Towards a Comprehensive Picture of the Great Firewall's DNS Censorship

Anonymous
zion.vlab2@gmail.com

Abstract

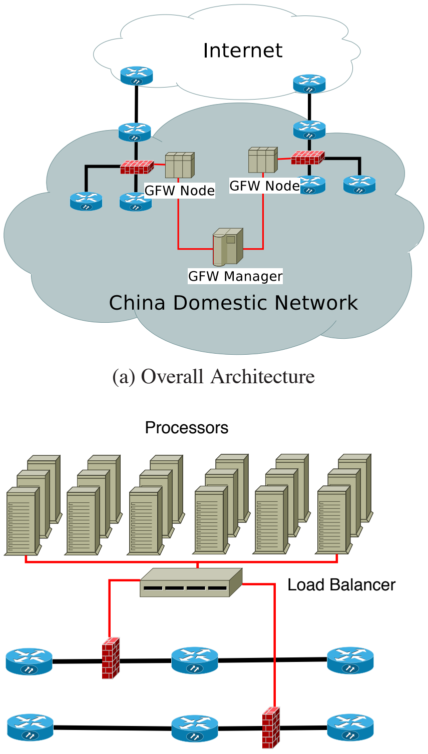
China's Great Firewall passively inspects network traffic and disrupts unwanted communication by injecting forged DNS replies or TCP resets. We attempted to comprehensively examine the structure of the DNS injector, using queries from both within and outside China. Using these probes, we were able to localize the DNS monitors' locations, extract the firewall's DNS blacklist of approximately 15,000 keywords, and estimate the cluster structure and active response rate by utilizing an information leakage in the Great Firewall's design.

1 Introduction

China's Great Firewall (GFW) passively inspects network traffic and disrupts unwanted communication by injecting forged DNS replies or TCP RSTs [1, 14, 15, 2, 6, 9, 11, 4, 13]. While numerous facets of the GFW's operation have seen prior study, the nature of its DNS censorship has not been previously examined in a *comprehensive* fashion.

In this work we undertake a range of systematic measurements to illuminate the workings of the GFW's DNS censorship. Our study employs DNS queries sent from both within and outside China to serve as probes to nominally nonresponsive addresses as well as DNS servers under our control. We in addition employ TTL-limited queries to determine topological information associated with the GFW's packet injectors. Finally, by leveraging the GFW's deterministic generation of IP TTL and IP ID fields for injected packets, we deduce relationships between individual GFW components.

In total our measurements cover all of the /24 subnets within China, as well as likely the strong majority of all domain names/keywords that China censors. We assess where in the network censorship occurs, which network regions operate free of censorship and how, which names the censor has decided to block and the patterns used to locate those names, how the blocklist evolves over time, the censor's use of load balancing and centralized management (per Figure 1), and estimate the volume of censored DNS queries. We amass strong evidence that the GFW performs DNS-based censorship essentially only at China's borders, using a blocklist of around 15,000 keywords. Individual GFW nodes appear to operate in clusters of several hundred processes that collectively inject censored responses at a rate of about 2,800 per second.



(a) Overall Architecture

(b) Structure of one GFW Node

Figure 1: The architecture of GFW as deduced from our study.

2 Background and Related Work

The Great Firewall operates in an *on-path* fashion: it passively examines passing traffic, but can only suppress communication by injecting additional packets. Injecting TCP RSTs can block individual connections, while fake DNS A record responses serve to block access by domain name; our study focuses on the second of these. We note that such injection works even when users employ third-party DNS resolvers outside the country, since the GFW will still react to the queries sent to those resolvers.

Reports of the GFW injecting DNS responses date to 2002 [7]. What began as a single poisoned response for all blocked domains evolved by 2007 into a level of keyword filtering (e.g., responding to "falungong" appearing anywhere in a domain name) and the use of at least 8 distinct addresses in injected replies [11]. WestChamber research developed fingerprints identifying injected DNS responses based on fields such as the IP ID, IP TTL and DNS TTL [2, 1]. They also confirmed the use of 8 "Bad IP" address, which enabled them to distinguish between injected packets and legitimate replies.

Previous work has demonstrated that the GFW does not distinguish traffic directionality [5], presumably to simplify configuration. This behavior results in collateral damage, where DNS resolvers outside of China, when contacting authoritative servers located in or at the end of paths that transit China, incur Chinese censorship on non-Chinese requests [4].

Two organizations monitor Chinese censorship of domains on an ongoing basis. `greatfire.org` has tested for blocking since 2011, reporting as of this writing 2,582 of their 22,525 monitored domains as blocked. `hikinggfw.org` began GFW monitoring in 2012, finding 1,638 domains in the Alexa Top 1 Million (Alexa 1M) blocked as of Feb 13th, 2014.

Previous studies have also looked at localizing GFW nodes [6, 15, 4], concluding that GFW nodes operate not only at the edge of China's Internet but also within its domestic networks, based on observations of packet injection occurring in non-border Chinese ASes. However, articles from the Chinese Internet community state that GFW deploys injecting nodes only at Internet exchange points focusing on international communications [1]. Thus, the issue of localization remains unclear.

3 Methodology

We extensively probed the GFW using techniques developed in previous measurements of GFW DNS injection [4]: either querying nominally unresponsive addresses with known censored domains, or issuing queries for non-existent domains that contain known censored keywords. When needed, we also manipulate the IP TTL to perform traceroute-like experiments, and use the King method [8] to trigger DNS queries indirectly by issuing requests to open resolvers for domain names under our control.

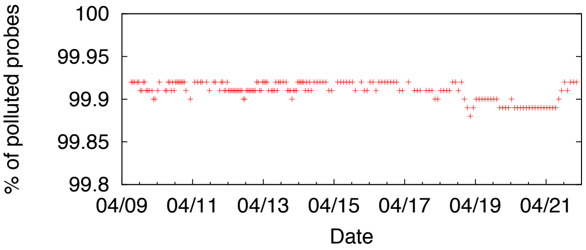


Figure 2: Percentage of DNS open resolvers inside China polluted by GFW measured over two weeks in 2014.

Accuracy Issues. The above methods generally work free of false positive: that is, because they base their operation on the use of either servers or domains that do not exist, or for which we know the correct response, any blocking-related answer clearly indicates action on the part of the GFW. However, a number of issues can cause false negatives, where we fail to receive a poisoned answer even though the query indeed matched one prohibited by the GFW. This can occur due to packet loss, rate-limiting, or overload of injectors. During our study we observed clear, non-negligible (roughly 0.5–2%) false negative rates, which could lead to confusing or misleading indications. We addressed this issue by repeating our tests to confirm previous results as much as possible within our resource constraints, and take this issue into consideration when formulating estimates and inferences. We also note the importance of the research community bearing this complication in mind when conducting similar experiments.

4 DNS Injection Effectiveness

We probed open DNS resolvers inside China to evaluate the effectiveness of the GFW. We located the resolvers by probing UDP port 53 for the entire IPv4 address space and then selecting those that we found consistently reachable, and that MaxMind's Geo-IP database located as in China, totaling about 150K. We then probed these resolvers every hour by querying them from within China for 3 blocked domains and 1 benign and popular domain (`www.qq.com`). In those (rare) cases that the benign name's query failed, we discarded the probes, as this indicated connectivity issues.

For our probes, if any of the other 3 domains elicited a "Bad IP" A record response then we treat the resolver as **polluted**. (Here, "Bad-IP" corresponds to any of the 174 IP addresses we found returned by GFW DNS injection in the experiments discussed below in Section 6.)

Otherwise, we term the resolver **clean**. We conducted this testing for 2 weeks, and per Figure 2 found that the GFW operates very effectively and comprehensively in terms of DNS pollution. Note that the high polluted percentage ($< 0.1\%$ clean rate) in Figure 2 does not conflict with the overall false negative rates (0.5–2%), because a resolver returns a clean response only when *both* its cache and all corresponding iterative rounds avoid pollution.

We noticed, however, that some open resolvers continuously provide correct answers for the blocked domains. By issuing queries to these resolvers for domains under our control, we determined that of the 78 such clean resolvers, 38 forward their queries to Google's public DNS and 4 forward to OpenDNS.

The others keep clean by dropping answers with "Bad IP" A records, which we confirmed by returning such records from our controlled domain (without any blocked keyword) and observing that the open resolver would not forward the reply back. Finally, one clean resolver, located in an Internet exchange point, does not show any apparent signs of attempting to evade DNS censorship. It instead appears that the resolver operates outside the range of GFW censorship; or the GFW employs a whitelist to ignore queries from certain resolvers.

5 Location of DNS Injectors

Previous studies indicated that DNS and TCP RST injection occurred not just near China's border but also within its domestic networks [6, 15, 3]. The Chinese online community, however, mostly believes that deployment occurs only at the edge [1]. We aim to clarify this issue with large-scale DNS probing from both internal and external vantage points.

Experiments. To identify the router interfaces monitored by the GFW nodes—the "injecting interfaces"—we first selected an unresponsive IP address from each /24 subnets in China. We then performed TTL-limited (traceroute-like) probes to these addresses from both an external server and from two servers in different Chinese ISPs. Only probes which pass a monitor point will trigger a response.

In addition, we identified 207 DNS open resolvers in 34 Chinese ASes: these enabled us to utilize the King method to indirectly scan the paths between these resolvers, as follows. To scan the path between *A* and *B*, we first send a non-censored query to *A* for a domain we control. The authority for this domain directs *A* to next query *B*. As long as the record pointing to *B* remains cached at *A* for a short time (which we confirmed), we can use the cached entry to probe the path between *A* and *B* by querying *A* for a name that we prepend to a name that the GFW blocks (a *P2*-type pattern per Section 6).

AS No.	Interface #	/24 Subnet #
4134	4,169	569,978
4837	596	276,286
9808	95	76,132
4538	17	46,984
9394	788	8,393
4812	931	4,241
7497	4	3,543
9929	5	2,173
4847	185	1,906
Others (7 ASes)	303	1,011
Unknown	6	59
Unknown (no ICMP at injecting hop)		11,526
No GFW packets observed		257,698

Table 1: ASes associated with DNS injecting interfaces, as seen by TTL-limited scans from outside China. A total of 1,259,930 /24s observed 7,099 distinct interfaces.

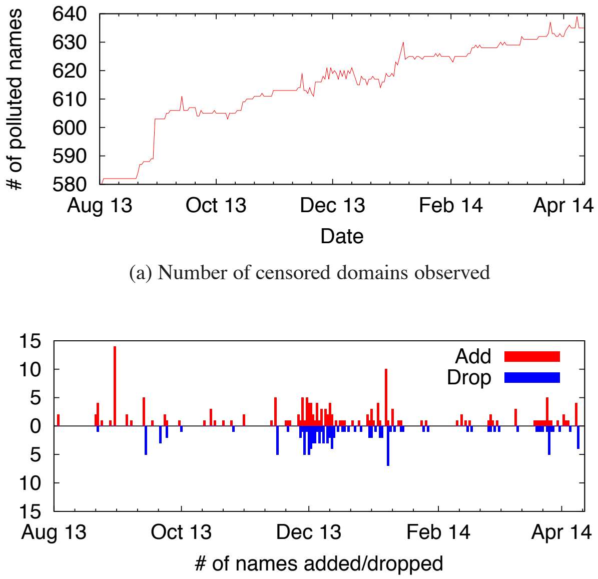
Results. Table 1 summarizes the results for probes from outside China. About 1 million ($\approx 80\%$) of the paths scanned exhibit GFW pollution, among which we identified 7,099 injecting interfaces belonging to 16 border ASes of China (using the classification in [15]). In most cases, the injecting interface manifested at either 2 (18.3%) or 3 (54.6%) hops inside China.

It is important to note that our results may overestimate the GFW injector locations due to the problem of false negatives, as mentioned in Section 3. In our experiment, intermittent responses will lead to a false conclusion that the GFW monitoring occurs deeper within the Chinese Internet than it actually does, leading to false additions. To reduce this effect we repeated each probe five times.

We see from Table 1 that even with potential false additions, very few ASes deploy DNS injectors. In fact, of 7,099 detected injecting interfaces, only about 270 appeared in probes covering 100 or more /24s. Thus we suspect the strong majority of the 7K interfaces reflect false additions, and the actual total lies in the range of a few hundred.

We also to our surprise find that 20% of /24s did not elicit any GFW responses. This may reflect geolocalization errors in associating those subnets with China.

In sharp contrast to our external probing, when conducting our probes from two ISPs in China we rarely observe DNS injection (on only about 16% of the paths probed from one ISP and 4% from the other). Examining the paths that did experience injection in most cases reveals that the responsible interface reflects a path that either traversed links previously observed as hosting GFW injectors, or within the same /24 subnet as such injectors. When employing the King method for 207 open resolvers covering 42,849 domestic routing paths, we find



(a) Number of censored domains observed

(b) Daily variations

Figure 3: Observing GFW policy changes by monitoring Alexa 1M domains from Aug. 2013 to Apr. 2014.

only 18 paths (0.04%) exhibited pollution.

These results highlight that the GFW appears to deploy DNS injection nodes only at the edge of China's Internet, generally within a few hops of an international transit point. Such deployment can still lead to a small degree of domestic traffic experiencing GFW censorship due to routing policies that generate boomerang routes.

6 The GFW's DNS Injection Policy

We now turn to discovering the ruleset the GFW uses to govern when to inject DNS replies (which domains, or more precisely which rules, its policy list contains), and assessing how the management of this ruleset evolves over time.

To do so, we issued DNS queries towards non-existent addresses to trigger GFW responses. These queries included both all domains present on the Alexa 1M list (obtained in August 2013) and those in the zone files for `.com`, `.net`, `.org`, and `.info` (obtained in 2011). We also added `www.` to any names that did not already have that as a prefix, since we observed that the GFW sometimes blocks only these refinements (e.g., `www.nytimes.com` vs. `nytimes.com`). In total, we queried about 130 million names.

We first conducted long-term daily monitoring on the Alexa 1M site list from August 1 2013 to April 12 2014, to detect changes in blocking patterns. As shown in Figure 3a, the number of censored domain names increased by about 10% over our eight months of monitoring. Figure 3b presents the daily progression of the censor adding and dropping names. Some of the deletions however may reflect false negatives. Thus, Figure 3a more accurately captures the overall trend.

The more volatile nature of Figure 3b naturally raises concern about measurement noise. We manually verified several cases, such as the peak on August 30th, which occurred when a group of porn sites became blocked. The dynamics convey that the GFW censors actively uncover and censor new unwanted domain names, but do not particularly attend to unblocking. Indeed, to our surprise we found that more than two-thirds of the censored domains had expired registrations (these are primarily from the zone-file probing we conducted, as discussed next).

In addition, we probed the full domain set (all 130M names) in April 2014 to detect blocked names. For these blocked names we used binary search to extract the actual keyword that triggers injection. For each keyword we then determined where it must appear by testing instances where we added random prefixes and suffixes. We find that some keywords must appear only as a suffix, while others match anywhere in a request.

This testing discovered 35,332 blocked domain names, from which we extracted 14,495 associated keywords. Due to the breadth of our seed list, we believe it likely that these measurements cover a large portion of the GFW's entire DNS policy. Table 2a gives the 10 keywords that caused most blocked domain names.

Our probing uncovered several distinct matching patterns (Table 2b). We suspect these variations reflect a legacy of long-term, non-deliberative management, including inconsistent use of name separators ('.') in rule creation and optionally including an ending wildcard in a rule. Previous studies highlight the GFW targeting any domain name that *contains* a keyword [11, 4]. Since then, the matches appear to have grown more precise, with the bulk of the rules now requiring that the request *end* in an exact match. Such rules can still cause significant overblocking (such as blocking of `purefacebook.com`).

We also performed the keyword tests for two destinations (i.e., two different routing paths and consequently two separate GFW nodes) to verify the results. After employing repeated tests, we obtained equivalent results, suggesting that at least those two GFW locations have a synchronized policy.

Finally, we note that our long term monitoring did not observe evidence of the Jan. 21 2014 DNS poisoning event [10], which some believe arose due to a misconfiguration in blocking rules [12].

Keyword	Pattern No.	Domain #
facebook.com	P1	8,933
twitter.com	P1	4,702
youtube.com	P1	2,907
was.info	P2	509
33a.com	P1	383
88sf.com	P1	242
appspot.com	P2	239
kproxy.com	P1	139
mefans.com	P2	110
sf888.com	P1	99
Total		35,332

(a) Top 10 keywords

Pattern No.	Tested Patterns					# Entries
	kw	{rnd}kw	{rnd}.kw	kw{rnd}	kw.{rnd}	
P1	✓	✓	✓	✗	✗	10,461
P2	✓	✓	✓	✓	✓	2,803
P3	✓	✗	✓	✗	✗	856
P4	✓	✗	✓	✓	✓	335
P5	✗	✗	✓	✗	✗	38
P6	✓	✗	✗	✗	✗	1
P7	✗	✗	✓	✓	✓	1
Total						14,495

(b) Matching Patterns

Table 2: Measured GFW DNS censorship policy.

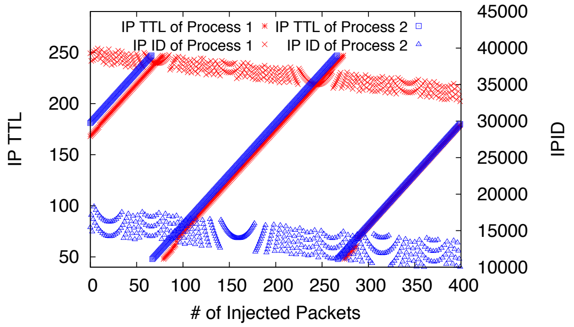


Figure 4: IP TTL and ID seen in DNS packets generated by two separate processes within the same GFW node. The third pair of lines for ID are nearly identical.

7 Internal Structure of GFW Nodes

We probed the structure of an individual GFW node by analyzing side-channels present in the packets it injects. Previous studies found patterns in the IP ID and IP TTL fields of TCP RSTs injected by GFW [1, 2], and we discovered similar deterministic patterns in injected DNS packets.[1]

From outside of China, we selected one /24 China subnet to which we issued a series of sensitive DNS queries. In most cases, these elicited two injected responses: one with a fixed IP ID (28944) and initial IP TTL (likely 64), and the other manifesting clear patterns in how its IP ID and IP TTL fields varied. For the latter, the TTLs sequentially increased while the IDs appear generated by a quadratic function (Figure 4). After several rounds of probes, we identified lower and upper bounds for the latter injected responses of a TTL between 48 and 247.

Mapping Interfaces to Nodes. Our probes often encountered routers employing load-balancing strategies: probes to different destinations within the same /24, or merely to the same destination but with different ports, could result in different paths and consequently different injecting interfaces. We probed all addresses of a selected subnet and some of its sibling /24 subnets with the same AS-level path. In total we identified 4 distinct injecting interfaces. The routing load-balancing strategy of the tested network was consistent during our experiments.

The TTL side-channel enabled us to isolate individual GFW processes. We identified pairs of addresses sharing the TTL counter, indicating that the same process handled both censorship requests. On this basis, we determined that the same GFW node actively monitored multiple interfaces.

We first divided all probed addresses into 4 groups according the injecting interfaces. We then selected a pair of addresses from two different groups, issuing DNS queries to each in turn. If all TTLs and IDs in the injected responses followed the same patterns, we concluded that a single GFW node monitored both interfaces, and that the monitoring for those individual addresses happened within the same process.

Figure 5 shows a typical case of a correlated pair. The IP TTLs for the two addresses increase in tandem, forming a consistent pattern. The IP IDs for the two addresses stay within the same band, but they appear to use a different function to determine the location within the band. Using this technique we found correlations between all four interfaces, suggesting that a single GFW node monitors all of the interfaces.

Load Balancing and Volume. Figure 4 also suggests that the GFW load-balances between processes based on IP address, as the two different destination addresses exhibited different TTL and ID counters. We attempted to determine the fields used for load balancing, the algorithm, and an estimate of the number of processes behind the load balancer.

[1] We have altered minor details in our discussion to defend against the censor identifying us or those who assisted our measurements.

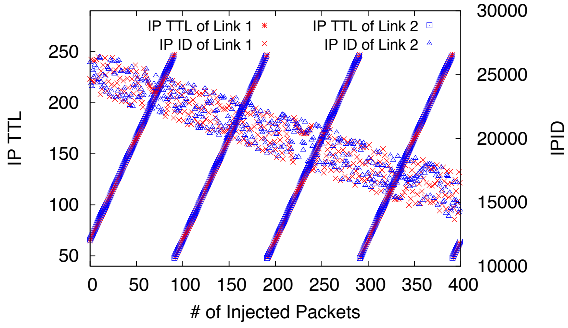


Figure 5: The injected TTL and IP IDs from the same process monitoring two distinct network links.

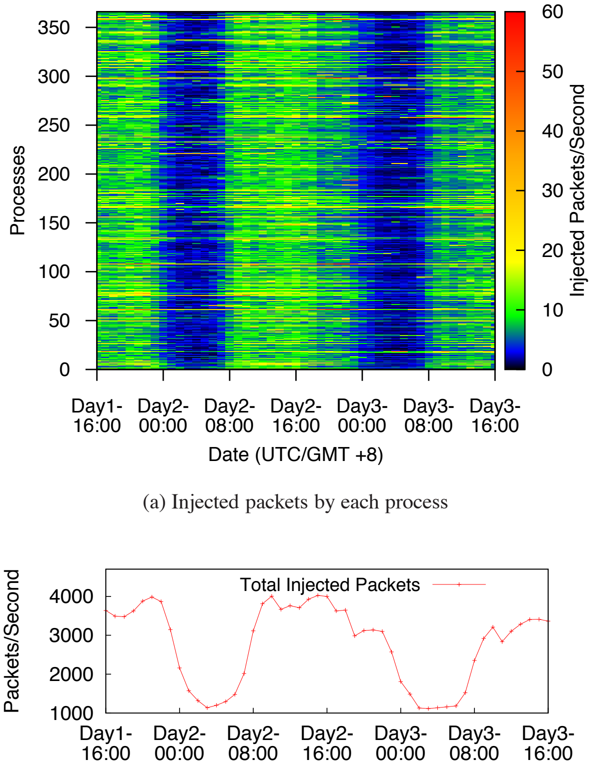
We tested source/destination IP, IP TTL, IP ID, source port, DNS ID, and query name, to assess whether the load balancer selected these fields by varying just a single field and conducting multiple tests. Only the source and destination IP appear to affect the load-balancing.

We then probed the GFW node using different destinations and limited TTLs to determine if, for the same source, two destinations mapped to the same process. Although not enabling us to fully infer the load-balancing algorithm, patterns in the results suggest an implementation based on bit shifting and modulo operations.

Our analysis provided a mechanism to estimate the number of processes behind the load balancer. We probed the censor with 2,048 random destination addresses and observed the resulting mapping of destination addresses to processes. We repeated this experiment several times to minimize grouping errors, after which we discerned 367 separate monitoring IP ID/TTL patterns. Although our methodology may underestimate if we miss a process, it does appear to provide a reasonable estimate that this GFW node employs hundreds of distinct processes.

Mapping out the processes then enables a "packet-pair" type measurement of censorship volume: by sending two packets to the same process on the same GFW node in a one-second interval, the resulting difference in IP TTL should reflect the number of censorship events occurring in that interval. (Here we assume that few processes would inject more than 200 pps, the dynamic range that we can observe in the TTL field).

Based on 240 samples per hour from each of the 367 processes, collected over two days, we computed the average volume of censored traffic for each hour. Figure 6a shows the number of the injected packets for each of the 367 processes, and Figure 6b shows the total. These findings suggest that although a few processes exhibit hot spots (yellow rows in Figure 6a), the load balancer generally works well. We estimate the total number of censorship events for the node as summing on average to around 2,800 pps, with a range from 1,100 to 4,000 pps over the course of a day.



(a) Injected packets by each process

(b) Injected packets by the censor node

Figure 6: Estimated volume of censored traffic.

8 Summary

We have attempted a detailed characterization of how the Great Firewall of China censors sites by injecting bogus DNS replies, based on a series of extensive measurements from both external and internal vantage points.

We find that for the more than 1.2 million /24 subnets within China, the Great Firewall's monitors view at most about 7,000 (and likely just a few hundred) associated network links, with all of these links belonging to border ASes. Thus, contrary to previous studies, it appears that China deploys GFW DNS nodes only at the country's edge; for domestic traffic, it affects only the small portion that happens to transit through border ASes while remaining within the country.

Based on testing of 4 TLDs and the Alexa 1M sites, we discovered 35,332 censored domains. From these we further extracted 14,495 keywords matched in 7 different patterns, 2 of which predominate. We argue that this list covers the majority of the GFW's block list. Experiments with two distinct GFW locations obtained the same blocklist, suggesting central management.

From probing about 150K open DNS resolvers/forwarders inside China over a 2-week period, we found that more than 99.85% provided polluted answers. For the small number of "clean" resolvers, most forward their DNS queries to external public resolvers such as Google's, presumably through an uncensored tunnel. A few instead drop any responses that contain a known GFW "Bad IP", and one appears to operate in an uncensored Internet exchange point.

Finally, by targeting one GFW node, we confirmed its use of source and destination addresses for load-balancing DNS analysis. Leveraging patterns in the IP ID and IP TTL fields of injected packets, we estimate that the single node employs about 360 distinct processes that collectively inject censored responses at an average rate of roughly 2,800 per second.

References

[1] GFW Technology Review. `http://gfwrev.blogspot.com/`, 2009.

[2] Scholar Zhang: Intrusion Detection Evasion and Black Box Mechanism Research of the Great Firewall of China. `https://code.google.com/p/scholarzhang/`, 2010.

[3] Online Censorship In China — GreatFire.org. `https://en.greatfire.org/`, 2014.

[4] ANONYMOUS. The Collateral Damage of Internet Censorship by DNS Injection. *ACM SIGCOMM Computer Communication Review 42*, 3 (2012), 21–27.

[5] BROWN, M. A., MADORY, D., POPESCU, A., AND ZMIJEWSKI, E. DNS Tampering and Root Servers. `http://www.renesys.com/tech/presentations/pdf/DNS-Tampering-and-Root-Servers.pdf`, 2010.

[6] CRANDALL, J., ZINN, D., BYRD, M., BARR, E., AND EAST, R. Conceptdoppler: A Weather Tracker for Internet Censorship. *14th ACM Conference on Computer and Communications Security (CCS)* (2007), 1–4.

[7] DONG, B. Forbidden Sites Hijacked All over China. `http://www.dit-inc.us/report/hj.htm`, 2002.

[8] GUMMADI, K. P., SAROIU, S., AND GRIBBLE, S. D. King: Estimating latency between arbitrary internet end hosts. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment (IMW)* (2002), ACM, pp. 5–18.

[9] KHATTAK, S., JAVED, M., AND ANDERSON, P. D. Towards Illuminating a Censorship Monitor's Model to Facilitate Evasion. *3rd USENIX Workshop on Free and Open Communications on the Internet(FOCI)* (2013).

[10] LEYDEN, J. DNS Poisoning Slams Web Traffic from Millions in China into the Wrong Hole. `http://www.theregister.co.uk/2014/01/21/china_dns_poisoning_attack/`, 2014.

[11] LOWE, G., WINTERS, P., AND MARCUS, M. L. The Great DNS Wall of China. `http://cs.nyu.edu/~pcw216/work/nds/final.pdf`, 2007.

[12] MOZUR, P. China Websites Hit With Disruptions. `http://blogs.wsj.com/digits/2014/01/21/chinas-sina-baidu-and-other-big-websites-are-hit-with-disruptions`, 2014.

[13] 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 *IEEE 30th International Conference on Distributed Computing Systems (ICDCS)* (2010), pp. 315–326.

[14] WEAVER, N., SOMMER, R., AND PAXSON, V. Detecting Forged TCP Reset Packets. In *Proceedings of the Network And Distributed Security Symposium (NDSS)* (2009).

[15] XU, X., MAO, Z., AND HALDERMAN, J. Internet Censorship in China: Where does the Filtering Occur? *Passive and Active Measurement* (2011), 133–142.