How the Great Firewall of China is Blocking Tor

Philipp Winter and Stefan Lindskog
Karlstad University
{*philwint, steflind*}*@kau.se*

Abstract

Internet censorship in China is not just limited to the web: the Great Firewall of China prevents thousands of potential Tor users from accessing the network. In this paper, we investigate how the blocking mechanism is implemented, we conjecture how China's Tor blocking infrastructure is designed and we propose circumvention techniques. Our work bolsters the understanding of China's censorship capabilities and thus paves the way towards more effective circumvention techniques.

1 Introduction

On October 4, 2011 a user reported to the Tor bug tracker that unpublished bridges stop working after only a few minutes when used from within China [17]. Bridges are unpublished Tor relays and their very purpose is to help censored users to access the Tor network if the "main entrance" is blocked [4]. The bug report indicated that the Great Firewall of China (GFC) has been enhanced with the potentiality of dynamically blocking Tor.

This censorship attempt is by no means China's first attempt to block Tor. In the past, there have been efforts to block the website [27], the public Tor network [24, 22] and parts of the bridges [18]. According to a report [27], these blocks were realised by simple IP blacklisting and HTTP header filtering. All these blocking attempts had in common that they were straightforward and inflexible.

In contrast to the above mentioned censorship attempts, the currently observable block appears to be much more flexible and sophisticated. The GFC blocks bridges dynamically without simply enumerating their IP addresses and blacklisting them (cf. [7]).

In this paper, we try to deepen the understanding of the infrastructure used by the GFC to block the Tor anonymity network. Our contributions are threefold: (1) we reveal how users within China are hindered from accessing the Tor network, (2) we conjecture how China's Tor blocking infrastructure is designed and (3) we discuss and propose circumvention techniques.

We also point out that censorship is a fast moving target. Our results are only valid at the time of writing[1] and might—and probably will—be subject to change. Nevertheless, we believe that a detailed understanding of the GFC's current capabilities, a "censorship snapshot", is important for future circumvention work.

2 Related Work

In [28], Wilde revealed first crucial insights about the block of Tor traffic. Over a period of one week in December 2011, Wilde analysed how *unpublished* Tor bridges are getting scanned and, as a result, blocked by the GFC.

Wilde's results showed that when a Tor user in China establishes a connection to a bridge or relay, deep packet inspection (DPI) boxes *identify* the Tor protocol. Shortly after a Tor connection is detected, *active scanning* is initiated. The scanning is done by seemingly random Chinese IP addresses. The scanners connect to the respective bridge and try to *establish a Tor connection*. If it succeeds, the bridge is *blocked*.

Wilde was able to narrow down the suspected cause for active scanning to the cipher list sent by the Tor client inside the TLS client hello[2]. This cipher list appears to be *unique and only used by Tor* although for a long time it was identical to the cipher list advertised by Firefox 3. That gives the GFC the opportunity to easily identify Tor connections. Furthermore, Wilde noticed that active scanning is started at the beginning of multiples of 15 minutes. An analysis of the Tor debug logs yielded that Chinese scanners initiate a TLS connection, conduct a renegotiation and start building a Tor circuit, once the TLS connection was set up. After the scan succeeded, the

[1] The data was mostly gathered in March 2012 and the paper written in April 2012.

[2] The TLS client hello is sent by the client after a TCP connection has been established. Details can be found in the Tor design paper [5].

IP address together with the associated port (we hereafter refer to this as "IP:port tuple") of the freshly scanned bridge is blocked resulting in users in China not being able to use the bridge anymore.

With respect to Wilde's contribution, we (1) revisit certain experiments in greater detail and with significantly more data, we (2) rectify observations which changed since Wilde's analysis, and we (3) answer yet open questions.

3 Experimental Setup

During the process of preparing and running our experiments we took special care to not violate any ethical standards and laws. In addition, all our experiments were in accordance with the terms of service of our service providers. In order to ensure reproducibility and encourage further research, we publish our gathered data and developed code[3]. The data includes Chinese IP addresses which were found to conduct active scanning of our bridge. We carefully configured our Tor bridges to remain unpublished and we always picked randomly chosen high ports to listen to so that we can be sure that the data is free from legitimate Tor users.

3.1 Vantage Points

In order to ensure a high degree of confidence in our results, we used different vantage points. We had a relay in Russia, bridges in Singapore and Sweden and clients in China. There is, however, no technical reason why we chose Russia, Singapore and Sweden.

Bridge in Singapore: A large part of our experiments was conducted with our Tor bridge located in Singapore. The bridge was running inside the Amazon EC2 cloud [1, 23]. The OpenNet Initiative reports Singapore as a country conducting minimal Internet filtering involving only pornography [9]. Hence, we assume that our experimental results were not interfered with by Internet filtering in Singapore.

Bridge in Sweden: To reproduce certain experiments, we set up Tor bridges located at our institution in Sweden. Internet filtering for these bridges was limited to well-known malware ports, so we can rule out filtering mechanisms interfering with our results.

Relay in Russia: A public Tor relay located in a Russian data center was used to investigate the type of block, public Tor relays are undergoing. The relay served as a middle relay, meaning that it is not picked as the first hop in a Tor circuit and it does not see the exit traffic of users.

Clients in China: To avoid biased results, we used two types of vantage points inside China: open SOCKS proxies and a VPS. We compiled a list of public Chinese SOCKS proxies by searching Google. We were able to find a total of 32 SOCKS proxies which were distributed amongst 12 distinct autonomous systems. We connected to these SOCKS proxies from computers outside of China and used them to rerun certain experiments on a smaller scale to rule out phenomena limited to our VPS.

Our second vantage point and primary experimental machine is a VPS we rented. The VPS ran Linux and resided in the autonomous system number (ASN) 4808. We had full root access to the VPS which made it possible for us to sniff traffic and conduct experiments below the application layer. Most of our experiments were conducted from our VPS, whereas the SOCKS proxies' primary use was to verify the results.

3.2 Shortcomings

Active analysis of a censorship system can easily attract the censor's attention if no special care is taken to "stay under the radar". Due to the fact that China is a sophisticated censor with the potential power to actively falsify measurement results, we have to point out potential shortcomings in our experimental setup.

We have no reliable information about the owners of our public SOCKS proxies. Whois lookups did not yield anything suspicious but the information in the records can be spoofed. It might even be possible that the SOCKS proxies are operated by Chinese authorities. Second, our VPS was located in a data center where Tor connections typically do not originate. We also had no information about whether our service provider conducts Internet filtering and the type or extent thereof.

4 Analysis

4.1 How Are Bridges and Relays Blocked?

The first step in bootstrapping a Tor connection requires connecting to the directory authorities to download the consensus which contains all public relays. We noticed that seven of all eight directory authorities are blocked *on the IP layer*. These machines responded neither to TCP, nor to ICMP packets. One authority turned out to be reachable and it was possible for us to download the consensus. We have no explanation why this particular machine was unblocked.

After the consensus has been downloaded, clients can start creating a circuit. Using our Russian relay, we found out that when a client in China connects to a relay, the GFC lets the TCP SYN pass through but drops the SYN/ACK sent by the bridge to the client. The same happens when a client tries to connect to a blocked

[3] http://www.cs.kau.se/philwint/static/gfc/

bridge. However, clients are still able to connect to different TCP ports as well as ping the bridge. We believe that the reason for the GFC blocking relays and bridges by IP:port tuples rather than by IPs is to minimise collateral damage.

4.2 How Long Do Bridges Remain Blocked?

To answer this question, we started two Tor bridges on our machine in Singapore. Both Tor processes were private bridges and listening on TCP port 27418 and 23941, respectively. Both ports were chosen randomly.

In the next step, we made the GFC block both IP:port tuples by initiating Tor connections to them from our VPS in China. After both tuples were blocked, we set up iptables [16] rules on our machine in Singapore to whitelist our VPS in China to port 23941 and drop all other connections to the same port. That way, the tuple appeared unreachable to the GFC but not to our Chinese VPS. Port 27418 remained unchanged and hence reachable to the GFC. We then started monitoring the reachability of both Tor processes by continuously trying to connect to them using telnet from our VPS.

After approximately 12 hours, the Tor process behind port 23941 (which appeared to be unreachable to the GFC) became reachable again whereas connections to port 27418 still timed out and continued to do so. In our iptables logs we could find numerous connection attempts originating from Chinese scanners. This observation shows that once a Tor bridge has been blocked, it only remains blocked if Chinese scanners are able to *continuously connect* to the bridge. If they cannot, the block is *removed*.

4.3 Is the Public Network Reachable?

To verify how many public relays are reachable from within China, we downloaded the consensus published at February 23 at 08:00 (UTC). At the time, the consensus contained descriptors for a total of 2819 relays. Then, from our Chinese VPS we tried to establish a TCP connection to the Tor port of every single relay. If we were able to successfully establish a TCP connection we classified the relay as reachable, otherwise unreachable.

We found that our VPS could successfully establish TCP connections to 47 out of all 2819 (1.6%) public relays. We manually inspected the descriptors of the 47 relays, but could not find any common property which could have been responsible for the relays being unblocked. We checked the availability of the reachable relays again after a period of three days. Only one out of the original 47 unblocked relays was still reachable.

4.4 Where Does the Fingerprinting Happen?

We want to gain a better understanding of where the Chinese DPI boxes are looking for the Tor fingerprint. In detail, we tried to investigate whether the DPI boxes also analyse domestic and ingress traffic.

We used six open Chinese SOCKS proxies (in ASN 4134, 4837, 9808 and 17968) as well as six PlanetLab nodes (in ASN 4538 and 23910) to investigate domestic fingerprinting. We simulated the initiation of a Tor connection multiple times to randomly chosen TCP ports on our VPS, but could not attract any active scans.

Previous research confirmed that HTTP keyword filtering done by the GFC is bidirectional [2], i.e., keywords are scanned in ingress as well as in egress traffic. We wanted to find out whether this holds true for the Tor DPI infrastructure too. To verify that, we tried to initiate Tor connections to our Chinese VPS from our vantage points in Sweden, Russia and Singapore. Despite multiple attempts we were not able to attract a single scan.

The above mentioned results indicate that Tor fingerprinting is probably *not done in domestic traffic* and only with traffic going from *inside China to the outside world*. We believe that there are two reasons for that. First, fingerprinting domestic traffic in addition to international traffic would dramatically increase the amount of data to analyse since domestic traffic is believed to be the largest fraction of Chinese traffic [12]. Second, at the time of this writing there are no relays in China so there is no need to fingerprint domestic or ingress traffic. Tor usage in China means being able to connect to the outside world.

4.5 Where Are the Scanners Coming From?

To get extensive data for answering this question, we continuously attracted scanners over a period of 17 days ranging from March 6 to March 23. We attracted scanners by simulating Tor connections from within China to our bridge in Singapore[4]. To simulate a Tor connection, we developed a small tool whose sole purpose was to send the Tor TLS client hello to the bridge and terminate after receiving the response. We could also have used the original Tor client to do so, but our tool was much more lightweight which was helpful, given our resource-constrained VPS. After every Tor connection simulation, our program remained inactive for a randomly chosen value between 9 and 14 minutes. The experiment yielded *3295 scans of our bridge*. Our findings described below are based on this data.

[4] We reproduced this experiment with a bridge in Sweden and with open Chinese SOCKS proxies. Our findings were the same.

4.5.1 Scanner IP Address Distribution

We are interested in the scanner's IP address distribution, i.e., how often can we find a particular IP address in our logs? Our data exhibits two surprising characteristics:

1. More than half of all connections—1680 of 3295 (51%)—were initiated by *a single* IP address: 202.108.181.70.
2. The second half of all addresses is almost *uniformly distributed*. Among all 1615 remaining addresses, 1584 (98%) were unique.

The IP address 202.108.181.70 clearly stands out from all observed scanners. Aside from its heavy activity we could not observe any other peculiarities in its scanning behaviour. The whois record of the address states a company named "Beijing Guanda Technology Co.Ltd" as owner. We could only find a company named "Guanda Technology Amusement Equipment Co., Ltd" on the Internet. It is not clear whether this is the same company. However, as explained below, we have reason to believe that the scanner's IP addresses are spoofed by the GFC so the owner of the IP address, assuming that it even exists, might not be aware of the scanning activity.

Whois and reverse DNS lookups of all the seemingly random IP addresses suggested that the IP addresses were coming from ISP pools. For example, all valid reverse DNS lookups contained either the strings *adsl* or *dynamic*.

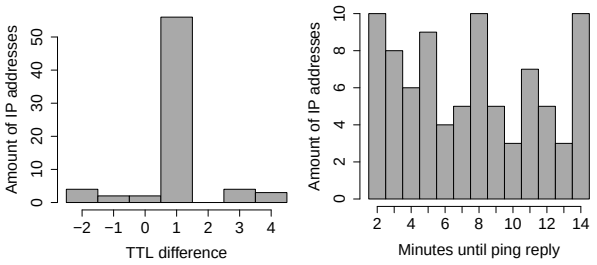
4.5.2 Autonomous System Origin

We used the IP to ASN mapping service provided by Team Cymru [15] to get the autonomous system number for every observed scanner. The result reveals that all scanners come from *one of three* ASes[5] with the respective percentage in parantheses:

- *AS4837*: CHINA169-BACKBONE CNCGROUP China169 Backbone (65.7%)
- *AS4134*: CHINANET-BACKBONE No.31,Jinrong Street (30.5%)
- *AS17622*: CNCGROUP-GZ China Unicom Guangzhou network (3.8%)

AS4134 is owned by China Telecom while AS4837 and AS17622 is owned by China Unicom. AS4134 and AS4837 are the two largest ASes in China [13] and play a crucial role in the country-wide censorship as pointed out by Xu, Mao and Halderman [29]. Furthermore, Roberts et al. [13] showed that China's AS level structure is *far from uniform* with a significant fraction of the countries traffic being routed through AS4134 or AS4837.

[5]Recent research efforts by Roberts et al. showed that China operates 177 autonomous systems [13].



(a) The IP TTL difference between after and during the scan.

(b) The amount of minutes until the hosts started replying to pings.

Figure 1: IP TTL difference (a) and duration until ping replies (b).

4.5.3 IP Address Spoofing

During manual tests we noticed that sometimes shortly after a scan, the respective IP address starts replying to pings[6], but with a *different IP Time-to-Live* (TTL) than during the scan. In order to have more data for our analysis, we wrote a script to automatically collect additional data as soon as a scanner connects and again some minutes afterwards. In particular, the script (1) runs TCP, UDP and ICMP traceroutes immediately after a scan and again 15 minutes later, (2) continuously pings the scanning IP address for 15 minutes and (3) captures all network traffic during these 15 minutes using tcpdump.

Between March 21 and 26 we started an independent experiment to attract scanners and let our script gather the above mentioned data. We caused a total of 429 scans coming from 427 unique IP addresses. From all 429 scans we then extracted all connections where the continuous 15 minutes ping resulted in at least one ping reply. This process yielded a subset of 85 connections which corresponds to approximately 20% of all observed connections. We analysed the 85 connections by computing the amount of minutes until the respective IP address started replying to our ping requests and the IP TTL difference (new TTL – old TTL) between packets during the scan and ping replies.

The results are shown in Figure 1(a) and 1(b). Figure 1(b) illustrates how long it took for the hosts to start replying to the ping requests. No clear pattern is visible. Figure 1(a) depicts all IP TTL differences after the scan (when the host starts replying to ping packets) and during the scan. We had 14 outliers with a TTL difference of 65 and 192 but did not list them in the histogram. It is clearly noticeable that the difference was mostly one, meaning that after the scan, the TTL was *by one more than during the scan*.

One explanation for the changing TTL, but definitely

[6]Note that this is never the case during or immediately after scans. All ICMP packets are being dropped.

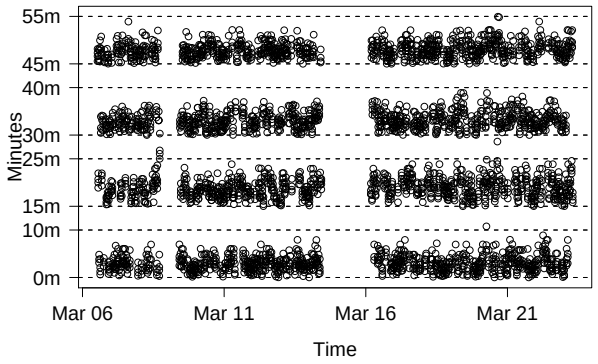


Figure 2: Time points when scanners were found connecting to our bridge.

not the only one, is that the GFC could be *spoofing IP addresses*. The firewall could be abusing several IP address pools intended for Internet users to allocate short-lived IP addresses for the purpose of scanning.

4.6 When Do the Scanners Connect?

Figure 2 visualises when the scanners in our data set connected. The *y*-axis depicts the minutes of the respective hour. Contrary to December 2011, when Wilde ran his experiments, the scanners now seem to use a broader time interval to launch the scans. In addition, the data contains two time intervals which are free from scanning. These intervals lasted from March 8 at around 16:30 to March 9 at 09:00 and from March 14 at 09:30 to March 16 at 3:30 (UTC). We have no explanation why the GFC did not conduct scanning during that time.

Closer manual analysis yielded that the data exhibits a *diurnal pattern*. In order to make the pattern visible, we processed the data as four distinct time series with every 15 minutes interval forming one time series, respectively. We smoothed the time series' data points using simple exponential smoothing with a smoothing factor $\alpha = 0.05$. The result—a subset of the data ranging from March 16 to March 23—is shown in Figure 3. Each of the four diagrams represents one of the 15 minutes intervals. The diagrams show that depending on the time of the day, on average, scanners connect either close to the respective 15 minutes multiple or a little bit later.

We conjecture that the GFC maintains *scanning queues*. When the DPI boxes discover a potential Tor connection, the IP:port tuple of the suspected bridge is added to a queue. Every 15 minutes, these queues are processed and all IP:port tuples in the queue are being scanned. We believe that during the day, the GFC needs more time to process the queues since there are probably more Chinese users trying to access the Tor network which leads to more scans.

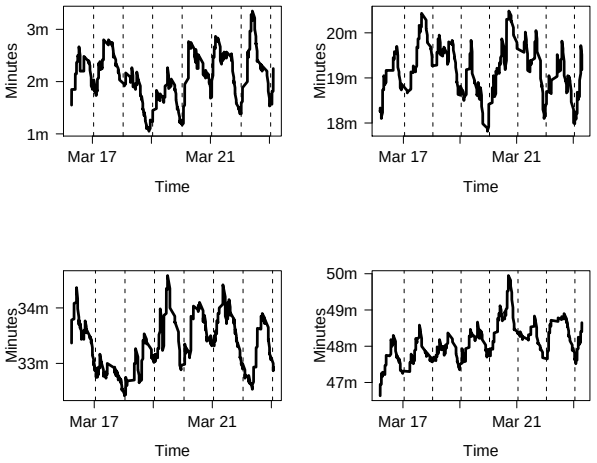


Figure 3: Diurnal scanning connection pattern.

4.7 Blocking Malfunction

During our experiments we noticed several sudden disappearences of active scanning. This lack of scanning made it possible for us to successfully initiate Tor connections without causing bridges to get scanned and blocked. The Tor bridge usage statistics between January and June 2012 contain several usage spikes which confirm outages in the blocking infrastructure [26]. Another downtime was observed by Wilde [19].

5 Circumvention

While there is a large body of work dedicated to anti-censorship, we limit this section to the discussion of the recently developed *obfsproxy* [21] and propose a novel way to evade the GFC's DPI boxes.

5.1 Obfsproxy

The Tor Project is developing a tool called obfsproxy. The tool runs independently of Tor and is obfuscating the network traffic it receives from the Tor process. As long as both, the bridge and the client, are running the tool, the Tor traffic transmitted between them can be obfuscated so that the Chinese DPI boxes are not able to identify the TLS cipher list anymore.

Obfsproxy implements a *pluggable transport layer* which means that modules can be written that support different types of obfuscation[7]. At the time of this writing, obfsproxy contains an obfuscation module called *obfs2* [20] which is based on Leidl's obfuscated ssh [6]. Obfs2 relies on a key establishment phase which is followed by the two involved parties exchanging superenciphered messages. A passive woman-in-the-middle is

[7] An overview of currently developed modules can be found at [25].

able to decrypt obfs2's obfuscation layer and reveal the encapsulated data[8] but this is more complex than conducting simple pattern matching as it is frequently done by DPI boxes.

As of March 24, the official obfsproxy bundle [21] contained a list of 13 hard-coded obfsproxy bridges in its configuration file. From our VPS we tested the reachability of all of these bridges by trying to connect to them via telnet. We found that not a single connection succeeded. One bridge seemed to be offline and the connection to the remaining 12 bridges was aborted by spoofed RST segments.

The above result raises the question whether the GFC is able to block all obfsproxy connections or just the 13 hard-coded bridges. To answer this question, we set up a private obfsproxy bridge in Sweden and connected to it from within China. We initiated several connections to it over several hours and we could always successfully establish a Tor circuit. We conclude that the IP:port tuples of the 13 hard-coded obfsproxy bridges were added to a blacklist to prevent widespread use of the official obfsproxy bundle. However, private obfsproxy bridges remain undetected by the GFC.

Similar to obfs2, Moghaddam et al. proposed a pluggable transport for Tor to mimic Skype video traffic [8]. This concept will make it significantly harder to block Tor by fingerprinting because DPI boxes would have to distinguish between legitimate and camouflaged Skype traffic.

5.2 Packet Fragmentation

One circumvention technique described by Ptacek and Newsham [11] is *packet fragmentation* which exploits the fact that some network intrusion detection systems do not conduct packet reassembly. Crandall et al. also considered fragmentation to evade the GFC's keyword-based detection [3].

We used the tool *fragroute* [14] to enforce packet fragmentation on our VPS in China. We configured fragroute to split the TCP stream to segments of 16 bytes each. In our test it took 5 TCP segments to transmit the fragmented cipher list to our bridge. Despite initiating several fragmented Tor connections, we never observed any active scanning and could use Tor without interference. This experiment indicates that the GFC does not conduct packet reassembly. A similar observation was made by Park and Crandall [10].

However, client side fragmentation is an unpractical solution given that this method must be supported by *all* connecting Chinese users. A single user who does not use fragmentation, triggers active scanning which then leads to the block of the respective bridge. Another disadvantage is the significant protocol overhead due to the shortened TCP segments which leads to a decrease in throughput.

Due to these shortcomings, we propose a way to realise *server side fragmentation*. We developed a tool[9] which transparently *rewrites the TCP window size* announced by the bridge to the client inside the SYN/ACK segment. The diminished window size makes the client split its TLS cipher list across two TCP segment. That way, the DPI boxes are not able to identify the Tor connection. At the time of this writing, the tool is already deployed to several bridges. So far, these bridges were not scanned and keep getting connections from legitimate users in China.

Our server side fragmentation tool has the advantage that it is easy to deploy and it only interferes with Tor connections by rewriting the announced TCP window during the TCP handshake. Hence, there are virtually no performance implications.

6 Conclusions

We showed how access to Tor is being denied in China and we conjectured how the blocking infrastructure is designed. In addition, we discussed countermeasures intended to "unblock" the Tor network. Our findings include that the Great Firewall of China might spoof IP addresses for the purpose of scanning Tor bridges and that domestic as well as ingress traffic does not seem to be subject to Tor fingerprinting. We also showed that the firewall is easily circumvented by fragmented packets. Tor traffic is currently distinguishable from what is regarded as harmless traffic in China. Since Tor is being used more and more as censorship circumvention tool, it is crucial that this distinguishability is minimised.

Acknowledgments

We thank the Tor developers for their helpful feedback and support, the anonymous reviewers for their valuable suggestions, Harald Lampesberger, Simone Fischer-Hübner and Rose-Mharie Åhlfeldt for feedback and Fabio Pietrosanti for helping with the experiments.

The work conducted by the second author has been supported by the Compare Business Innovative Centre phase 3 (C-BIC 3) project, funded partly by the European Regional Development Fund (ERDF).

Our raw data and code are available at:
http://www.cs.kau.se/philwint/static/gfc/.

[8] We note that this does not affect the security of the TLS connection used by Tor.

[9] The tool is available on our project website: http://www.cs.kau.se/philwint/static/gfc/

References

[1] AMAZON WEB SERVICES LLC. Amazon Elastic Compute Cloud (Amazon EC2). https://aws.amazon.com/ec2/ [Accessed: Jun. 29, 2012].

[2] CLAYTON, R., MURDOCH, S. J., AND WATSON, R. N. M. Ignoring the Great Firewall of China. In *Privacy Enhancing Technologies* (Cambridge, 2006), Springer, pp. 20–35.

[3] CRANDALL, J. R., ZINN, D., BYRD, M., BARR, E., AND EAST, R. ConceptDoppler: A Weather Tracker for Internet Censorship. In *Computer and Communications Security* (Alexandria, VA, 2007), ACM, pp. 352–365.

[4] DINGLEDINE, R., AND MATHEWSON, N. Design of a blocking-resistant anonymity system. Tech. rep., The Tor Project, 2006.

[5] DINGLEDINE, R., MATHEWSON, N., AND SYVERSON, P. Tor: The Second-Generation Onion Router. In *USENIX Security Symposium* (San Diego, CA, 2004), USENIX Association, pp. 303–320.

[6] LEIDL, B. obfuscated-openssh. https://github.com/brl/obfuscated-openssh/blob/master/README.obfuscation, [Accessed: Jun. 29, 2012].

[7] LING, Z., FU, X., YU, W., LUO, J., AND YANG, M. Extensive Analysis and Large-Scale Empirical Evaluation of Tor Bridge Discovery. In *International Conference on Computer Communications* (Orlando, FL, 2012), IEEE.

[8] MOGHADDAM, H. M., LI, B., DERAKHSHANI, M., AND GOLDBERG, I. SkypeMorph: Protocol Obfuscation for Tor Bridges. Tech. rep., University of Waterloo, 2012.

[9] OPENNET INITIATIVE. Singapore. http://opennet.net/research/profiles/singapore [Accessed: Jun. 29, 2012].

[10] PARK, J. C., AND CRANDALL, J. R. Empirical Study of a National-Scale Distributed Intrusion Detection System: Backbone-Level Filtering of HTML Responses in China. In *Distributed Computing Systems* (Genova, 2010), IEEE, pp. 315–326.

[11] PTACEK, T. H., AND NEWSHAM, T. N. Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection. Tech. rep., Secure Networks, Inc., 1998.

[12] ROBERTS, H. Local Control: About 95% of Chinese Web Traffic is Local. https://blogs.law.harvard.edu/hroberts/2011/08/15/local-control-about-95-of-chinese-web-traffic-is-local/ [Accessed: Jun. 29, 2012].

[13] ROBERTS, H., LAROCHELLE, D., FARIS, R., AND PALFREY, J. Mapping Local Internet Control. In *Computer Communications Workshop* (Hyannis, CA, 2011), IEEE.

[14] SONG, D. fragroute. http://monkey.org/~dugsong/fragroute/ [Accessed: Jun. 29, 2012].

[15] TEAM CYMRU, INC. IP to ASN Mapping. https://www.team-cymru.org/Services/ip-to-asn.html [Accessed: Jun. 29, 2012].

[16] THE NETFILTER.ORG PROJECT. netfilter/iptables project homepage. http://www.netfilter.org [Accessed: Jun. 29, 2012].

[17] THE TOR PROJECT. Bridge easily detected by GFW. https://trac.torproject.org/projects/tor/ticket/4185 [Accessed: Jun. 29, 2012].

[18] THE TOR PROJECT. China blocking Tor: Round Two. https://blog.torproject.org/blog/china-blocking-tor-round-two [Accessed: Jun. 29, 2012].

[19] THE TOR PROJECT. Knock Knock Knockin' on Bridges' Doors. https://blog.torproject.org/blog/knock-knock-knockin-bridges-doors [Accessed: Jun. 29, 2012].

[20] THE TOR PROJECT. obfs2 (The Twobfuscator). https://gitweb.torproject.org/obfsproxy.git/blob/HEAD:/doc/obfs2/protocol-spec.txt [Accessed: Jun. 29, 2012].

[21] THE TOR PROJECT. obfsproxy. https://www.torproject.org/projects/obfsproxy [Accessed: Jun. 29, 2012].

[22] THE TOR PROJECT. Picturing Tor censorship in China. https://blog.torproject.org/blog/picturing-tor-censorship-china [Accessed: Jun. 29, 2012].

[23] THE TOR PROJECT. Tor Cloud. https://cloud.torproject.org [Accessed: Jun. 29, 2012].

[24] THE TOR PROJECT. Tor partially blocked in China. https://blog.torproject.org/blog/tor-partially-blocked-china [Accessed: Jun. 29, 2012].

[25] THE TOR PROJECT. Tor: Pluggable Transports. https://www.torproject.org/docs/pluggable-transports, [Accessed: Jul. 1, 2012].

[26] THE TOR PROJECT. Tor users via bridges. https://metrics.torproject.org/users.html?graph=bridge-users&start=2012-01-01&end=2012-06-18&country=cn&dpi=72#bridge-users [Accessed: Jun. 29, 2012].

[27] THE TOR PROJECT. Torproject.org Blocked by GFW in China: Sooner or Later? https://blog.torproject.org/blog/torprojectorg-blocked-gfw-china-sooner-or-later [Accessed: Jun. 29, 2012].

[28] WILDE, T. Great Firewall Tor Probing Circa 09 DEC 2011. https://gist.github.com/da3c7a9af01d74cd7de7 [Accessed: Jun. 29, 2012].

[29] XU, X., MAO, Z. M., AND HALDERMAN, J. A. Internet Censorship in China: Where Does the Filtering Occur? In *Passive and Active Measurement Conference* (Atlanta, GA, 2011), Springer, pp. 133–142.