Can Censorship Measurements Be Safe(r)?

Ben Jones
Princeton University
bj6@cs.princeton.edu

Nick Feamster
Princeton University
feamster@cs.princeton.edu

Abstract

Understanding censorship requires performing widespread, continuous measurements "on the ground". Yet, measuring censorship is potentially dangerous, due to the threat of retaliation against citizens who perform measurements. We must balance measurement accuracy, reliability, and scalability with user safety which leads us to the question: *Can we design censorship measurements that mitigate risk to the users who consent to perform them?* Although it is almost certainly impossible to eliminate risk (or even determine if we have succeeded in doing so), we posit that we may be able to *reduce* risk with measurement techniques that are difficult to observe or distinguish from innocuous network activity. We observe that surveillance and censorship systems have different goals, and thus certain types of measurement techniques may be able to characterize a censorship system without triggering a surveillance system. We design and implement several techniques for measuring censorship that controlled tests suggest might be less risky than existing methods; we also highlight potential pitfalls, limitations, and avenues for future work.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: Security and protection (e.g., firewalls); C.2.3 [**Network Operations**]: Network monitoring

General Terms

Measurement, Security, Design

1 Introduction

Political scientists, the designers of censorship circumvention tools, and citizens at large need better information about how countries implement censorship. The state of the art for trying to answer these questions—and the only way to detect certain blocking methods—is to enlist activist citizens (or visitors) in a country to collect and report measurements about Internet censorship. Unfortunately, this approach exposes users to governments with the motive and means to retaliate and in some cases it may even be illegal to perform these measurements (e.g., amoral content in Iran [4] or *lese-majeste* content in Thailand [6]). Furthermore, governments have deployed surveillance systems that provide the means to observe censorship measurements, and 38 countries used such capabilities to crack down on dissent in 2014 [18]. Yet, gathering these measurements is at least as important as it is dangerous, so it is incumbent on us to address this problem.

In this paper, we posit that we can design censorship measurements that reduce the risk to citizens who consent to collect them, with acceptable compromises in accuracy. We explore this possibility by designing measurement techniques that we believe may be less likely to trigger a surveillance system. We believe this goal may be achievable because censorship and surveillance typically have different requirements. For example, surveillance requires capturing large volumes of traffic over time, so they must discard most of the traffic they receive (even the NSA can only retain 7.5% of their traffic [31]). In contrast, censorship systems need only store enough data to reassemble flows and store access control lists, so they do not have the same long-term storage requirement. Our analysis of two real-world surveillance systems (the NSA and our campus IDS) in Section 2 suggests that these systems are more selective than censorship systems—if traffic does not stand out from the population, it is discarded. If we can construct measurements that trigger broader censorship rules without triggering the more selective surveillance rules, it may be possible to measure censorship "under the radar".

We observe that surveillance systems can be modeled as an IDS which tries to differentiate population and measurement traffic to identify the users who collect data. Based on this model, we apply two techniques in an attempt to reduce risk: (1) adapting the measurement traffic to look like the population traffic (Section 3); (2) making the innocuous "cover" traffic from a population to look like measurement traffic (Section 4). Given a client-based measurement platform with the ability to construct raw packets (*e.g.*, OONI [16], Centinel [24]), we show that it is possible to determine whether an IP address, domain, URL, or keyword is reachable using techniques that are likely less risky than simply performing overt measurements. We evaluate these techniques using leaked

HotNets '15, November 16–17 2015, Philadelphia, PA USA Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-4047-2 ...$15.00
DOI: http://dx.doi.org/10.1145/2834050.2834066

Snowden documents, leaked censorship logs from Syria, and a reference implementation of several measurements. We acknowledge that, while unambiguous guarantees about safety are difficult to provide since the capabilities of surveillance systems cannot be known with certainty, we believe these techniques pose important considerations for future work and may nonetheless reduce risks to users, even if these risks can never be completely eliminated.

The rest of the paper is organized as follows. Section 2 enumerates our assumptions and models censorship and surveillance systems based on real systems, Section 3 describes techniques to make the measurement traffic appear more like population traffic, and Section 4 describes techniques to skew the population traffic to look more like the measurement traffic. Section 5 discusses related work, Section 6 considers ethics, and Section 7 concludes with a summary and discussion of limitations.

2 Censorship vs. Surveillance

In this section, we model censorship and surveillance systems based on real exemplars, highlight differences between the two types of systems, and explain how to exploit these differences to measure censorship in possibly safer ways.

2.1 Modeling Censorship and Surveillance

To model censorship and surveillance systems, we explore the goals, structure, and capabilities of these systems. We base our censorship system on academic papers describing the Great Firewall of China (GFC). We base our surveillance model on the NSA's system, as revealed by the Snowden leaks, and the monitoring system of a campus network that provides Internet access to approximately 21,000 users, with roughly 25,000 devices online at any time and an aggregate traffic capacity of about 150 Gbps.

Surveillance Broadly speaking, surveillance systems aim to identify bad actors and stop their behavior. In our case, the goal of the surveillance system is to identify users who measure Internet censorship and retaliate. To accomplish these objectives, a surveillance system is *user-focused*, typically retaining data to track user behavior across time. We assume a system that performs passive analysis of network traffic and do not consider active reconnaissance, such as installing malware on a user's machine or scanning end hosts. We also assume that the surveillance apparatus does not randomly attack people due to the high cost of implementing such searches (manpower and citizen goodwill).

In pursuit of these goals, surveillance systems are limited by practical constraints, which we model as a two-stage process. Surveillance systems cannot store everything, so the first stage of analysis involves discarding irrelevant traffic (most of the volume) and capturing traffic of interest to analysts. From the Snowden leaks, we know that, as of 2009, the NSA could only store 7.5% of the traffic they received [31] and although they tapped 592 10 Gbps links, they only had 69 10 Gbps links to backhaul data [38]. Therefore, the NSA engages in what we call *Massive Volume Reduction (MVR)* to reduce the volume of captured traffic by roughly 30% [28], in part by throwing away all peer-to-peer traffic. In the short term, the NSA also stores all content for three days and all connection metadata for 30 days [28]. Similarly, our campus network does not store a complete traffic capture on the network for any length of time; non-alert metadata (*e.g.*, traffic flow records, similar to call-data records in a phone network) is stored for about 36 hours; and IDS alerts are stored for about a year. After the data reduction phase (MVR), surveillance systems pass the data to a human analyst. Surveillance responses may include sending the police to a user and are typically expensive; thus, false positives are costly and may have other repercussions (*e.g.*, protests against random police action), so we assume that the analyst must winnow down the data significantly before action is possible.

Censorship The goal of censorship is to restrict communication or access to specific pieces of content. To accomplish this goal, the censorship system is *transaction-focused* and only retains data for processing real-time traffic requests. Censorship systems are often simply Intrusion Detection Systems (IDSes) that perform an action such as injecting a TCP RST if a rule is triggered. Extensive work on the GFC has validated this assumption and probed the limits of its rules and reassembly procedures [2,10,26,47]. Additionally, the cost of censoring content is low (sending a packet), as is the cost of false positives (*e.g.*, people email the censor and get content unblocked). Because censorship systems automate responses and the cost of acting, even on a false positive, is low, censors block a lot of content and often have a tendency to overblock.

2.2 Differences in Requirements

Censorship aims to prevent access, whereas surveillance systems aim to monitor network traffic. This divergence in goals leads to three key requirements differences forcing surveillance systems to be more selective than censors. We exploit this selectivity to evade surveillance while measuring censorship by blending population and measurement traffic.

1. *Storage requirements.* To monitor and track users, surveillance systems must store historical traffic traces for subsequent analysis that censorship systems do not. We can exploit this requirement by generating measurements that would require significant storage for the surveillance system to process and retain.
2. *Cost to Trigger.* The cost to trigger a surveillance system at all is higher, due to the need for human intervention, than the cost to trigger a censorship system, and the false positive cost is even higher. By blending population and measurement traffic, we can trigger censorship while forcing the surveillance system to make more selective rules or not trigger.
3. *Interest in user-attributable traffic.* The goal of surveillance is to track users, so surveillance systems care about who generated traffic. In contrast, the goal of censorship is to prevent access to restricted content, regardless of

the requester. As a result, we can measure censorship without triggering surveillance if we can generate traffic that does not appear to come from a user.

We note that simply raising alarms on all censored queries is likely an infeasible approach for user-focused targeting: An analysis of two days of leaked censorship log files from Syria shows that 1.57% of the population accessed at least one censored site, far too many people for the surveillance system to pursue [9].

Of course, in addition to these differences in design decisions, these systems may have different hardware capabilities. We assume that the surveillance system is at least as powerful as the censorship system in terms of computation and storage because we aim to exploit fundamental differences in design, not hardware capabilities.

3 Mimicking Population Traffic

By crafting measurements that mimic population traffic, we can create censorship measurements that do not trigger the surveillance system. Taking advantage of a surveillance system's need to prioritize user-attributed traffic, we disguise measurements as malware traffic. We draw inspiration from the design of malware that aims to evade IDSes, such as polymorphic blending attacks [17].

3.1 Mimicking Malware Behavior

A surveillance system may not catalog and investigate malware infections because a user may not know they are infected (and being infected with malware is not cause for suspicion *per se*). We thus aim for the unconventional goal of triggering malware rules to evade detection by the surveillance system. If measurements are difficult to distinguish from malware traffic, then the surveillance system may discard them. We generate such measurements by mimicking two out of three aspects of the malware lifecycle: infection/scanning and monetization, but not command and control. Botnets need to continually infect new hosts; we mimic this part of the infection cycle to stealthily measure TCP/IP censorship. We also mimic several types of malware monetization behavior (*e.g.*, the aspects of malware that aim to convert infections to clicks or money): sending spam and launching DDoS attacks.

Method #1: Scanning Traffic We can stealthily measure TCP/IP censorship by sending scanning and exploit traffic to potentially censored services. Machines on the Internet are constantly being scanned; botnets contribute to this scanning as they try to add vulnerable machines to the botnet. To scan a target service, we start an `nmap` SYN scan to the most commonly open 1,000 TCP ports on potentially censored services. For each service, certain ports must be open for the service to work: for example, we know that port 80 will be open on `BBC.com` because that site runs a web server. We conclude that censorship occurs if either (1) the sender does not receive a SYN/ACK; or (2) the sender receives a RST.

Method #2: Spam We send spam to (and, hence, perform MX lookups for) censored domains as a stealthy way to measure DNS and IP censorship. To perform a measurement, we perform an MX lookup for a domain's mail server, then look up the mail server's A record. If the domain resolves successfully, then we conclude that there is no censorship, although in practice there could be other confounding factors like an ISP blackholing all mail traffic to its mail server. If the mail server lookup succeeds, we initiate an SMTP connection with the IP address and send a spam message. We can measure censorship by checking that the MX and A lookups and the TCP connect all succeed.

The MVR will discard our traffic because the traffic fits the pattern of how botnets and spammers operate. If spammers send traffic to every domain in the `.com` zone, then they are bound to send traffic to censored domains; and in these cases, the MVR will discard the traffic, as it would be perceived to have little intelligence value. There is good reason to believe spammers enumerate the `COM` zone; other projects have operated a spam blackhole on a `.COM` domain that has never been made public [33], yet the blackhole still sees high volumes of spam email.

Method #3: (Part of) a DDoS Attack Botnets are also commonly used to launch Distributed Denial of Service (DDoS) attacks; we can mimic an HTTP DDoS attack to gather stealthy DNS, IP, and HTTP censorship measurements. DDoS attacks consume a small amount of resources from a large number of hosts, so as long as the traffic is being observed closer to the attackers (*i.e.* our measurement client) than it is to the victims, we do not need to send many requests to appear to be part of an attack. Repeated requests are also advantageous because we can treat each request as a measurement sample and better determine how content is being censored. DDoS attacks also significantly differ from typical user traffic, causing the MVR to discard the traffic more aggressively.

3.2 Feasibility Evaluation

The effectiveness of these techniques depends on whether it can satisfy two criteria: (1) *evasion* (*i.e.*, can it evade the surveillance apparatus); and (2) *accuracy* (*i.e.*, does the measurement capture the state of the censorship system). We evaluate these two characteristics using the reference systems and test traffic described in this section.

3.2.1 Reference Surveillance and Censorship Systems

To evaluate our measurements, we create reference censorship and surveillance systems. To demonstrate *accuracy*, we created Snort rules to mimic known censorship mechanisms and validated that we detected these mechanisms. For example, the Great Firewall of China (GFC) is known to censor certain keywords by injecting RST packets [10], so we created a Snort rule for this behavior and validated that we detected the RSTs as censorship. To demonstrate *evasion*, we show that commercial Intrusion Detection Systems (IDS), specifically Snort, cannot differentiate our measurement traffic from the population traffic. We know from leaked documents that the NSA surveillance system and GFC are functionally off-

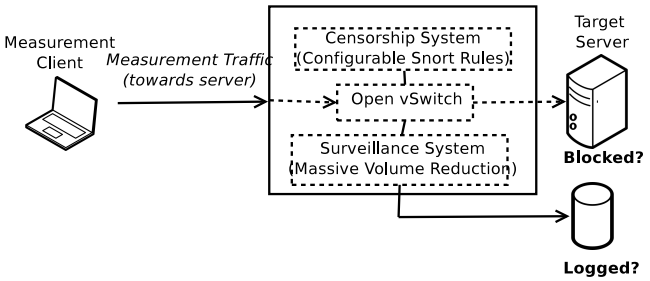


Figure 1: *We validate our measurements by checking that they* accurately *detect censorship and correctly* evade *surveillance from the reference systems.*

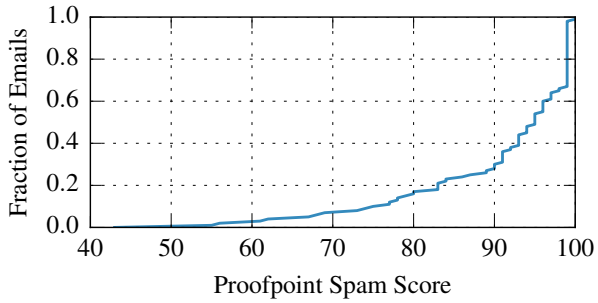


Figure 2: *This CDF shows Proofpoint's (our university spam detection service) spam scores for n=100 measurements. Possible scores range from 0 (not spam) to 100 (spam).*

path, signature-based IDS systems, like Snort, because they perform actions on the basis of rule matches [2, 20, 28, 49].

Our evaluation depends on how we configure the censorship and surveillance systems, so constructing a faithful representation is difficult and, as a result, our results cannot be conclusive for all settings and circumstances. Notably, evading a signature-based IDS is in some sense doomed to succeed when we control the signature database, and existing work implies that it is possible, at least in principle, to design application fingerprinting rules that can differentiate between our measurements and real botnets [19, 22]. However, our ruleset should be similar to a surveillance system's because Snort has a wide industry adoption (Snort's engine is used in Cisco's products [40]) and most organizations just subscribe to rulesets rather than writing their own (Cisco's products are just rule subscriptions for Snort). Given this use case, it would be expensive for a surveillance system to build their own malware detection rules, reducing deviation from our tested rules. Additionally, implementation fingerprinting would be difficult in this context because malware is constantly changing and existing work shows significant diversity, even within the same piece of malware [3, 27, 34, 35, 41].

We performed our evaluation in a controlled environment using a simple three-node Mininet topology (a client, server, and software switch) as shown in Figure 1. We ran Open vSwitch on the switch node, as well as two instances of Snort: one instance emulated a censorship system and the other instance served as our MVR. We declared a measurement successful if it can detect blocking (as controlled by our modifications to the censorship system) *without* triggering the MVR to log its traffic.

3.2.2 IDS Evaluation

Using our IDS configuration, we verified that our scanning and DDoS measurements satisfied evasion and accuracy. Our scanning traffic is evasive because we use `nmap` for SYN scanning, and many IDSes include rules to detect these scans. Scanning traffic may be discarded by the MVR because it consumes a large amount of bandwidth and is easy to distinguish from typical traffic. Durumeric *et al.* [13] found 10.8 million scans from 1.76 million hosts to their darknet of 5.5 million IP addresses in January 2014, demonstrating the high volume of scanning traffic. Our scanning measurement is accurate because `nmap` can detect which ports are open, thereby enabling us to infer censorship if a port that should be open is not (*e.g.*, port 80 for `BBC.com`). The DDoS-based measurements aim for evasion by mimicking a single source of a DDoS attack.

3.2.3 Spam Evaluation

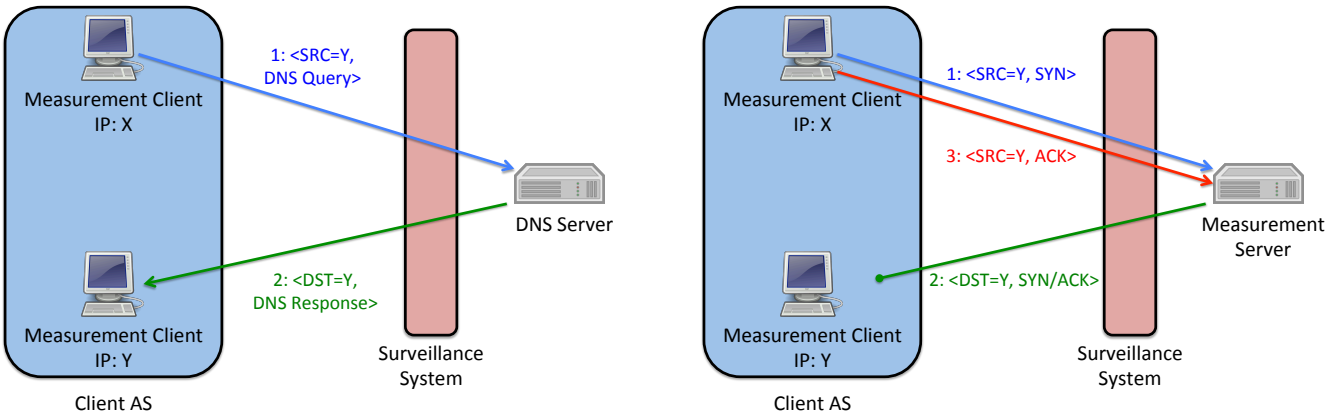
We verified the evasion and accuracy of our spam measurements with a few test measurements. We sent traffic, cloaked as spam, to an email account we control and checked whether our university's spam filter, Proofpoint, classified the measurements as spam. Figure 2 shows that Proofpoint classified our measurements as spam, validating evasion. We validated accuracy by sending MX queries from a PlanetLab node in China. We verified that the Great Firewall of China (GFC) injected bad A DNS responses for both A and MX requests for `twitter.com` and `youtube.com`.

4 Manipulating Population Traffic

We now discuss possible ways to evade surveillance by manipulating the population "cover" traffic. We create cover traffic that both resembles censorship measurements and appears to originate from every host on the network, effectively confusing a surveillance system.

4.1 Measurement Mimicry with IP Spoofing

A host can originate traffic with IP addresses that belong to other hosts in the same network; in many cases, we can also control the nature of the replies to these messages, thereby causing measurement probes to resemble similar looking cover traffic. These techniques may make it more difficult to identify which individual is initiating measurements and apply to a variety of different types of censorship measurement, including IP reachability, DNS, HTTP, keywords, and other protocols. We describe how it may be possible to perform such measurements for both stateless protocols (*e.g.*, DNS) and stateful ones (*e.g.*, protocols that use TCP). The techniques we propose can mimic measurements of stateless



(a) Stateless Mimicry. *To mimic DNS queries, the measurement client can send traffic directly to any DNS server with the spoofed IP of another device in the AS.*

(b) Stateful mimicry. *To mimic a TCP flow, the measurement client spoofs a SYN from another client in the AS, the measurement server responds to the spoofed client with a TTL limited query which dies in the network, and the measurement client sends an ACK.*

Figure 3: *Generating cover traffic from other users in the same AS.*

protocols to any destination. For stateful protocols, we can only generate cover traffic to destinations that we control.

Stateless Protocols We can measure censorship of stateless protocols to any destination. These protocols are easy to measure because, from the perspective of the surveillance system, we can mimic a complete transaction to any destination without ever involving the spoofed client. To collect measurements, we conduct measurements directly from our measurement client while spoofing measurements from other users, as shown in Figure 3a for DNS. In addition to DNS, we can use similar principles to measure IP reachability by sending TCP SYNs, checking if a SYN/ACK was correctly received, and sending a RST in response. If packets are dropped, the SYN/ACK will never arrive, otherwise, a RST provides cover traffic. While this measurement is stateless, a stateful measurement is required to test full TCP/IP reachability.

Stateful Protocols Although it is only possible to generate cover traffic to destinations that we control, the ability to mask censorship measurement for stateful protocols broadens the range of techniques we can detect. The basic idea is to send measurements towards a target with the source IP address of other hosts on the same network, making it more difficult for a surveillance system to implicate any individual host. The rise of cloud services makes it possible to host the measurement target in a location that may resemble a real target of interest, thereby evading blocking. For example, the target could be hosted on Amazon Web Services, which shares IP ranges with real measurement targets. The use of specially crafted Web requests and the use of domain fronting may also make it possible to create a wide range of stateful mimicry traffic.

One complication is the issue of replay: upon receiving a reply, a spoofed client would send a RST, possibly forcing the censorship system's TCP reassembler to stop looking at the flow. To avoid this scenario, we could TTL limit our queries to ensure that they never reach the client as shown in Figure 3b. Scanning the network from the server could yield the number of hops between the network boundary and each host, thus making it possible to set reply TTLs so they are dropped after they pass through the surveillance system but before they reach the client.

4.2 Feasibility Discussion

We argue that generating cover traffic should be feasible in practice, and that the mimicry traffic should be difficult to distinguish from real measurement probes, thus providing adequate cover for real censorship measurements. The feasibility of these techniques depend on the ASN's ability to filter our spoofed packets. Although filters to detect IP spoofing are prevalent, Beverly *et al.* determined that 77% of clients can spoof other addresses within their own /24, and 11% can spoof addresses within their own /16; these characteristics hold across a wide range of countries and regions [7]. Because so many clients can spoof adjacent IPs, our approach should work in practice on many networks.

The mimicked probes also resemble real measurements. All users in an AS generate traffic with the same properties, so an IDS that triggers on a particular measurement behavior may generate false positives for large numbers of users (if any rule is triggered at all). Hosting the measurement target on AWS may make it more difficult to identify the probe traffic based on the target IP, and identifying the traffic based on TTL might also be difficult, due to the small differences in TTLs and the natural variation in TTL values that result from routing dynamics and diversity. Traffic normalization may be able to identify odd TTL values in our packets, but

these approaches come at a high cost; for example, they may require disabling traceroute and ping [21].

5 Related Work

We believe that we may be the first to propose risk reduction for censorship measurements, but our work draws from several other areas.

Censorship Measurement Many studies have explored the extent of censorship, though we only consider the most relevant work on DNS manipulation detection and censorship measurement platforms. Several experiments have explored DNS manipulation using client measurements [1, 39]. We can apply similar analysis techniques, such as looking at the AS of returned IP addresses to determine if censorship is occurring. Several projects have also developed censorship measurement platforms [12, 16, 39].

Covert Channels/Deniable Systems Papers on the design and implementation of covert channels, ways of hiding information in innocuous data streams, are prevalent due to their importance for censorship circumvention. Covert channels have been created at every layer of the network stack, ranging from exploiting noise in the physical layer [30] to randomizing packet statistics at the network layer [48], and mimicking protocols or tunneling through implementations at the application layer [8, 14, 15, 23, 25, 29, 46]. Researchers have also analyzed the corpus of existing tools and summarized lessons for creating robust designs [19, 22].

Stealth Probing Stealth probing has previously been used for secure fault localization in routing, but has never been applied to censorship measurement. Avramopoulos *et al.* used encrypted tunnels between routers to send probe packets and securely isolate reachability failures [5], but required multiple vantage points. More similar to our approaches, Padmanabhan *et al.* developed a version of traceroute using normal packets with a unique, covert marking scheme between each pair of hosts [32].

IDS Evasion Signature-based IDS can be evaded using polymorphism; anomaly based IDS are more difficult to evade. Numerous papers have explored mimicry based IDS evasion [17, 21, 42, 45]. We extend these same ideas to surveillance evasion and censorship measurement.

6 Ethics

Our measurement methods touch uncharted ethical territory: First, measuring censorship in the first place introduces unknown risk to the users who perform these measurements. Second, although we can show that we likely *reduce* risk, it would be disingenuous (and impossible) to claim that we eliminate all risk. Even determining how much we reduce risk to users and whether the measurements are "safe enough" is a difficult question for which there are likely no fixed or quantifiable answers. As a result of this uncertainty, we have not yet deployed any measurements on real networks. Before we can encourage users to deploy these types of measurements, a better consideration of the associated risks is warranted.

We apply the principles from the Belmont report: autonomy (enabling subjects to give informed consent), beneficence (maximize benefits and minimize harm), and justice (ensuring a fair and equitable distribution of risks and rewards) [36, 44] to analyze our tool. We improve beneficence by reducing legal and physical risks of government retribution, but possible harm may include interruption of service. Many ISPs forbid spoofing in their Acceptable Use Policies (*e.g.*, [11, 43]), along with often vague restrictions such as "use of excessive bandwidth", and malware traffic may risk termination of service or incurring financial penalties. The spoofed packets that we generate may increase the load on service without their consent (reducing autonomy), and we must take into consideration evolving community norms in this regard. We believe that impact of our measurements should be similar to the now commonly accepted practice of conducting measurements with open DNS resolvers: Schomp *et al.* found 32 million open forwarders and 60–70k recursive DNS servers used by open DNS forwarders [37]. In contrast, if we conducted a single DNS measurement from every IP in an ASN's /16, we would send roughly 65k queries. Finally, we increase load on network operators by creating more spurious alerts (reducing beneficence), but our campus network shows that the increased number of alerts will be dwarfed by those from normal operational traffic.

7 Summary and Open Questions

Understanding censorship requires better measurements, but the users who perform these measurements almost certainly assume some level of risk. In this paper, we suggest that it might be possible to reduce the risk to users who perform these measurements without compromising accuracy. Although it is almost certainly impossible to guarantee that these measurement techniques are completely risk-free, we believe that it may be possible to at least demonstrate that there are ways to *reduce* user risk in many situations.

We acknowledge that the efficacy of the techniques that we propose are difficult—if not impossible—to evaluate, particularly given that the capabilities of surveillance systems are rapidly evolving and in some cases extremely difficult to ascertain. Furthermore, the risks of being observed are difficult to quantify, particularly in countries where retribution may not follow due process or even the rule of law. Yet, experience suggests that many users will assume risks to measure censorship regardless, so we believe that it is incumbent on us to attempt to design ways that might reduce risk. And, while it may always be difficult to provide unflappable assurances that any censorship measurement is completely risk-free, we believe that the result of not exploring risk-reduction techniques might be worse.

Acknowledgments

We thank Russ Clark and the GT-RNOC for detailing Georgia Tech's intrusion detection system. This work was supported by an OTF Information Controls Fellowship and NSF award CNS-1540066.

References

[1] C. Anderson, P. Winter, and Roya. Global Network Interference Detection Over the RIPE Atlas Network. In *4th USENIX Workshop on Free and Open Communications on the Internet (FOCI 14)*, San Diego, CA, Aug. 2014. USENIX Association.

[2] Anonymous. Towards a Comprehensive Picture of the Great Firewall's DNS Censorship. In *4th USENIX Workshop on Free and Open Communications on the Internet (FOCI 14)*, San Diego, CA, Aug. 2014. USENIX Association.

[3] M. Antonakakis, J. Demar, K. Stevens, and D. Dagon. Unveiling the Network Criminal Infrastructure of TDSS/TDL4. Technical report, Georgia Institute of Technology and Damballa Inc., September 2012.

[4] Article 19. Islamic Republic of Iran: Computer Crimes Law. `http://www.article19.org/data/files/medialibrary/2921/12-01-30-FINAL-iran-WEB%5B4%5D.pdf`, 2012.

[5] I. C. Avramopoulos and J. Rexford. Stealth Probing: Efficient Data-Plane Security for IP Routing. In *USENIX Annual Technical Conference, General Track*, pages 267–272, 2006.

[6] BBC News. Thailand's lese majeste laws explained. `http://www.bbc.com/news/world-asia-29628191`, 2014.

[7] R. Beverly, A. Berger, Y. Hyun, and k. claffy. Understanding the Efficacy of Deployed Internet Source Address Validation Filtering. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '09, pages 356–369, New York, NY, USA, 2009. ACM.

[8] S. Burnett, N. Feamster, and S. Vempala. Chipping Away at Censorship Firewalls with User-Generated Content. In *Presented as part of the 19nd USENIX Security Symposium (USENIX Security 10)*, pages 453–469. USENIX, 2010.

[9] A. Chaabane, T. Chen, M. Cunche, E. De Cristofaro, A. Friedman, and M. A. Kaafar. Censorship in the Wild: Analyzing Internet Filtering in Syria. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, IMC '14, pages 285–298, New York, NY, USA, 2014. ACM.

[10] R. Clayton, S. Murdoch, and R. Watson. Ignoring the Great Firewall of China. In G. Danezis and P. Golle, editors, *Privacy Enhancing Technologies*, volume 4258 of *Lecture Notes in Computer Science*, pages 20–35. Springer Berlin Heidelberg, 2006.

[11] Comcast. Acceptable Use Policy for XFINITY Internet. `http://www.xfinity.com/corporate/Customers/Policies/HighSpeedInternetAUP.html`, 2015.

[12] J. Crandall, D. Zinn, M. Byrd, E. Barr, and R. East. ConceptDoppler: A Weather Tracker for Internet Censorship. In *Proceedings of the 14th ACM Conference on Computer and Communications Security*, CCS '07a, pages 352–365, New York, NY, USA, 2007. ACM.

[13] Z. Durumeric, M. Bailey, and J. A. Halderman. An Internet-Wide View of Internet-Wide Scanning. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 65–78, San Diego, CA, Aug. 2014. USENIX Association.

[14] K. Dyer, S. Coull, T. Ristenpart, and T. Shrimpton. Protocol Misdentification Made Easy with Format-Transforming Encryption. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security*, pages 61–72. ACM, 2013.

[15] N. Feamster, M. Balazinska, G. Harfst, H. Balakrishnan, and D. R. Karger. Infranet: Circumventing Web Censorship and Surveillance. In *USENIX Security Symposium*, pages 247–262, 2002.

[16] A. Filastò and J. Appelbaum. OONI: Open Observatory of Network Interference. In *Presented as part of the 2nd USENIX Workshop on Free and Open Communications on the Internet*, Berkeley, CA, 2012. USENIX.

[17] P. Fogla, M. I. Sharif, R. Perdisci, O. M. Kolesnikov, and W. Lee. Polymorphic Blending Attacks. In *USENIX Security*, 2006.

[18] Freedom House. Freedom on the Net 2014. `https://freedomhouse.org/report/freedom-net/freedom-net-2014`, 2014.

[19] J. Geddes, M. Schuchard, and N. Hopper. Cover Your ACKs: Pitfalls of Covert Channel Censorship Circumvention. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, CCS '13, pages 361–372, New York, NY, USA, 2013. ACM.

[20] G. Greenwald. XKeyscore: NSA tool collects 'nearly everything a user does on the internet'. `http://www.theguardian.com/world/2013/jul/31/nsa-top-secret-program-online-data`, 2013.

[21] M. Handley, V. Paxson, and C. Kreibich. Network Intrusion Detection: Evasion, Traffic Normalization, and End-to-End Protocol Semantics. In *USENIX Security*, 2001.

[22] A. Houmansadr, C. Brubaker, and V. Shmatikov. The Parrot is Dead: Observing unobservable network communications. In *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, 2013.

[23] A. Houmansadr, T. J. Riedl, N. Borisov, and A. C. Singer. I want my voice to be heard: IP over Voice-over-IP for unobservable censorship circumvention. In *NDSS*, 2013.

[24] ICLab. Centinel: a Censorship Measurement Platform. `https://gihub.com/iclab/centinel`, 2014.

[25] B. Jones, S. Burnett, N. Feamster, S. Donovan, S. Grover, S. Gunasekaran, and K. Habak. Facade: High-Throughput, Deniable Censorship Circumvention Using Web Search. In *4th USENIX Workshop on Free and Open Communications on the Internet (FOCI 14)*, San Diego, CA, Aug. 2014. USENIX Association.

[26] S. Khattak, M. Javed, P. D. Anderson, and V. Paxson. Towards Illuminating a Censorship Monitor's Model to Facilitate Evasion. In *Presented as part of the 3rd USENIX Workshop on Free and Open Communications on the Internet*, Berkeley, CA, 2013. USENIX.

[27] C. Longmore. Some TDL/TDSS rootkit sites to block. `http://blog.dynamoo.com/2011/10/some-tdltdss-rootkit-sites-to-block.html`, October 2011.

[28] E. MacAskill, J. Borger, N. Hopkins, N. Davies, and J. Ball. GCHQ taps fibre-optic cables for secret access to world's communications. `http://www.theguardian.com/uk/2013/jun/21/gchq-cables-secret-world-communications-nsa`, June 2013.

[29] H. Mohajeri Moghaddam, B. Li, M. Derakhshani, and I. Goldberg. Skypemorph: Protocol obfuscation for tor bridges. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 97–108. ACM, 2012.

[30] A. Narain, N. Feamster, and A. C. Snoeren. Deniable Liaisons. In *Proceedings of the 21th ACM Conference on Computer and Communications Security*. ACM, 2014.

[31] NSA/GCHQ. TEMPORA. `http://www.spiegel.de/media/media-34103.pdf`, May 2012.

[32] V. N. Padmanabhan and D. R. Simon. Secure Traceroute to Detect Faulty or Malicious Routing. *SIGCOMM Comput. Commun. Rev.*, 33(1):77–82, Jan. 2003.

[33] A. Ramachandran and N. Feamster. Understanding the Network-level Behavior of Spammers. In *ACM SIGCOMM*, Pisa, Italy, Sept. 2006.

[34] E. Rodionov and A. Matrosov. The Evolution of TDL: Conquering x64. Technical report, ESET, June 2011.

[35] V. Rusakov and S. Golovanov. TDSS. `http://securelist.com/analysis/publications/36314/tdss`, August 2010.

[36] K. J. Ryan, J. V. Brady, R. E. Cooke, D. I. Height, A. R. Jonsen, P. King, K. Lebacqz, D. Louisell, D. W. Seldin, E. Stellar, and R. Turtle. The Belmont Report: Ethical Principles and Guidelines for the Protection of Human Subjects of Research. `http://www.hhs.gov/ohrp/humansubjects/guidance/belmont.html`, April 1979.

[37] K. Schomp, T. Callahan, M. Rabinovich, and M. Allman. On Measuring the Client-side DNS Infrastructure. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC '13, pages 77–90, New York, NY, USA, 2013. ACM.

[38] Sean Gallagher/ Ars Technica. New Snowden docs: GCHQs ties to telco gave spies global surveillance reach. `http://arstechnica.com/tech-policy/2014/11/new-snowden-docs-gchqs-ties-to-telco%2Dgave-spies-global-surveillance-reach/`, November 2014.

[39] A. Sfakianakis, E. Athanasopoulos, and S. Ioannidis. CensMon: A Web Censorship Monitor. In *Presented as part of the 1st USENIX Workshop on Free and Open Communications on the Internet*. USENIX, 2011.

[40] Snort. Does Cisco sell Snort? `https://www.snort.org/faq/does-cisco-sell-snort`, 2014.

[41] I. Soumenkov and S. Golovanov. TDL4 - Top Bot. `http://securelist.com/analysis/36152/tdl4-top-bot/`, June 2011.

[42] K. M. Tan, K. S. Killourhy, and R. A. Maxion. Undermining an anomaly-based intrusion detection system using common exploits. In *Recent Advances in Intrusion Detection*, pages 54–73. Springer, 2002.

[43] Time Warner Cable. Time Warner Cable Internet Acceptable Use Policy. `http://help.twcable.com/twc_misp_aup.html`, 2015.

[44] US Government. Code of federal regulations title 45 public welfare department of health and human services part 46 protection of human subjects. `http://www.hhs.gov/ohrp/humansubjects/guidance/45cfr46.html`, January 2009.

[45] D. Wagner and P. Soto. Mimicry attacks on host-based intrusion detection systems. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 255–264. ACM, 2002.

[46] Z. Weinberg, J. Wang, V. Yegneswaran, L. Briesemeister, S. Cheung, F. Wang, and D. Boneh. StegoTorus: a Camouflage Proxy for the Tor Anonymity System. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, pages 109–120, 2012.

[47] P. Winter and S. Lindskog. How the Great Firewall of China is Blocking Tor. In *Presented as part of the 2nd USENIX Workshop on Free and Open Communications on the Internet*, Berkeley, CA, 2012. USENIX.

[48] P. Winter, T. Pulls, and J. Fuss. ScrambleSuit: A Polymorphic Network Protocol to Circumvent Censorship. In *Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society*, WPES '13, pages 213–224, New York, NY, USA, 2013. ACM.

[49] xkeyscore@nsa. XKEYSCORE. `http://www.theguardian.com/world/interactive/2013/jul/31/nsa-xkeyscore-program-full-presentation`, 2014.