This paper will appear at the 30th International Conference on Distributed Computing Systems (ICDCS 2010), Genoa, Italy, June 21–25, 2010

Empirical Study of a National-Scale Distributed Intrusion Detection System: Backbone-Level Filtering of HTML Responses in China

Jong Chun Park Jedidiah R. Crandall

Univ. of New Mexico, Dept. of Computer Science
Mail stop: MSC01 1130, 1 University of New Mexico, Albuquerque, NM 87131-0001
{joumon, crandall}@cs.unm.edu

***Abstract*—We present results from measurements of the filtering of HTTP HTML responses in China, which is based on string matching and TCP reset injection by backbone-level routers. This system, intended mainly for Internet censorship, is a national-scale filter based on intrusion detection system (IDS) technologies. Our results indicate that the Chinese censors discontinued this HTML response filtering for the majority of routes some time between August 2008 and January 2009 (other forms of censorship, including backbone-level GET request filtering, are still in place). In this paper, we give evidence to show that the distributed nature of this filtering system and the problems inherent to distributed filtering are likely among the reasons it was discontinued, in addition to potential traffic load problems. When the censor successfully detected a keyword in our measurements and attempted to reset the connection, their attempt to reset the connection was successful less than 51% of the time, due to late or out-of-sequence resets.**

In addition to shedding light on why HTML response filtering may have been discontinued by the censors, we document potential sources of uncertainty, which are due to routing and protocol dynamics, that could affect measurements of any form of censorship in any country. Between a single client IP address in China and several contiguous server IP addresses outside China, measurement results can be radically different. This is probably due to either traffic engineering or one node from a bank of IDS systems being chosen based on source IP address. Our data provides a unique opportunity to study a national-scale, distributed filtering system.

I. Introduction

"A system is distributed if the message transmission delay is not negligible compared to the time between events in a single process [1]." For the typical network intrusion detection systems that have been studied in the literature thus far, one endhost is trusted and the IDS system is placed close to that endhost. While problems of keeping the state consistent still present themselves in this context [2], [3], intrusion detection on this scale and at the scale of large organizations has been well studied. What if both endhosts are untrusted and the filter is geographically separated by a long distance from both of them? Even if neither endhost tries to evade the filter, having a consistent view of, *e.g.*, the sequence and acknowledgment numbers of a TCP connection can be very challenging. In this paper, we present empirical results to suggest that China's backbone-level filtering of HTTP HTML responses, which is the only filtering system of this scale to date and appears to have been discontinued, did indeed have exactly this problem.

Using open web proxy servers, we tested the packet- and application-level dynamics of HTTP keyword filtering, specifically for HTML responses, for 47 locations in geographically and topologically diverse parts of the Chinese Internet. In total, our data consists of 123GB of raw tcpdumps and 613GB in an annotated SQL database. These 47 locations are from three datasets, 12 locations were measured in August 2008 immediately after the end of the 2008 Olympics in Beijing (which would have interfered with our measurements prior to the end of the Olympics), 20 more locations in January 2009, and 15 more in August 2009. The 2008 data was based on a preliminary measurement methodology so only conclusions about the aggregate effects can be drawn from this earlier data. The January 2009 data is based on an improved measurement methodology that enables us to support conclusions about the causes of uncertainties, but does not contain any late RST packets after a connection is closed due to a local stateful firewall[1]. The August 2009 data is also based on the improved measurement methodology and contains no stateful firewall effects because we added a rule to not keep state for or interfere with measurement IP addresses.

Our data provides a unique opportunity to empirically study how a national-scale, distributed, deep-packet filtering system performs. Past studies of censorship in the backbone of China's Internet have focused on GET request filtering, which is very effective and still present today. GET requests happen early in the TCP connection and usually consist of a single packet that is piggybacked on the third part of the three-part TCP handshake. Before TCP congestion control has ramped up and multiple packets are en route, there is not an opportunity for the state (*e.g.*, sequence and acknowledgment numbers of the server, client, and offending packet) to be inconsistent. Thus, injecting TCP reset (RST) packets using information from the

[1]We determined that late RSTs, which came after an ACK or successful RST, are the only packets that this firewall could have dropped or interfered with in any way, because it was an OpenBSD firewall that we determined takes the strictest possible interpretation of RST packet sequence numbers [4]. So the firewall would never consider a connection to be closed unless the endhost also considered the connection to be closed.

header of an offending packet (*i.e.*, a packet that contains a banned keyword) is much more likely to be effective for GET requests than for HTML responses. The difference between GET request and HTML response filtering via RST injection goes well beyond just the direction of the traffic, and is really the difference between non-distributed *vs.* distributed filtering.

A. HTTP keyword filtering in China

China performs an array of different forms of Internet censorship, including blacklisting by IP address, blacklisting by domain name in the DNS system [5], keyword censorship on blog and news servers [6], and keyword filtering of queries by search engines. Keyword filtering of HTTP traffic by routers in the backbone of the Chinese Internet was first reported in December 2002 by the Global Internet Freedom Consortium [7], and studied in more detail by Clayton *et al.* [8], [9] and the ConceptDoppler project [10].

When a client attempts to load a web page from a remote server, assuming that the entire web site is not blacklisted by its IP address or DNS domain name, then the client's machine will perform a TCP handshake with the server and then send a GET request, *e.g.*: "**GET http://www.example.com/falun.html HTTP/1.1**". If there is one or more routers on the path from the client to the server that implement keyword censorship, then this packet will be scanned for blacklisted keywords. For example, "falun" is a word that appears on nearly every blacklist, because of its association with the Falun Gong movement. This GET request filtering, which was the subject of all of the past work on HTTP keyword filtering [7], [8], [9], [10], is only one part of this particular censorship implementation.

The HTML response from the server to the client is also sometimes scanned for keywords. The 2005 Open Net Initiative (ONI) report on China [11] stated that HTML responses are not filtered, though there was some small amount of evidence of HTML response filtering reported in the study that the ONI report was based on [12]. The 2009 ONI report mentions that HTML response content is filtered. The media has also reported on HTML response filtering [13], but to the best of our knowledge there is no definitive technical evidence of this filtering in the published literature. Our results demonstrate that HTML response filtering in China exists, though it is rare—and even on routes where it occurs it is relatively scant compared to GET request filtering on the same route. Furthermore, it appears to have been discontinued in many parts of China between August 2008 and January 2009.

In HTML response filtering, when a keyword is detected within an HTTP transfer, the censors attempt to interrupt the connection and stop the transfer by using TCP reset (RST) injection. Typically, RSTs are sent to both the source and destination of the packet. Sometimes multiple RSTs are sent, although the exact details of HTTP keyword filtering vary throughout the country for different implementations. The sequence numbers are chosen to make the RST packets appear to both server and client to be valid. Correctly guessing this sequence number is a key factor in the efficacy of this form of censorship. Because the server and client continue to exchange traffic as the filter is scanning for keywords, and also because multiple packets can be en route at any time and can be reordered, multiple RSTs increase the chances that the sequence number will be interpreted by the client as a valid packet from the server, or vice versa.

TCP protocol dynamics include effects such as flow control, congestion control, retransmission, and packet loss that are highly dependent on the particular routes that packets take. Therefore the application-level dynamics of censorship based on RST injection are largely determined by the route or routes between the server and client (note that routes can be asymmetric and change over time [14]).

B. Contributions of this paper

The data we present in this paper supports several conclusions. Among the main three conclusions listed below, the first two relate to empirical lessons from a national-scale, distributed filtering system and the third relates to sources of uncertainty that should be accounted for in future measurements of Internet censorship.

- We demonstrate that HTTP HTML response filtering was likely discontinued on many routes between August 2008 and January 2009, and that the apparent ineffectiveness of this form of filtering, which results from its distributed nature, was likely a cause for this. In fact, less than 51% of the packet-level attempts of censorship (where the censor detected the keyword and attempted to reset the connection) succeeded in reseting the connection at the application layer. This is due to both late RSTs and RSTs with incorrect sequence numbers.
- We show that diurnal patterns in censor effectiveness [10], which would indicate that traffic load is a significant problem for the censors, are present for HTML responses but not a major factor for most routes. Also, HTML response filtering occurred on many fewer routes than GET request filtering, suggesting that less resources were devoted to it.
- We found that contiguous IP addresses outside the censorship domain, measured from a single client IP address inside the censorship domain (an open web proxy), can give radically different results. There are two plausible explanations for this, both of which must be considered for future measurements of Internet censorship.

C. Implications

While distributed intrusion detection systems have been studied and it is intuitive that TCP protocol state can become a problem at the national scale, we provide the first empirical study to quantify this. Furthermore, studying HTML response filtering adds to the body of knowledge about global Internet censorship and our data supports the view that between competing approaches, namely backbone-level filtering *vs.* local control and non-technical methods of controlling Internet content, the latter appears to be more successful from the censors' point of view based on the empirical evidence thus far. Lastly, from IP address blacklisting to DNS filtering to any form of deep-packet inspection, routing and protocol dynamics

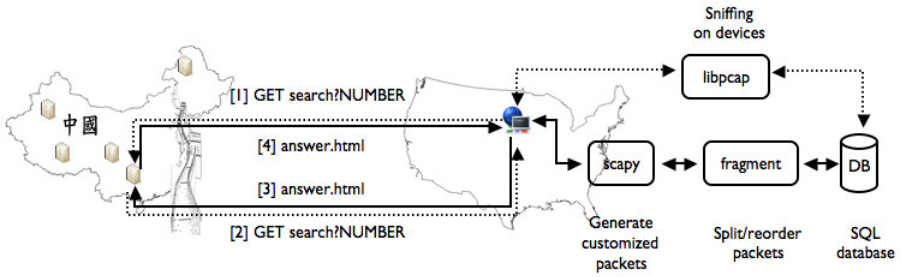


Fig. 1. **Overview of our probing infrastructure.**

such as those we document in this paper can have a dramatic effect on measurement results and should be accounted for in any future Internet censorship measurements. For application-level measurements, such as those performed by the ONI [11], these effects can average out over enough space and time, but packet-level measurement is also important since only through packet-level measurement can we discover in detail exactly what is and is not filtered at different places and times. Application-level measurements give a less fine-grained picture than packet-level measurements.

D. Organization of the rest of the paper

Section II describes the architecture of the probing infrastructure we have built and details the challenges we addressed with our experimental setup. Our results are presented in Section III. This is followed by related work in Section IV, future work in Section V, and the conclusion.

II. Measurement Infrastructure and Experimental Setup

In this section, we describe our probing infrastructure, the challenges our experimental methodology was designed to address, and the experimental setup for the measurements for results presented in Section III.

A. Probing infrastructure

Our probing infrastructure is based on using open web proxies, which can act as both a server and a client within a censorship domain. In China several new open proxies are announced every day in geographically diverse locations and typically have a lifetime of days to weeks. This is because university students must often pay for international Internet traffic while domestic Internet use is free. Thus they can save money by setting up open proxies outside the university but within China's borders. These are public proxies that are announced as free to use by anybody.

Figure 1 shows an overview of the probing infrastructure we have developed. Active probing is performed from outside of a censorship domain such as China (currently we have 28 IP addresses dedicated to this purpose: 14 servers and 14 clients). An individual probe consists of a GET request that is sent to the proxy and then echoed back to one of our web server IP addresses. This GET request can contain a keyword for testing GET request censorship, or it can simply be a randomized identification number of a keyword, for testing HTML responses. That is, when we test HTML responses the GET requests that we use contain only numbers and so will not elicit censorship during the request. The HTML response is then sent to the proxy and echoed back to the client IP address that originated the request. All packets that result from a probe (which can come from the proxy, from one or more censorship devices between the measurement location and the proxy, or from other devices such as IDSes or NATs) are captured using Python's libpcap library. The headers of these packets are split up into the relevant fields and, along with their timestamps, used to populate an SQL database that associates all packets with a particular TCP flow and specific experiment. Some RSTs arrive late and are marked to be later associated with a TCP flow based on the source and destination IP addresses and sequence numbers. All traffic into and out of the probing machine is also stored to a tcpdump, including TCP payloads. The GET requests and HTML responses for probes can be generated either at the application level, using wget or a custom web server, or are customized or manipulated using the Scapy library [15] or our own TCP fragmentation implementation that is similar to fragroute [16]. We implemented TCP fragmentation, rather than IP fragmentation, because small IP fragments are not handled properly by China's tunneling IPv6 backbone, probably for security reasons not having to do with censorship [17], [18].

B. Experimental methodology

The following are the challenges that our experimental methodology was designed to address, and how each was addressed:

Diurnal patterns in both the traffic load of the route and the effectiveness of the censoring router [10] must not be allowed to skew the results of measurement. For example, if small HTML pages were tested first, and then successively larger pages in a test that lasts one day, observed variance of censor effectiveness due to diurnal patterns could be falsely attributed to a dependency on the page size. This is why we varied page size, keyword location, and keyword placement uniformly at random over the 24-hour measurement period for the January 2009 and August 2009 experiments. Similar randomization was performed in the August 2008 experiments, but for four proxy locations in the August 2008 data set (Guangdong, Shanghai #1, Shandong #1, and Shandong #2) larger page sizes took significantly longer to complete leading to the possibility of some interference between diurnal patterns and the varying of the page size for these four data points.

Spurious RST packets from the proxy server itself or other sources such as NATs and load balancers must not be misinterpreted as censorship attempts. RSTs have been observed to be present for as much as 15-20% of all TCP connections

Dataset	Notes	HTTP proxy locations
August 2008	Preliminary methodology; late RSTs possibly dropped by local firewall; possible interference by diurnal patterns in GD, SH1, SD1, and SD2; ZJ and GS only 97% and 84% complete, respecitvely	Beijing (BJ) , Changchun (CC), Gansu (GS), Guangdong (GD), Haiyan (HY), Jilin (JL), Shandong #1 (SD1), Shandong #2 (SD2), Shanghai #1 (SH1), Shanghai #2 (SH2), Wuhan (WH), Zhejiang (ZJ).
January 2009	Improved methodology; late RSTs possibly dropped by local firewall; very little HTML response filtering observed	Beijing #1 (BJ1), Beijing #2 (BJ2), Beijing #3 (BJ3), Beijing #4 (BJ4), Chengdu (CD), Fuzhou (FZ), Guangdong #1 (GD1), Guangdong #2 (GD2), Guangxi (GX), Hebei (HB), Heilongjiang (HLJ), Heilongjiang #2 (HLJ2), Jiangsu (JS), Jilin (JL), Ningxia (NX), Shandong (SD), Shanghai (SH), Shannxi (SX), Tianjin (TJ), Zhejiang (ZJ)
August 2009	Improved methodology; very little HTML response filtering observed	Beijing (BJ), Guangzhou (GZ), Hangzhou (HZ), Harbin (HB), Jinan (JN), Nanjing (NJ), Nanning (NN), Qingdao (QD), Shanghai (SH), Shantou (ST), Shenyang (SY), Suzhou (SZ), Tianjin (TJ), Xiamen (XM), Xian (XA).

TABLE I
HTTP proxy locations.

Algorithm 1 : Pseudocode for August 2008 experiments on a single proxy.

```
K = the 133 keywords we tested with
for IP address pair 0 do
    Perform tests for TCP reordering and reassembly
end for
The five possible page sizes, S, are statically assigned to IP address pairs [1..5]
for all IP address pairs (i.e., page placements s) ∈ [1..5] do
    P = a random permutation of the three keyword placements
    for all 399 probes k, p ∈ K × P do
        for Attempt = 1 to 3 do
            Attempt an HTTP transfer with page size s ∈ S, keyword k ∈ K, and placement p ∈ P
            if Reset is observed at the application level then
                repeat
                    Delay and test non-blacklisted word "hello"
                until Timeout period has expired
            end if
        end for
    end for
end for
for IP address pair 6 do
    Perform GET request keyword testing
end for
```

Algorithm 2 : Pseudocode for January 2009 and August 2009 experiments on a single proxy.

```
for IP address pair 0 do
    Perform tests for TCP reordering and reassembly
end for
for all IP address pairs ∈ [1..12] do
    repeat
        s = a page size chosen uniformly at random from the five possibilities
        k = a keyword, with probability 11/12 of "falunfalun" or 1/12 of "hellohello"
        p = a page placement chosen uniformly at random from the three possibilities
        Attempt an HTTP transfer with page size s, keyword k, and keyword placement p
        Wait 169 seconds
    until 24 hours has passed
end for
for IP address pair 13 do
    Perform GET request keyword testing
end for
```

in normal traffic [19]. We used an injected RST fingerprinting technique that is due to Weaver *et al.* [20] to distinguish RSTs injected for censorship purposes from other RSTs. Particularly in the later data sets, however, censorship RSTs became more difficult to fingerprint due to apparent improvements in the RST injection technique compared to GET request filtering. For this reason we also compare the numbers of RSTs for "falunfalun" (always blocked by the censors) to RSTs for "hellohello" (never blocked). These improvements, which include removing anomalies in the TTL values of forged RSTs, are discussed in Section III-C.

Fragmentation of probes is an issue because, if the censors do not do TCP flow reconstruction, the keyword will not be detected if it is fragmented across multiple packets. In the January 2009 and August 2009 data we used, *e.g.*, "falunfalun" instead of "falun" to negate the effects of fragmentation on measurements not intended to measure fragmentation.

Censorship-based RST packets that arrive late or out of sequence should be properly associated with the connection that elicited the censorship. We saved all packets that arrived at any client or server into the SQL database, and then refactored the database to associate each RST packet with the correct

connection based on the source and destination IP addresses and sequence numbers.

Proxy caching and proxy failures should not be reflected in the data. For each experiment, the GET request is performed using a unique random identifier so that the proxy cannot cache the HTML responses and respond directly without contacting one of our servers. HTML error pages from the proxy, such as **403** (forbidden), **502** (bad gateway), **503** (service unavailable), and **504** (bad gateway) errors must be accounted for. For all but one proxy (Zhejiang from August 2008), the number of these errors was negligible (less than 1%). For all three sets of experiments, we account for cases where the proxy was simply unable to serve the page due to high load or other reasons by exluding all data points where the proxy did not contact the server for reasons other than reset connections due to censorship.

Traffic engineering, where traffic can be routed differently based on its source and destination IP address, can affect the results of measurements as we will demonstrate in Section III. For the January 2009 and August 2009 experiments we varied all parameters (page size, keyword, and keyword placement) uniformly at random (except for keywords, where "falunfalun" was preferred to "hellohello" by a ratio of 11 to 1) for all IP address pairs, so that variance across IP addresses indicates traffic engineering while variance across other parameters indicates variance based on those parameters. The August 2008 experiments measured the five page sizes on five different IP addresses, so that it is impossible to determine if variance along this axis is due to page size or due to traffic engineering in the earlier data set.

Timeout periods are enforced by all of the different HTTP keyword filtering implementations that we have found in China. This means that after an offending keyword is detected and a connection reset attempt is made, the censors will attempt to reset all connections between the same two IP addresses for some time into the future [8], [9] (on the order of one or two minutes), regardless of whether packets contain blacklisted keywords or not. The length of this timeout period can be based on either a fixed length buffer or an actual timer. For the January 2009 and August 2009 experiments we always waited 169 seconds between each connection (this is greater than two minutes but still ensures that 12 client-server pairs can each perform 512 experiments within 24 hours). The August 2008 experiments wait until "hello" is not blocked to continue, rather than being evenly spaced.

The last challenge complicates measurement in the August 2008 data set because it can lead to dependence between subsequent measurements on the same pair of IP addresses. As our results in Section III show, HTML response keyword censorship can be unreliable. Thus, we may be able to load a web page seemingly unfettered at the application-level when the censor actually did detect the keyword and then sent RSTs that arrived late or had the wrong sequence numbers and placed those IP addresses in the timeout period. Thus the SYN packet for the next test will be automatically reset before any keywords are actually transferred. One solution to this problem would be to enforce a delay between tests whenever a censorship RST is present in a test. The problem with this is that we would have to fingerprint all types of RSTs *a priori*. Another solution, which we adopted for the January 2009 and August 2009 experiments, would be to always wait for over two minutes between tests on the same IP addresses.

In the August 2008 experiments our solution was to take a two-pronged approach to reducing the effects of dependence due to timeout periods: minimizing their effects at the application level through redundancy, and ordering individual probes in such a way as to negate any possibility of these effects skewing the results. For each individual probe, which is composed of a $\{\text{Keyword}, \text{Page size}, \text{Keyword placement}\}$ tuple, the HTTP transfer is attempted three times. For a blacklisted keyword, the second and third attempts are very likely to be reset at the application level if the first attempt is reset at the packet-level, even if that combination of page size and keyword placement is one that that particular censor on that route is very unlikely to reset at the application level. This is because during the timeout period RSTs are sent in response to the initial SYN packet, not the packet with a keyword in it, and thus are much more reliable at the application level. Furthermore, each HTTP transfer has four steps: the GET request from client to proxy and from proxy to server, and the HTML response from server to proxy and from proxy to client. Thus, if a keyword is blacklisted, it is very unlikely that the probe will observe three **200 OK** responses from the proxy in a row at the application level. Also, in the August 2008 data set, all of the probes for a given page size are performed on a unique client-server pair of IP addresses. Therefore it is impossible for probes for different page sizes to interfere with one another. We also randomly permute the order in which the 3 placements are tested, so that any possible interference between different placements is minimized. The January 2009 and August 2009 data sets do not have these same concerns, we present these concerns here since much of the censorship we measured is in the August 2008 data set so interpreting this dataset, despite the flaws in the August 2008 measurement methodology, is important.

Prior to July 2009 our network had a stateful firewall that may have dropped late RSTs that arrived after connections had been closed on our end. We repeated our experiments in August 2009 for two reasons, to see what effect this stateful firewall may have had on the overall results from January 2009 and also to see if the apparent reduction in HTML response filtering we noticed was temporary or more long-term. While we were not able to take measurements during the 2008 Beijing Olympics due to a high load on all proxy servers in the country (Olympic coverage video could be viewed online for free from IP addresses within China, so there was a large increase in public proxy usage), we did notice a large reduction in the amount of censorship, as was reported in the media. China was undergoing a U.N. human rights review during early 2009, so to determine if the January 2009 reduction in HTML response filtering was a similar temporary reduction, we repeated the experiments in August 2009. The reduction in HTML response filtering is also apparent in August 2009, while GET request filtering appears to be pervasive in August 2009, suggesting that this reduction in censorship is likely specific to HTML response filtering. We also changed our

measurement IP addresses between January 2009 and August 2009 to make sure that the apparent reduction in HTML response filtering was not a result of our measurement IP addresses being whitelisted in some parts of the country.

C. Experimental setup

We searched for open proxies on publicly available lists for testing, selecting proxy locations that gave us a good geographical and routing-topological cross-section of the Chinese Internet. The proxy locations that we tested are shown in Table I. For two proxy locations in the August 2008 data set, Changchun and Jilin, we were forced to discard all HTML response data because of apparent IP address blacklisting. Two more proxy locations in the August 2008 data set, Zhejiang and Gansu, did not complete the full set of probes but are included because they completed 97% and 84%, respectively. All other proxy locations that were included in all data sets completed the full set of probes with no problems. While it appears that our measurement IP addresses may be blacklisted in the extreme north of the country, proxy locations in all other parts of the country are unaffected.

The sequence of tests that we performed for each proxy is shown in Algorithm 1 for August 2008 and Algorithm 2 for January 2009 and August 2009. The source of all probes was in North America. For any HTTP transfer, there are three variables in our measurements: the keyword, the page size, and the keyword placement. We used 133 keywords obtained from GET request testing in one part of China for the August 2008 experiments, and "falunfalun", which is consistently blocked in all parts of China at all times, for the January 2009 and August 2009 experiments. These 133 keywords were all determined to be blocked in GET requests using a probing technique similar to ConceptDoppler [10].

We also tested five different page sizes. A minimal HTTP HTML response with all of the framing is about 750 bytes. We padded this with innocuous characters (that do not elicit censorship) in increments of 0, 1, 10, 50, and 100 for page sizes of approximately 750 bytes, 4 KB, 34 KB, 175 KB, and 340 KB. The final variable is the placement of the keyword, which can be placed in the beginning, middle, or end of the page. For each proxy, we executed a series of tests to determine if TCP reassembly or reordering was performed by the censors on the routes to/from that proxy. One server-client IP address pair was dedicated to this. One more IP address pair was dedicated to testing GET request filtering.

D. Identifying censorship

For distinguishing between censorship RST packets and other RSTs, we used a RST fingerprinting technique that is due to Weaver *et al.* [20] in combination with other heuristics such as comparing "hellohello" to "falunfalun" block rates and looking for anomalous TTL ranges. To factor out causes of application-level failures to download a full web page that were not due to censorship, we excluded proxy errors and timeouts from the data and consider only connections reported as reset at the application layer when a GET request was received by our server (so that resets from the proxy are not considered to be application-level censorship).

An assumption of our measurement methodology is that RSTs are sent in both directions, *i.e.*, to both the client and server, when censorship is attempted at the packet level. The only two proxies where this assumption is not confirmed by the packet-level data are Zhejiang from August 2008, where a significant number of proxy errors were observed, and Wuhan from August 2008, which contained some apparent application-level censorship for small HTML transfers that was not matched at the packet level by fingerprinted RSTs. For all other proxies we found RSTs at the packet level when censorship was apparent at the application level.

III. Results and Analysis

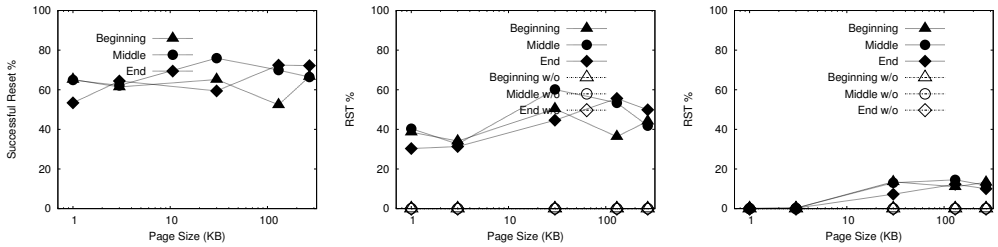
In this section, we support the conclusions enumerated in Section I.

A. HTML response filtering effectiveness

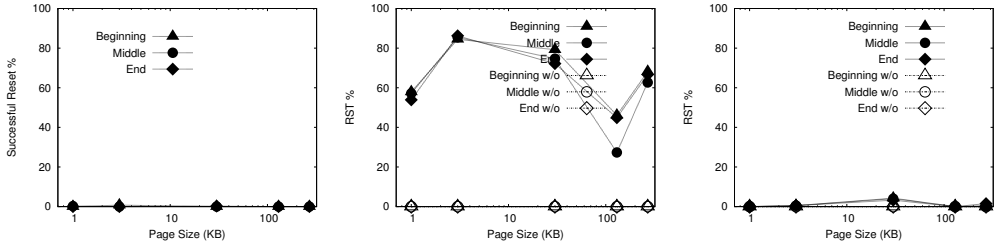
We observed that HTTP HTML response filtering by backbone-level routers in China was discontinued on most routes sometime between August 2008 and January 2009. While 6 of 10 proxy locations in August 2008 showed significant evidence of HTML response filtering, only 1 of 20 showed significant evidence of HTML response filtering in January 2009, and 1 out of 15 in August 2009. Other forms of censorship, including GET request filtering at the backbone level, are still prevalent. What was the reason for HTML response filtering to be discontinued? Here we give evidence to support the hypothesis that the distributed nature of the TCP/IP protocol in the middle of a connection (as opposed to the GET request which is usually piggybacked on the three-way handshake), in particular inconsistencies in the state of the acknowledgment and sequence number, was likely a major factor.

Figure 2 shows the dynamics of filtering in the August 2008 data in 5 cities that showed a significant amount of resets. The three graphs for each proxy from left to right, respectively, are the application-level effectiveness of resets, the percentage of connections between client and proxy that had resets in them, and the percentage of connections between proxy and server with resets in them. The y-axis is the successful reset rate (%) for application level, and the percentage of connections with resets for the two graphs at the packet level. Triangles, circles, and diamonds correspond to the placement of the keyword within the page. Black *vs.* white for the packet-level graphs is the number of resets that come before the connection is closed *vs.* the number of stale, *i.e.*, late RSTs, respectively. The x-axis is the size of the web page. Note that some late RSTs were dropped by a local stateful firewall, which may explain why Wuhan has a higher application-level censorship effectiveness than the sum of the packet-level attempts at censorship. It is likely that the proxy received resets for small HTML transfers and closed the connection before the resets on our end arrived.

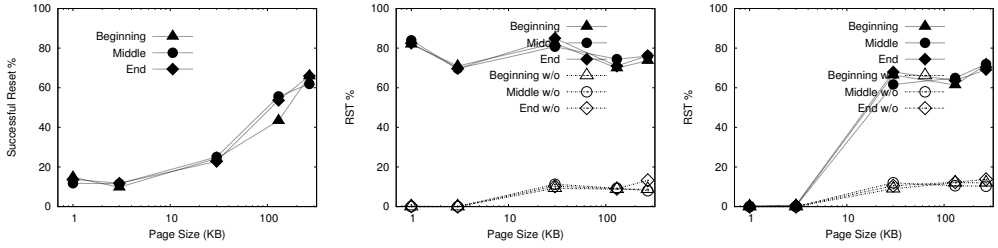
Note that, in Figure 2, when the censor successfully detects a banned keyword at the packet level and attempts censorship by injecting a RST packet, this does not always translate into



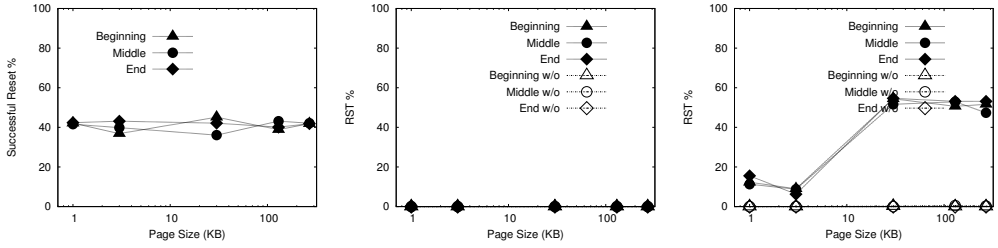
(a) Gansu, August 2008



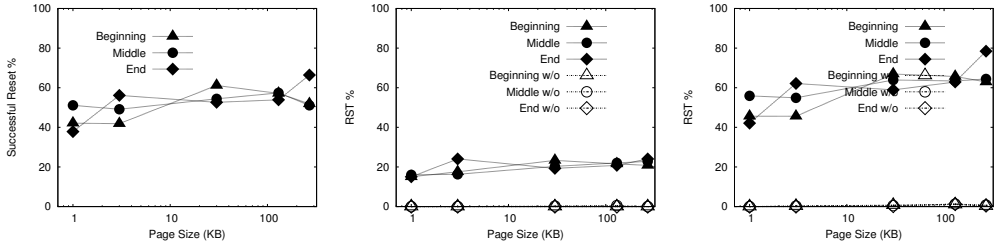
(b) Guangdong, August 2008



(c) Shandong #1, August 2008



(d) Wuhan, August 2008



(e) Zhejiang, August 2008

Fig. 2. **Application-level, client-proxy packet-level, and server-proxy packet-level censor effectiveness for proxy locations from the August 2008 data that showed significant evidence of HTML response filtering.**

a failure to transfer the web page due to an application-level reset of the connection. In fact, our measurements show that the censor is less than 51% effective at turning detected keywords into reset connections at the application layer, and for, *e.g.*, Guangdong, the HTML response filtering is completely ineffective even though the keywords are detected most of the time between the client and the proxy. Our measured rate of application-layer resets given that the censor attempted to reset the connection at the packet-level can only be an overestimate of censor effectiveness. The late RSTs that were dropped

in our August 2008 data could only make this measured rate lower, and the fact that we measure application-layer effectiveness as an aggregate of the client-proxy and proxy-server connections can also only lead to overestimation (not underestimation), since the censor has two attempts per HTML transfer to successfully reset connections at the application level.

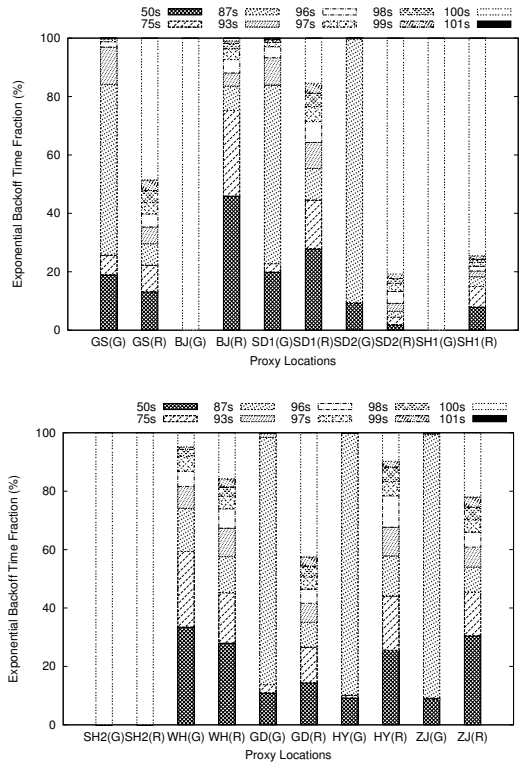


Fig. 3. **Backoff timeouts, August 2008 (G = GET requests, R = HTML responses).**

To confirm that the RSTs in the August 2008 data were due to censorship, we examined the timings of backoffs that occurred after connections were reset. Figure 3 shows the distribution of for how long RST packets were sent after an application-level reset. We tested routes that received application-level resets with "hello" and an exponential backoff. Certain times for certain proxy locations are strongly represented, which is behavior that would be expected of keyword censorship, not an overloaded proxy that is dropping connections.

We did witness apparent HTML response filtering for a proxy in Beijing in January 2009, but were not able to independently confirm that the resets were censorship related. For a proxy in Beijing in August 2009, we witnessed resets at the packet level between the proxy and our server that were definitely censorship-related since 100% of these RSTs came for "falunfalun" and 0% for "hellohello." As shown in Figure 4, these resets were completely ineffective at the application level. Figure 4(b) shows, from left to right, the reset success rate at the application layer, and the percentage of connections with a reset between the client and proxy and the proxy and server, respectively. Figure 4(a) shows candlestick graphs between the proxy and server for the beginning, middle, and end placements of the keyword within the webpage, respectively, where the candlestick shows the mean, one standard deviation in both directions, and the minimum and maximum values. These candlestick graphs are for the 12 client-server pairs that were all running probes simultaneously, so the variance depends only on IP address.

For the January 2009 and August 2009 data, there is an independent way to confirm the presence of forged RSTs that are due to censorship: the ratios for "falunfalun" *vs.* "hellohello" resets. For 12 proxy locations, there was a significant difference in the ratio between resets for "falunfalun" and for "hellohello" that suggests censorship, with nine proxy locations receiving 100% of resets for "falunfalun" and zero for "hellohello". There were several proxy locations that had a significant bias towards "hellohello," with up to 85% of RSTs occuring for "hellohello." However, these were proxies where very little censorship was observed at either the packet or application level.

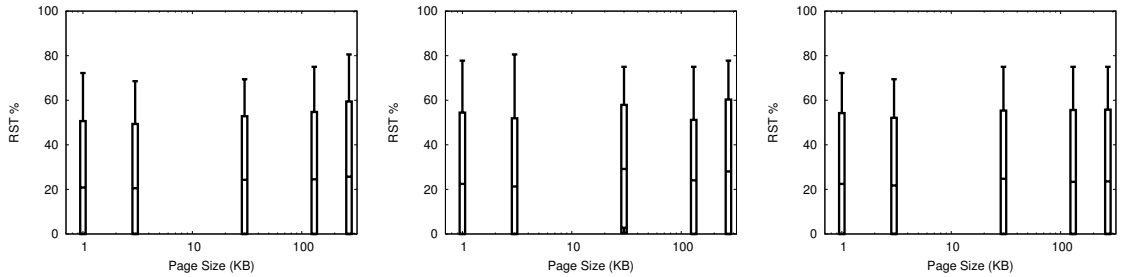
B. Diurnal patterns

Another hypothesis is that the filters which had been deployed for HTML response filtering in China's backbone could not handle the amount of traffic that needs to be scanned for keywords. There are two ways that this can manifest: routes that have no filtering router and diurnal patterns that indicate that the filter on a particular route is overloaded during busy Internet periods. The ConceptDoppler project [10] reported both of these phenomena for GET request filtering. Our data shows many routes that have no HTML response filtering. While Figure 5 shows some evidence of diurnal patterns, particularly in Zhejiang, these diurnal patterns are not significant enough for filter load to be the only reason that HTML response filtering was discontinued.

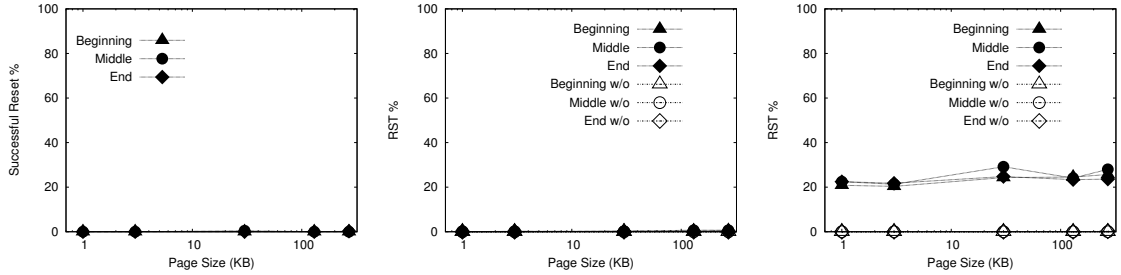
C. Uncertainties in censor effectiveness

For packet-level measurements of Internet censorship that can determine with high confidence whether a given keyword or host identifier is censored in a given place at a given time, it is important to understand sources of uncertainty in censor effectiveness. In our August 2008 data, censor effectiveness varied along the x-axis in Figure 2, which would indicate that it can depend on either web page size, IP address, or both. Our 2009 experiments were designed to determine which of these caused the variance, and with the HTML response filtering that was present in 2009 we were able to determine that contiguous measurement IP addresses can see dramatically different profiles of censor effectiveness. Figure 6 shows the server-proxy packet-level censorship effectiveness for Beijing in August 2009 but for 12 different measurement server IP addresses (all hosted on a single machine in north America).

There are two possible explanations for this. One is that traffic engineering is causing the packets to take different routes even though they are being routed between the same two subnets. This is very common. Another plausible explanation may be that multiple IDS systems are connected to a single filtering router and each is responsible for a subset of source IP addresses, with some IDS systems being more heavily loaded



(a) Beijing August 2009, proxy-server packet-level candlestick graphs for keyword placements in the beginning, middle, and end of a web page



(b) Beijing August 2009, censorship effectiveness at the application level, packet-level for client-proxy, and packet-level for proxy-server

Fig. 4. **Overview of Beijing August 2009 censor effectiveness.**

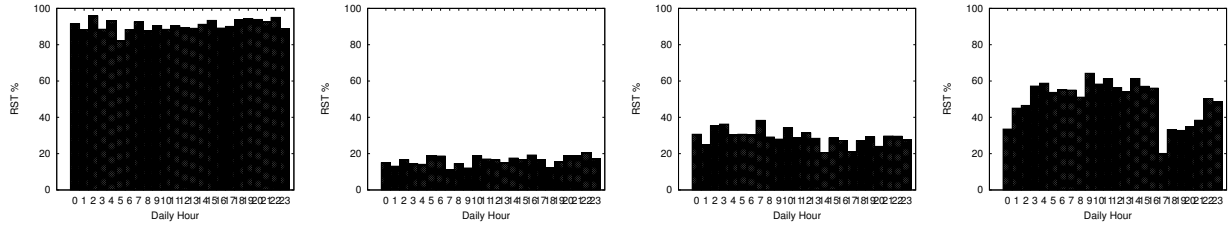


Fig. 5. **Application-level diurnal patterns for August 2008 proxies with significant application-level censorship.** 0 **on the** x**-axis corresponds to midnight in Beijing, the** y**-axis is the percentage of successful application-level resets during that hour out of all HTTP attempts. From left to right, these graphs are for Gansu, Shandong #1, Wuhan, and Zhejiang.**

than others. Clayton *et al.* [8] discovered that RST responses sent for offending packets that had been sent in sequence came back out of order, suggesting that the different packets were processed by different devices on the same hop of the same route.

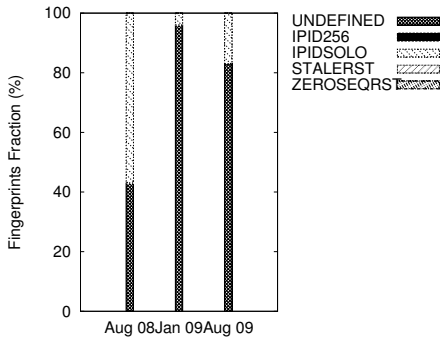


Fig. 7. **Overall RST fingerprint profiles.**

Another interesting result of our measurements was that detecting the RST packets was more challenging than for past studies on GET requests [8], [9], [10]. This suggests that HTML response filtering is a separate mechanism and that the censors have updated their methods for hiding the injected RSTs. For GET request studies, the TTL of injected RST packets was often less than that of normal packets for a given connection, since the injected RSTs had fewer hops between the censor and the endhosts. It is simple for the censors to use the TTL of the offending packet, however, and hide this effect. While some HTML response filtering occurred with obviously anomalous RST TTLs, Figure 8 shows that many filters use injected RSTs that are indistinguishable from normal packets in the connection if only the TTL is considered. In addition to TTLs, we also used RST fingerprinting that is due to Weaver *et al.* [20] (the overall results of which are shown in Figure 7) and compared rates of RSTs for "falunfalun" to "hellohello" to determine which RSTs were injected censorship RSTs. The fingerprints for Figure 7 are described by Weaver *et al.* [20].

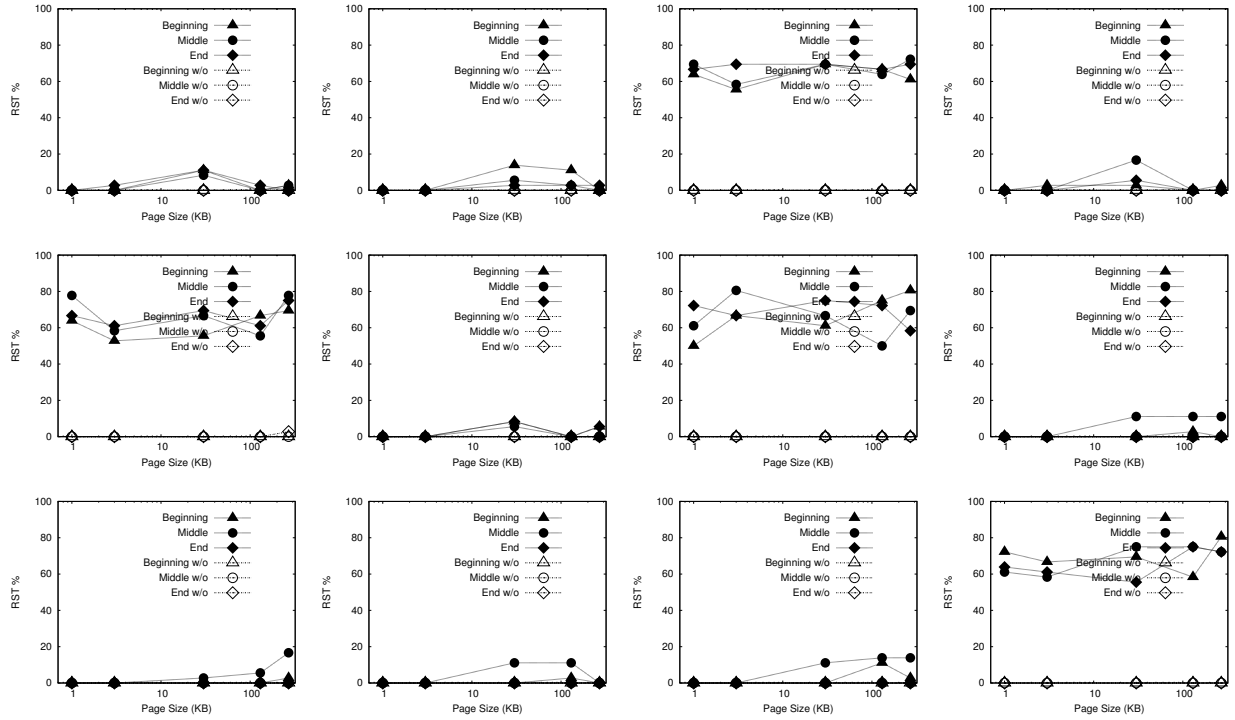


Fig. 6. **Beijing August 2009 packet-level censor effectiveness at detecting keywords, between server and proxy, for 12 different, but contiguous, server IP addresses.**

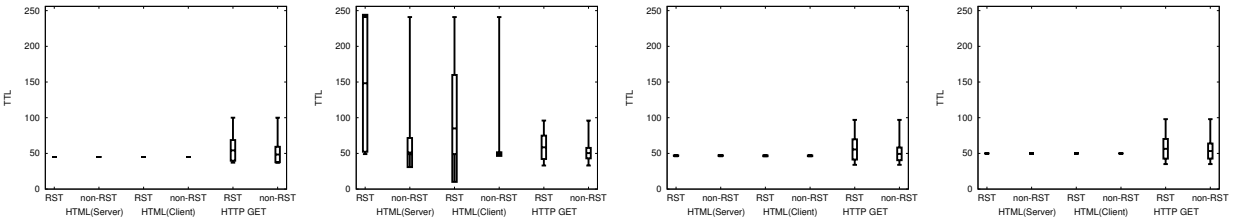


Fig. 8. **TTL ranges for proxy locations with evidence of censorship that is independent of TTL. From left to right, these graphs are for Beijing August 2009, Beijing #2 January 2009, Chengdu January 2009, and Hebei January 2009. These candlestick graphs show the mean, one standard deviation in both directions, and minimum and maximum values for observed TTLs, which roughly corresponds to the number of hops away from which the packet came.**

D. Filter state

In our experiments we also tested the statefulness[2] of both HTML response and GET request filtering. Thus far, by distributed state we have meant the acknowledgment and sequence numbers as seen by both endhosts *vs.* the acknowledgment and sequence numbers that the censor is able to infer from the offending packet's header. Another form of statefulness would be if TCP flows were reconstructed and possibly also reordered to detect keywords that were split between packets. Also, for GET requests, there is the question of whether an offending GET request only elicits censorship when it is part of a valid open TCP connection, or if any packet with the ACK bit set and an offending GET request elicits censorship.

[2]Here, we mean "stateful" in the general sense of any filtering that is based on stored state from past packets, not specifically in the sense that this term is often used in IDS research.

We found that none of the routes that performed HTML response filtering showed definitive evidence of TCP reassembly or reordering, *i.e.*, only probes where the keyword was contained in a single packet elicited censorship attempts (thus a keyword split across multiple packets will not be detected at the packet level). All of the failures to load the HTML page in the August 2008 reassembly and reordering experiments were attributed to issues having to do with the web server and using the HTTP 1.0 protocol (wget does not support HTTP 1.1), and not to censorship. Most of the January 2009 and August 2009 experiments showed very little evidence of HTML response filtering between the proxy and server, and we are not able to perform the reassembly and reordering experiments between the client and proxy because we do not control the proxy at the packet level. Thus we cannot say definitively at this time whether HTML response filtering performed reassembly and reordering of TCP packet flows, but for the proxy locations where we did witness censorship between the server and proxy we found no evidence of problems transferring split and

reordered keywords (at the TCP/IP packet level) that could not be attributed to causes other than censorship.

We ran a separate set of tests for GET request statefullness in August 2008, to resolve conflicting reports from Clayton *et al.* [8], [9] and ConceptDoppler [10]. The former had found GET request filtering to be stateless, and the latter found it to be stateful, in terms of whether an offending GET request that is not part of an open, valid TCP connection will elicit censorship or not. We tested this for 15 web servers in China, mostly near Beijing but also in Tianjin, Fujian, and Hebei. We tried GET requests with blocked keywords, both with and without valid TCP handshakes, and filtered out TCP RSTs from the server itself by fingerprinting. Two web servers, one in Hebei and one in Beijing, had censors on our routes to them that sent RSTs even for packets when no TCP handshake was performed. Therefore these two censors were stateless, but eight others were determined to be stateful, *i.e.*, RSTs from the censor are sent for offending packets when a TCP handshake is performed first, but not for offending packets that are not part of an active TCP connection. For five servers we could not determine stateless *vs.* stateful. These results show that GET request filtering is stateful in some parts of China, but not others.

IV. Related Work

The Open Net Initiative is an excellent source of information about censorship in a variety of countries [21]. Their application-level measurements give a very valuable overall picture of global Internet censorship, but to determine exactly what is filtered and how at different places and times packet-level measurements are more exact. For example, the report by Zittrain and Edelman [12] on which the 2005 ONI report on China is largely based acknowledges the different mechanisms of censorship (IP address, DNS address, keyword blacklisting) but does not distinguish between these different forms of censorship in regards to the availability of different web pages. This makes it difficult to discern over-blocking from what is actually targetted by the censor, because only clusters of filtered concepts can be discerned.

The methods of China's HTTP keyword filtering were first published by the Global Internet Freedom Consortium [7]. Clayton *et al.* [8], [9] published a more detailed study of this mechanism, focusing on a single route (the route may have changed during their study, or the implementation of filtering on the route changed). They provided valuable empirical data on how RST injection worked at the packet level and gave detailed insights on how this was implemented, including their discovery that there is possibly a bank of many IDS systems per router that do the actual string matching. The ConceptDoppler project [10] studied multiple routes, and found that HTTP keyword filtering in China is not peremptory and is not strictly implemented at the border of the Chinese Internet, with a significant amount of filtering occurring in the backbone. ConceptDoppler also showed the presence of diurnal patterns in censor effectiveness for GET requests.

Villenevue [22] presents evidence of both keyword filtering and surveillance in TOM-Skype. Clayton [23] and Dornseif [24] both give details of implementations of censorship in the United Kingdom and Germany, respectively (the former only proposed). Wolfgarten [5] presents details and measurements of DNS tampering in China. Danezis and Anderson [25] explore the economics of resisting censorship. Aycock and Maurushat [26] explore the possibility of releasing a "good" worm to test Internet censorship. Tygar [27] gives a survey of privacy issues and research with a focus on Asia.

In addition to measuring how censorship works, it is important to measure what is censored. Human Rights Watch [28], Reporters Without Borders [6], and others [29], [30] have released reports describing China's censorship regime. These reports often include insider information about what is censored [6] or perhaps full leaked blacklists, such as the list of keywords blocked in the QQChat chat program [28, Appendix I] or by a particular blog site [28, Appendix II]. These accounts are rare and only give a snapshot of that particular censorship implementation at one single point in time.

Tor [31], Psiphon [32], and other evasion technologies (whether they were originally intended for evasion or not) are available for censorship evasion. Sovran *et al.* [33] propose a method for disseminating proxy addresses in such a way that the censors cannot learn about them all to shut them down. Feamster *et al.* [34] propose Infranet, where traffic is modulated over seemingly innocuous standard HTTP connections in a covert manner. Fiat and Saia [35] present a peer-to-peer content addressable network that is resistant to censorship. We do not advocate evasion technologies alone as an effective strategy to address global Internet censorship, but they are an important component of addressing this issue. Most existing evasion technologies today still cause a decline in bandwidth and usability, however, and thus partially fulfill the goals of the censor. Furthermore, evasion of censorship is illegal in many countries, and no existing evasion technology can evade censorship without at least temporary changes to the client system.

V. Future Work

We have shown that the effectiveness of HTML response censorship based on RST injection can depend dramatically on the routing and protocol dynamics of the particular route. These dynamics can affect any type of filtering, so other forms of censorship around the world should be studied to understand their interactions with routing and protocol dynamics so that Internet censorship measurements can be based on a sound understanding of the underlying technologies. Also, different operating system network stacks and web clients exhibit different behaviors for RST packets, depending on how they are interpreted [4], thus further study using a variety of OSes and clients (our study used only Linux 2.6 and wget) within the context of RST injection is warranted to understand how the packet and application levels interact.

One interesting question about the interactions between Internet censorship and Internet surveillance is raised by our results: why do censoring routers in China keep state about open TCP connections if apparently no TCP reassembly or reordering is performed for filtering purposes? Recent work [22] showing that the availability aspect of censorship is

often implemented by the censors in concert with surveillance, *i.e.*, the privacy aspect, make this a particularly interesting question.

VI. Conclusion

We have presented results from measuring a national-scale distributed IDS system. Our results shed light on reasons why this system may have been discontinued. We showed that, due to the distributed nature of the state necessary for performing the censorship (in particular the sequence and acknowledgment numbers of a TCP/IP connection), censor effectiveness at resetting a connection even when a keyword is detected was less than 51%. We also found that diurnal patterns due to filter load were not significant enough for traffic load to fully explain why HTML response filtering appears to have been discontinued. Furthermore, we provided other empirical data about Internet censorship measurements and potential uncertainties that need to be accounted for in any future packet-level measurements of Internet censorship.

VII. Acknowledgments

We are grateful for support from NSF CAREER #0844880 for this work. Nicholas Weaver provided valuable input as well as source code for RST fingerprinting. We would also like to thank the anonymous reviewers and our shepherd, Xiaolan Zhang, for very insightful feedback.

Raw data and the graphs for all proxies, as well as all source code developed for collecting and processing the data, is available upon request by e-mailing the authors.

References

[1] L. Lamport, "Time, Clocks, and the Ordering of Events in a Distributed System," *Communications of the ACM*, vol. 21, no. 7, pp. 558–565, July 1978.

[2] T. H. Ptacek and T. N. Newsham, "Insertion, evasion, and denial of service: Eluding network intrusion detection," Secure Networks, Inc., Suite 330, 1201 5th Street S.W, Calgary, Alberta, Canada, T2R-0Y6, Tech. Rep., 1998. [Online]. Available: citeseer.ist.psu.edu/ptacek98insertion.html

[3] V. Paxson, "Bro: a system for detecting network intruders in real-time," in *SSYM'98: Proceedings of the 7th conference on USENIX Security Symposium, 1998*. Berkeley, CA, USA: USENIX Association, 1998, pp. 3–3.

[4] U. Shankar and V. Paxson, "Active mapping: Resisting nids evasion without altering traffic," in *SP '03: Proceedings of the 2003 IEEE Symposium on Security and Privacy*. Washington, DC, USA: IEEE Computer Society, 2003, p. 44.

[5] S. Wolfgarten, "Investigating large-scale Internet content filtering," M.Sc. in Security and Forensic Computing 2005/2006, Dublin City University, Ireland.

[6] Mr. Tao, "China: Journey to the heart of Internet censorship," Investigative report sponsored nu Reporters Without Borders and Chinese Human Rights Defenders, Oct 2007.

[7] "The Great Firewall Revealed," Whitepaper released by the Global Internet Freedom Consortium in December of 2002.

[8] R. Clayton, S. J. Murdoch, and R. N. M. Watson, "Ignoring the great firewall of china," *I/S: A Journal of Law and Policy for the Information Society*, vol. 3, no. 2, pp. 70–77, 2007.

[9] ——, "Ignoring the Great Firewall of China," in *6th Workshop on Privacy Enhancing Technologies*, 2006. [Online]. Available: http://www.petworkshop.org/2006/program.html

[10] J. R. Crandall, D. Zinn, M. Byrd, E. Barr, and R. East, "ConceptDoppler: a weather tracker for Internet censorship," in *Proc. of 14th ACM Conference on Computer and Communications Security (CCS)*, 2007.

[11] The Open Net Initiative (http://opennet.net), "China (including Hong Kong)," Country profile.

[12] J. Zittrain and B. Edelman, "Internet filtering in China," *IEEE Internet Computing*, vol. 7, no. 2, pp. 70–77, 2003.

[13] J. Fallows, "The Connection Has Been Reset," Atlantic Monthly, March 2008.

[14] V. Paxson, "End-to-end routing behavior in the Internet," in *SIGCOMM '96*, 1996.

[15] "Scapy (Home Page)," http://www.secdev.org/projects/scapy/.

[16] "Fragroute (Home Page)," http://www.monkey.org/~dugsong/fragroute/.

[17] G. Ziemba, D. Reed, and P. Traina, "Security Considerations for IP Fragment Filtering," RFC 1858 (Informational), Oct. 1995, updated by RFC 3128. [Online]. Available: http://www.ietf.org/rfc/rfc1858.txt

[18] I. Miller, "Protection Against a Variant of the Tiny Fragment Attack (RFC 1858)," RFC 3128 (Informational), Jun. 2001. [Online]. Available: http://www.ietf.org/rfc/rfc3128.txt

[19] M. Arlitt and C. Williamson, "An analysis of tcp reset behaviour on the internet," *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 1, pp. 37–44, 2005.

[20] N. Weaver, R. Sommer, and V. Paxson, "Detecting forged TCP reset packets," in *Proceedings of the Network and Distributed System Security Symposium, NDSS 2009, San Diego, California, USA*. The Internet Society, 2009.

[21] R. J. Deibert, J. G. Palfrey, R. Rohozinski, and J. Zittrain, "Access denied: The practice and policy of global internet filtering," *The MIT Press*, 2007.

[22] N. Villeneuve, "Breaching trust: An analysis of surveillance and security practices on China's TOM-Skype platform," Available at http://www.infowar-monitor.net/breachingtrust/.

[23] R. Clayton, "Failures in a hybrid content blocking system." in *Privacy Enhancing Technologies*, 2005, pp. 78–92.

[24] M. Dornseif, "Government mandated blocking of foreign web content," in *Security, E-Learning, E-Services: Proceedings of the 17. DFN-Arbeitstagung über Kommunikationsnetze*, ser. Lecture Notes in Informatics, J. von Knop, W. Haverkamp, and E. Jessen, Eds., 2003, pp. 617–648. [Online]. Available: citeseer.ist.psu.edu/dornseif03government.html

[25] G. Danezis and R. Anderson, "The economics of resisting censorship," *IEEE Security and Privacy*, vol. 3, no. 1, pp. 45–50, 2005.

[26] J. Aycock and A. Maurushat, ""good" worms and human rights," *SIGCAS Comput. Soc.*, vol. 38, no. 1, pp. 28–39, 2008.

[27] J. D. Tygar, "Technological dimensions of privacy in Asia," Asia-Pacific Review, Volume 10, Issue 2, November 2003, pages 120–145.

[28] ""Race to the Bottom": Corporate Complicity in Chinese Internet Censorship," in *Human Rights Watch*, August 2006, http://www.hrw.org/reports/2006/china0806.

[29] C. Liang, "Red light, green light: has China achieved its goals through the 2000 Internet regulations?" *Vanderbilt Journal of Transnational Law*, vol. 345, 2001.

[30] M. S. Chase and J. C. Mulvenon, *You've Got Dissent! Chinese Dissident Use of the Internet and Beijing's Counter-Strategies*. RAND Corporation, 2002.

[31] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," 2004. [Online]. Available: citeseer.ist.psu.edu/dingledine04tor.html

[32] R. D. et al, "Psiphon," http://psiphon.civisec.org/.

[33] Y. Sovran, A. Libonati, and J. Li, "Pass it on: Social networks stymie censors," in *Proceedings of the 7th International Workshop on Peer-to-Peer Systems (IPTPS '08), Feb 2008*, 2008.

[34] N. Feamster, M. Balazinska, G. Harfst, H. Balakrishnan, and D. Karger, "Infranet: Circumventing Web Censorship and Surveillance," in *11th USENIX Security Symposium*, San Francisco, CA, August 2002. [Online]. Available: http://wind.lcs.mit.edu/papers/

[35] A. Fiat and J. Saia, "Censorship resistant peer-to-peer content addressable networks," in *SODA '02: Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2002, pp. 94–103.