocol Identification

Protocol identification is a network analysis technique where one attempts to identify the protocols being used based on a variety of traffic characteristics. There have been a number well documented cases where traffic identification has been used to filter undesirable protocols. A very simple example of traffic identification would be to recognize all TCP traffic over port 80 as HTTP, but much more sophisticated methods, such as analyzing statistical properties of payload data and flow behavior, have been used [ref-16][ref-17]. Trade-offs: Protocol Identification necessarily only provides insight into the way information is traveling, and not the information itself. This can lead to massive
overblocking problems if used with popular protocols. Most often undesirable protocols are those which can be used to transmit information that is otherwise hard to analyze or considered to likely carry undesirable information; VoIP, P2P, SSL, and Tor have all been targets of protocol identification in the past. As statistical analysis is used, the methods tend to be expensive, both computationally and financially, and are occasionally imprecise and under-filter obfuscated protocols. Empirical Examples: Protocol Identification is easy to prove given the ubiquitous nature of the throttling/interruption; If only a specific protocol(s) are being prevented, then Protocol Identification is the most likely culprit. Iran censors have used Protocol Identification to identify and throttle SSH traffic by such a large amount as to make it unusable [ref-18]. The method used by censors in China to identify Tor connections could also be viewed as a type of Protocol Identification[ref-19]. Protocol Identification has also been used by industry from traffic management, such as the 2007 case where Comcast in the United States was using RST injection to interrupt BitTorrent Traffic [ref-20].

4. Technical Prevention

4.1. 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 overblocking all content from otherwise allowable IP’s 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 [ref-21]. 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 prevention 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 uses packet dropping as one of its primary mechanisms of technical censorship [ref-22]. Iran also uses Packet Dropping as the mechanisms for throttling SSH [ref-23]. These are but two examples of a ubiquitous censorship practice.
4.2. RST Packet Injection

Packet injection, generally, refers to a 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. Trade-offs: RST Packet Injection has a few advantages that make it extremely popular as a censorship technique. RST Packet Injection is an out-of-band prevention mechanism, allowing the avoidance of 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[ref-24]. 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[ref-25]. 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 sequence number. Sequence number is the hardest to get correct, as RFC 793 specifies an RST Packet should be in-sequence to be accepted, although the RFC also recommends allowing in-window packets as "good enough"[ref-26]. This in-window recommendation is important, as if it is implement it allows for successful Blind RST Injection attacks[ref-27]. When in-window sequencing is allowed, it is trivial to conduct a Blind RST Injection, a blind injection implies the censor doesn’t know any sensitive (encrypted) sequencing information about the TCP stream they are injecting into, they can simply enumerate the ~70000 possible windows; this is particularly useful for interrupting encrypted/obfuscated protocols such as SSH or Tor. 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. 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[ref-28], this later led to a FCC ruling against Comcast[ref-29]. China has also been known to use RST Packet Injection for censorship purposes. This prevention is especially evident in the interruption of encrypted/obfuscated protocols, such as those used by Tor[ref-30].
4.3. DNS Cache Poisoning

DNS Cache Poisoning refers to a mechanism where a censor interferes with the response sent by a DNS resolver to the requesting device by injecting an alternative IP address on the return path. Cache poisoning occurs after the requested site’s name servers resolve the request and attempt to forward the IP back to the requesting device; on the return route the resolved IP is recursively cached by each DNS servers that initially forwarded the request. During this caching process if an undesirable keyword is recognized, the resolved IP is poisoned and an alternative IP is returned. These alternative IP’s usually direct to a nonsense domain or a warning page.[ref-31]. Alternatively, Iranian censorship appears to prevent the communication en-route, preventing a response from ever being sent.[ref-32]. Trade-offs: DNS Cache Poisoning is one of the rarer forms of prevention due to a number of shortcomings. DNS Cache Poisoning requires the censor to force a user to traverse a controlled DNS resolver for the mechanism to be effective, it is easily circumvented by a technical savvy user that opts to use alternative DNS resolvers, such as the 8.8.8.8/8.8.4.4 public DNS resolvers provided by Google. DNS Cache Poisoning also implies returning an incorrect IP to those attempting to resolve a domain name, but the site is still technically unblocked if the user has another method to acquire the IP address of the desired site. Blocking overflow has also been a problem, as occasionally users outside of the censors region will be directed through a DNS server controlled by a censor, causing the request to fail. The ease of circumvention paired with the large risk of overblocking and blocking overflow make DNS Cache Poisoning a partial, difficult, and less than ideal censorship mechanism. Empirical Evidence: DNS Cache Poisoning, when properly implemented, is easy to identify based on the shortcomings identified above. Turkey relied on DNS Cache Poisoning for its country-wide block of websites such Twitter and Youtube for almost week in March of 2014 but the ease of circumvention resulted in an increase in the popularity of Twitter until Turkish ISP’s implementing an IP blacklist to achieve the governmental mandate.[ref-33]. To drive proverbial "nail in the coffin" Turkish ISPs started hijacking all requests to Google and Level 3’s international DNS resolvers [ref-34]. DNS Cache Poisoning, when incorrectly implemented, has as 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 Cache Poisoning attempt; this incident is thought to be the largest internet-service outage in history [ref-35][ref-36]. 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 [ref-37].
4.4. Distributed Denial of Service (DDoS)

Distributed Denial of Service attacks are a common attack mechanism used by "hacktivists" and black-hat hackers, but censors have used DDoS in the past for a variety of reasons. There is a huge variety of DDoS attacks[ref-38], but on 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 censorship technique. 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 [ref-39][ref-40]. Dissenting opinion websites are frequently victims of DDoS around politically sensitive events in Burma [ref-41]. Controlling parties in Russia[ref-42], Zimbabwe[ref-43], and Malaysia[ref-44] have been accused of using DDoS to interrupt opposition support and access during elections.

4.5. Network Disconnection or Adversarial Route Announcement

Network Disconnection or Adversarial Route Announcement The crudest of all censorship techniques, 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 cut off in a region when a censoring body withdraws all of the 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 with all the benefits the internet brings; this is never a long-term solution for any rational 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 [ref-45]. China disconnected the network
in the Xinjiang region during unrest in 2009 in an effort to prevent the protests from spreading to other regions [ref-46]. The Arab Spring saw the most frequent usage of Network Disconnection, with events in Egypt and Libya in 2011 [ref-47][ref-48], and Syria in 2012 [ref-49].

5. Non-Technical Aggregation

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 blacklists in that is 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 [ref-50]; more commonly manual filtering occurs on an institutional level. ICP’s, 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 ICP’s being held liable for the offending content by the Chinese government [ref-51].

6. Non-Technical Prevention

6.1. Self Censorship

Self censorship is one of the most interesting and effective types of censorship; a mix of Bentham’s Panopticon, cultural manipulation, intelligence gathering, and meatspace enforcement. Simply put, self censorship is when a censor creates an atmosphere where users censor themselves. This can be achieved through controlling information, intimidating would-be dissidents, swaying public thought, and creating apathy. Self censorship is difficult to document, as when it is implemented effectively the only noticeable tracing is a lack of undesirable content; instead one must look at the tools and techniques used by censors to encourage self-censorship. Controlling Information relies on traditional censorship techniques, or by forcing all users to connect through an intranet, such as in North Korea. Intimidation is often achieved through allowing internet users to post "whatever they want", but arresting those who post about dissenting views, this technique is incredibly common[ref-52][ref-53][ref-54][ref-55][ref-56]. A good example of swaying public thought is China’s "50-Cent Party", composed of somewhere between 20,000[ref-57] and 300,000[ref-58] contributors who are paid to "guide public thought" on local and regional issues as
directed by the Ministry of Culture. Creating apathy can be a side-effect of successfully controlling information over time and is ideal for a censorship regime [ref-59].

6.2. Domain Name Reallocation

As Domain Names are resolved recursively, if a TLD deregisters a domain all other DNS resolvers will be unable to properly forward and cache the site. Domain name registration is only really a risk where undesirable content is hosted on TLD controlled by the censoring country, such as .ch or .ru [ref-60].

6.3. Server Takedown

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 the hosting provider can be required to prevent access [ref-61].

7. References


Internet-Draft A Survey of Worldwide Censorship Techniques October 2014


[ref-54] Hopkins, C., ""Communications Blocked in Libya, Qatari Blogger Arrested: This Week in Online Tyranny"", March 2011, <http://readwrite.com/2011/03/03/communications_blocked_in_libya_this_week_in_online>.


Authors’ Addresses

Joeseph L. Hall
Center for Democracy and Technology

Email: jhall@cdt.org

Michael D. Aaron
Center for Democracy and Technology

Email: maaron@cdt.org