Using Clustering to Detect Chinese Censorware

[Extended Abstract]

Becker Polverini
Columbia University
Intrusion Detection Systems Lab
1214 Amsterdam Avenue, Mailcode 0401
New York, New York 10027
bpolveri@cs.columbia.edu

William M. Pottenger
Rutgers University
DIMACS and Computer Science
96 Frelinghuysen Road
Piscataway, New Jersey 08854
drwmp@rci.rutgers.edu

ABSTRACT

The Chinese government restricts access to religious, political, and pornographic content through the use of an intricate system of surveillance and censorship infrastructure. This infrastructure creates patterns that seem anomalous when compared to normal Chinese Internet traffic. Previous detection methods could neither detect zero-day attacks nor lower false-positives without first observing the attack and, later, collecting a large amount of training data. By using clustering, a technique from machine learning, fast detection of forged packets from Chinese government censorware becomes tractable. These new techniques expose previously hidden examples of sophisticated man-in-the-middle (MITM) attacks, and the algorithms used in these attacks. This paper provides both a methodology for using anomaly detection algorithms on Chinese censorware as well as examples of detected patterns. We provide circumvention strategies through the lens of exploiting computational complexity bottlenecks intrinsic to infrastructure required to censor the Internet for 250 million users. Lastly, approaches for building detectors for enterprise and civilian networks, like the power-grid, are discussed.

Categories and Subject Descriptors

C.2.3 [**Network Operations**]: Network monitoring; C.2.5 [**Local and Wide-Area Networks**]: Internet; K.5.2 [**Legal Aspects of Computing**]: Governmental Issues—*censorship*

General Terms

Security, Measurement, Experimentation

Keywords

Great Firewall of China, cybersecurity, anomaly detection, clustering, Internet surveillance, censorship, censorship circumvention

1. INTRODUCTION

In 1999 under the aegis of "guaranteeing the territorial integrity of the Mainland" and the "preservation of stable society," Chairman Jiang Zemin of the People's Republic of China (PRC) announced plans for a multiphase project that would span a decade, known today as the Golden Shield Project, or colloquially as "the Great Firewall of China" (GFC).

This mammoth project envisioned a modern infrastructure dedicated entirely to electronic information censorship and surveillance. The observable GFC, evidence of corporate complacency,[3] and tampered communication clients all point to a political and technological truth: China has one of the the most advanced censorware infrastructures in the world, affecting a netizen population equal to the entire United States population.[6]

As a result, machine learning techniques that can successfully model events, discover trends, and expose anomalous traffic given limited data are essential to cybersecurity and cybersurveillance countermeasures. For enterprise networks, the power-grid, and netizens wishing to circumvent surveillance, detection of censorware is mission-critical. Recent work in statistical relational learning has given the intrusion and anomaly detection community the methodology for model construction in the presence of limited data.[1]

The current corpus of research on Chinese censorship as it stands today is already obsolete. It has been shown in previous work that keyword-based detection of changes to the GFC provide much sociological data, but little information regarding algorithmic updates to censorware.[2] This means both application-layer and content-level monitoring are extremely difficult (and often untenable for those with limited resources). The motivation for this work is a technique by which researchers can remain abreast of changes to censorship logic at the cheapest computational cost.

Why does computational cost matter? The sheer number of netizens in China places great stress on Internet infrastructure. For example, with 2^{29} globally accessible NICs to monitor in their topology, the idea of maintaining a five-tuple for each connection is simply intractable, even given today's hardware.

If a simple tweak to a network stack implementation requires censorware to remember more fields for each user, how much have we cost the adversary? How much have we reduced our own performance? What does that mean for censorware when multiplied by 250,000,000 netizens (or 500,000,000 by 2020)? It is this perspective of increasing the computational complexity for censorware and decreasing it for the end-user that we use as an implicit metric for success in our proposal for circumvention technologies.

This paper is organized as follows: §3.1 contains the data-collection methodology; §3.2 shows results for developing a simple Chinese censorware detector; §4.1 argues why this new type of network element is faster and more adaptive than previous attempts; lastly, §4.2 provides examples of previously undetected censorship algorithms exposed by our anomaly detection method.

2. RELATED WORK

Research by [5, 7, 4, 6] prove conclusively that China's Internet censorware indeed exists. Previous Chinese censorware research has already exposed and defined the three most common, observable forms of censorship: Domain Name System (DNS) cache poisoning, reset (RST) packet flooding, and keyword censorship in both HTTP GET requests and search engines. These three mechanisms continue to be employed in censorship, but our research has revealed new techniques in each of the three catagories mentioned above.

By automating a system to check which keywords were banned in communications on Chinese network topologies, researchers were able to see, with finer granularity, the degree to which the internal Communist party leadership controlled information on the Internet.[2] Contrary to the rhetoric of the CPC, evidence exists that the censorware infrastructure is geared toward political control and not toward the removal of pornographic and illicit content, as their rhetoric would seem to indicate.[3]

Data from [5] regarding classifying types of forged RST packets in China is extremely useful to this work. [5] details the high computational complexity required to detect these forged packets absent a context. This paper is inspired by [5] to focus largely on a methodology for a less onerous means of labeling and detection of forged packets. In particular, clustering on TCP and IP header fields for packets in discrete timeslices is the approach by which we differentiate between the "legitimate" and "forged" classes.

3. METHODOLOGY

3.1 Data Collection

Active stimulation of Chinese Internet censorware involves two different approaches: either sending GET requests with sensitive keywords or sending GET requests to sensitive servers. To prevent against detecting our probes, our user-agent string was simplified to appear as Firefox 3.5 with Chinese as the native language. We observed initially that failing to alter the user-agent string can cause censoware to classify one's traffic differently than normal.

For security reasons, access points used for stimulation of the censorware were public wireless networks located throughout Beijing. However, due to the state-centric relationship between Chinese ISPs and strict regulation of communications traffic, these traffic patterns are likely similar in other Chinese urban centers: Traffic to IPs outside the PRC must traverse one of only three possible international gateways. Foreign websites were chosen such that censorware responses came from all three possible gateways in roughly equal volume.[6]

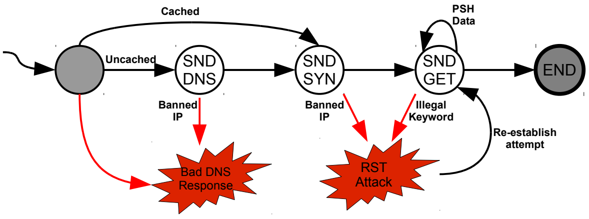

Figure 1: An automaton of interactions between censorware and an end-user viewing content.

Our keywords were based on a combination of the latest list from the ConceptDoppler project in addition to keywords from the Human Rights Watch.[7, 3] In addition to probing the censorware using keywords, GET requests were sent to a flat list of websites banned by the PRC. Furthermore, as DNS poisoning is common in China, data was collected regarding which websites were DNS poisoned (redirected to Chinese Public Security Bureau surveillance servers), had DNS records removed, or were restored to normal service (for example, the BBC is no longer blocked). If a website was DNS poisoned, Google's free, public DNS service (8.8.8.8 and 8.8.4.4) was used to resolve the address so a SYN packets could be still be sent.

3.2 Anomaly Detection based on Clustering

Clustering algorithms, without any training data, quickly divide packets into groups. This alone may not be sufficient for good anomaly detection in China; however, by combining clustering algorithms with some of the techniques used in previous research to detect forged packets, anomaly detection becomes far more tractable. For example, Chinese censorware uses well-chosen random values in its forged packets that make individual packet detection non-trivial.[5] However, clustering provides a discrete timeslice where both normal and forged packets coexist together.

This gives applications attempting to detect these forged packets a context by which to compare legitimate and potentially forged packets. For example, in a timeslice with a clustering pattern similar to a RST packet flood, if two or three RST packets contain roughly the same TTL value as the data packets also transmitted in the same timeslice, they are likely legitimate. In contrast, forged packets will likely contain radically different TTL and IP id values than the data packets (the Linux and 4.4BSD kernel have $IP_{id} = 64$). Chinese censorware cannot maintain that much state about an end-host, so it chooses random values when needed.

The WEKA workbench was used to conduct experiments with several clustering algorithms.[9] Two different types of instances were generated from the data: 1) the number of TCP, UDP, ICMP, IGMP, and "miscellaneous" packets in a given discrete timeslice; and, 2) an instance with only attributes related to TCP packets header fields in a given timeslice. Timeslices of 100, 500, and 1000 ms were compared.

Once all packet headers were amalgamated into discretized instances, various clustering algorithms from WEKA were applied. We found *simple-k-means* to be both quick and accurate. Clustering has many excellent properties in the packet sniffing context: iterative improvement and quick classification for streams. For tuning the classifier, parameters were optimized empirically: The clusters shown over a given network trace were given to an analyst for evaluation. Once the classifier was tuned, if a timeslice was classified into a cluster known to contain indicative censorware traffic, the timeslice itself was labeled "anomalous" and flagged for an analyst to reverse engineer.

4. RESULTS

4.1 Efficacy of Clustering

Clustering is highly effective at assisting with anomaly detection. A timeslice of 100 ms sufficiently detected forged packets and interference and significantly decreased the burden for the subject matter experts (SME). Since SMEs could focus on only the "anomalous" timeslices, it became far easier to reverse engineer Chinese censorware algorithms. Due to the zero-day nature of many censorware attacks, a false negative rate could not be determined, however, we feel §4.2 attests to the efficacy of our approach: By using clustering to eliminate a vast multitude of true negatives and false positives, SMEs were drawn more quickly to the needles in the haystack.

After applying a fast clustering algorithm like simple-k-means on 27,000 TCP headers, detection of timeslices likely to contain interference required only 100 ms of sampling time followed by minimal "sanity checks" on header values. Once attacks were reserve engineered after being revealed by clustering, quick detection of a wide variety of new forged RST packet and MITM attacks, once considered intractable, was easily automated.

4.2 New Censorship Techniques

4.2.1 *DNS Response Corruption*

When parsing the active dataset, an abnormally high number of corrupted checksums were detected within DNS requests to banned foreign websites. It was found that when a user specifies a DNS server other than a local Chinese ISP DNS server, checksums for the DNS response are corrupted if the requested address is poisoned domestically. In order to verify the checksum corruption, OpenDNS, also located outside of China, was used in place of Google DNS and the same pattern of DNS checksum corruption was observed.

4.2.2 *Revised Sequence Number Prediction Attack*

Through clustering we discovered characteristic network traffic revealing that Chinese censorware can now perform sequence number prediction attacks so well that only one RST packet is needed to terminate TCP connections.

When controversial content is detected, RST packets are sent to both source and destination. Detecting spoofed RST packets sent from Chinese censorware is challenging, yet clustering showed one set of RST packets containing random values and other containing the characteristic id=64, typical of the Linux kernel. Furthermore, while one packet contains a ttl between 219 and 221, legitimate RST packets contained a TTL similar to traffic with the PSH flag set (legitimate HTTP traffic).

While directly detecting these spoofed packets is difficult, using clustering as a mechanism by which to highlight particular RST packets converts the problem to detection within a context. Designing tools by which to detect single, anomalous packets with without previous state is indeed non-trivial. However, implementing tools that view these RSTs within a discrete timeslice allows for RSTs to be compared with other packets in the same timeslice, reducing the complexity of the problem significantly.

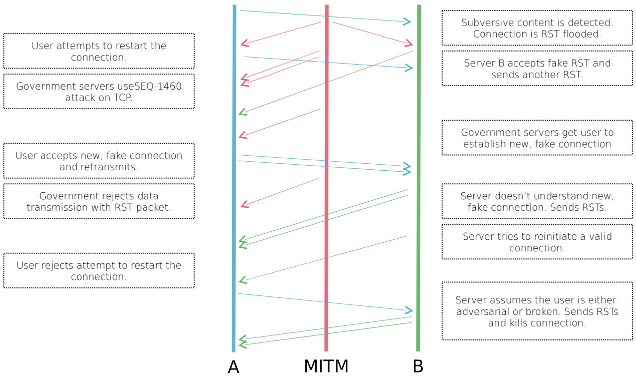
User attempts to restart the connection.
Government servers useSEQ-1460 attack on TCP.
User accepts new, fake connection and retransmits.
Government rejects data transmission with RST packet.
User rejects attempt to restart the connection.
Subversive content is detected. Connection is RST flooded.
Server B accepts fake RST and sends another RST.
Government servers get user to establish new, fake connection
Server doesn't understand new, fake connection. Sends RSTs.
Server tries to reinitiate a valid connection.
Server assumes the user is either adversarial or broken. Sends RSTs and kills connection.
A
MITM
B

Figure 2: A reserve engineered explanation for SYN-ACK forgeries seen on Chinese Internet. Censorware essentially MITM attacks TCP connections with only a few packets and minimal state.

4.2.3 *Forged SYN-ACK Reponse*

Analysis of RST packet data revealed that Chinese censorware infrastructure does not drop packets when censoring HTTP GET requests with controversial keywords. This result was expected, since previous work has already shown that the infrastructure is ultimately based on fiber optic "mirroring" that allows the packet to be both transmitted to the destination and processed by a separate system which can inject forged packets.[3]

We were able to discover forged SYN-ACK packets sent to end users after an attempt to reestablish a connection killed by a *single* forged RST packet. First, the user sends a new SYN packet to re-establish the connection. Once the infrastructure detects a new connection being established, it responds to the SYN with a forged SYN-ACK, while still allowing the first SYN packet to continue toward the destination. Eventually, the user acknowledges the forged SYN-ACK with a new ACK, retransmits the packet that triggered the RST, and receives a new RST from the censorware after the retransmission of the HTTP GET request.

This does not require a great deal of state for the censorware, as a SYN-ACK is merely a repetition of a previous sequence number and a new, random sequence number. This also thoroughly eliminates any potential chance of reconnection for the legitimate end host, as the user's operating system is very likely to send a RST to kill the legitimate SYN-ACK. The server attempting to receive the new SYN does not appear to receive RST packets, revealing that RST packet flooding is no longer necessarily bidirectional in all cases.

4.2.4 *Search Engine Collusion*

Datasets containing controversial keyword searches on Baidu, the search engine of greatest marketshare in the PRC, produced an abnormally high number of new TCP handshakes when compared to searches on Baidu with normal keywords. The payloads of TCP packets containing controversial search results from Baidu showed different Javascript than those with innocuous searches. The Javascript for controversial search terms caused a GET request for a file called “a.gif” to be sent to unregistered, domestic Chinese IP addresses. The GET request sends the users user-agent string, source IP address, and additional information in HTTP and TCP headers that could easily be used for user and operating system fingerprinting.

These servers, when pinged, responded neither to ICMP ECHO requests nor requests to download the gif directly. Using tools like *wget* and *telnet* to generate a HTTP/1.0 GET produced a 503 Service Unavailable error. Geolocation of the IP addresses revealed that all these servers are located in a technology park in Shanghai where other censorware belonging to the Ministy of Public Security is colocated.

The implication of this result is that Baidu might be colluding with the Chinese Ministry of Public Security in cracking down on controversial search results. Netizens unwillingly reveal information about their browsing habits simply by GET-ing the non-existent content. This type of Javascript was not found on similar Chinese search services, such as Sohu, Sogou, Google China, or Bing China.

5. FUTURE WORK

Having shown that computational complexity is a new and promising technique for defeating Chinese censorware, the question remains, “What kinds of simple hardware and operating system network stack tweaks will produce intractable complexity for censorship infrastructure?” Seeking out technologies that cost the end user a few cycles, but cost censorware hundreds of cycles, should be the endgame for researchers seeking to develop uncensorable Internet. Additionally, future research will tackle how end users can leverage their superior numbers and distribution to stress censorware beyond limits.

We plan to develop new routers using Click[8] that can modify IP headers and TCP options, classify traffic, and reroute traffic over IPSec using simple solutions like split-tunneling. It would be a noticeable advance if commodity hardware with multiple NICs and Click-modified Linux kernels could serve as censorware detection devices and anonimity assistance hardware for both corporations securing their networks and civilian defense applications like the power-grid.

Clustering makes detecting forged RST packets a simple problem for networks of sufficiently small size. Developing devices that can classify traffic at line-rate, send alerts, change routes, or kill a flow and restart it over IPSec upon detecting interference are crucial for defense. Questions about the limits, effectiveness, and demand for censorship detection devices remain open.

6. CONCLUSIONS

We have shown that computation complexity is a useful metric by which to evaluate the vulnerability of censorware to attack. As censorship of the Internet has transcended to the national and global scale, netizens need mechanisms to defend themselves against censorware that operates on packets in only a few cyles. We have provided a technique for anomaly detection using clustering algorithms borrowed from machine-learning that enables detection of forged packets more efficiently than previous approaches. Using machine learning we have discovered and reverse engineered new censorship algorithms, like DNS checksum corruption, malicious Javascript injection from government-corporation collusion, and sequence number prediction attacks, previously unknown to the censorware research community. In closing, we provided the groundwork for new, low-cost censorware detectors that secure civilian and enterprise from surveillance and denial-of-service: Clustering holds the key to simplying the packet processing complexity.

7. ACKNOWLEDGMENTS

The authors wish to thank colleagues at DIMACS, Princeton University, Rutgers University and the National Science Foundation. This material is based upon work supported by the National Science Foundation under Grant Numbers 0703698 and 0712139. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation, DIMACS, Rutgers or Princeton University.

We are also grateful for the help of co-workers, family members and friends. The authors gratefully acknowledge the continuing help of their Lord and Savior, Yeshua the Messiah (Jesus the Christ) in their life and work.

8. REFERENCES

[1] Menon, V.; Pottenger, W. M. *A Higher Order Collective Classifier for Detecting and Classifying Network Events.* In the Proceedings of the IEEE International Conference on Intelligence and Security Informatics, 2009.

[2] Crandall, Jedidiah R.; Zinn, Daniel; Byrd, Michael; Barr, Earl; East, Rich. *ConceptDoppler: A Weather Tracker for Internet Censorship.* 14th ACM Conference on Computer and Communications Security. pp. 1-4, 2007.

[3] The Human Rights Watch. *Race to the Bottom: Corporate Complacency in Chinese Internet Censorship.* pp. 9-27, 30, 42, 52, 67.89-106. 2006.

[4] OpenNet Initiative. *Filtering by Domestic Blog Providers in China.* OpenNet Initiative: Bulletin 008. http://www.opennetinitiative.net/bulletins/008/. 2005.

[5] Weaver, Nicholas, Robin Sommer, and Vern Paxson. *Detecting Forged RST Packets.* ICSI and U of California at Berkeley. 2008.

[6] Zittrain, Jonathan and Benjamin Edleman. *Empirical Analysis of Internet Filtering in China.* Harvard Law School, Berkman Center for Internet and Society. http://cyber.law.harvard.edu/filtering/china. 2003.

[7] Park, Jong Chun and Jedidiah R. Crandall. *Empirical Study of a National-Scale Distributed Intrusion Detection System: Backbone-Level Filtering of HTML Responses in China.* In the Proceedings of the 30th International Conference on Distributed Computing Systems (ICDCS 2010). Genoa, Italy. June 2010.

[8] Kohler, E. The Click Modular Router. Laboratory for Computer Science, MIT. ACM 2000.

[9] Holmes, G., Donkin, A., and Witten, I. WEKA: A Machine Learning Workbench. Department of CS, U. of Waikato. Hamilton, New Zealand.

[10] Zhang, Junhua, Martin Woesler. *China”s Digital Dream: The Impact of the Internet on Chinese Society.* 2004.