Censorship Resistance: Let a Thousand Flowers Bloom?

~~Author name(s) removed for anonymous review~~

Steven Murdoch

Steven.Murdoch@cl.cam.ac.uk

Abstract

A wide variety of options for building censorship resistance systems are available, but how best should they be combined. This paper argues that one of the most important decisions is whether one set of system options should be selected (the best), or whether there should be several good ones. Which alternative should be chosen depends on the value the censor associates with total system censorship versus partial, and what the censorship resistant system operator values in the same terms. If partial blocking is considered as almost as good as total blocking by the censor, or partial blocking is almost as strongly disliked by the network operator as total blocking, one system should be selected. If anything but total blocking is worthless to the censor, or anything but total blocking is considered valuable to the network operator, multiple systems should be selected.

1 Achieving censorship resistance

In this paper we will discuss how to best provide censorship-resistance access to an anonymous communication system. As an example, we will use Tor, but the principles discussed here apply equally well to other systems. Any such system is built of at least two components: one to resist blocking by IP address, and another to resist blocking based on payload. Additionally, the system might be designed to resist active probing.

1.1 IP-address blocking resistance

IP-address blocking resistance is achieved by either having a wide variety of IP addresses which will provide access to the network, possibly in combination with distributing IP addresses to users of the network such that the censor isn't able to discover them. "Bridges" [1] are Tor's approach to this part of the problem. Here, Tor users in countries which do not block access to the Tor network are encouraged to run a Tor node which is not listed in the public Tor directory. IP addresses of bridges are distributed such that an adversary with limited resources (in particular, IP addresses and Gmail email addresses) is unable to enumerate all bridges. By making it as easy as possible to set up bridges, it was hoped there would be many, and the distribution strategy was designed such that Tor users in countries which block Tor will be able to find at least one IP address that the censor does not know about. This approach is deployed in the Tor network and has been successful. Although China was able to learn and block a large proportion of bridges, Tor users in other countries from which access to the Tor network is blocked by IP address still can use bridges.

An alternative approach to be proposed is to have a relatively small number of bridge nodes, but have them accessible from a very large range of IP addresses. These IP addresses are shared with other services, firstly to avoid wasting IP addresses and secondly to force censors to block access to services that are not banned in the country (false positives) in order to block access to the anonymous communication system, and thus discourage the attempt. These goals are achieved through "Decoy Routing" [7, 2], where network traffic destined to the anonymous communication system is steganographically tagged and sent to an IP address in a network which supports decoy routing. The border router for this network, or a specialized computer designed for this task, detects the tag and routes the traffic to the anonymous communication system.

Decoy routing requires support from network operators, and has yet to be deployed on a large scale. However it can be considered an extension of "triangle routing"[1]. In itself, this does provide any additional IP addresses for accessing the anonymous communication network, so bridges or a similar approach is still re-

[1] `http://www.webrant.com/safeweb_site/html/www/tboy_whitepaper.html`

quired. What triangle routing achieves is to permit these bridges to be on low-bandwidth connections yet still offering high performance. Network traffic exiting the censored country does get relayed via the bridge node, to a network-entry node, but return traffic is sent directly from the network-entry node to the user, spoofing the IP address of the bridge.

1.2 Payload-blocking resistance

The advantage of IP-address based blocking is that standard IP routers are, by definition, capable of routing traffic based on IP address and can thus redirect or block traffic destined for certain IP addresses. However, more sophisticated routers and specialized censorship equipment is capable of looking within packets (known as Deep Packet Inspection, DPI) and blocking network traffic which fulfils particular criteria. Therefore censorship resistance systems need to also disguise packet content. For the purpose of this paper the payload-blocking resistance will also include resisting blocking based on port number, packet timing and packet size, because these characteristics are all closely linked.

1.2.1 Impersonating nothing

One approach to payload-blocking resistance is for network traffic to have no static characteristics. Payload data is encrypted to appear indistinguishable from random; also packet size and timing are randomized. Some sort of key-exchange is needed due to the encryption. This could be unauthenticated, such as simply sending the key in the clear at the start of the communication, or performing ephemeral Diffie-Hellman. The former is would be vulnerable to network blocking equipment capable of extracting the key and decrypting subsequent traffic. The latter is resistant to passive attack, but vulnerable to an active man-in-the-middle. Alternatively, key-exchange could be performed out-of-band or authenticated based on credentials exchanged out-of-band.

One system that takes the "impersonating nothing" approach is obfs2 [3]. This is a wrapper around the Tor bridge protocol, which works by sending the obfuscation key in the clear at the start of the communication. Subsequent traffic is encrypted by AES under keys derived from the obfuscation key. As with normal Tor bridges, obfs2 can use any TCP port, and the choice made by the obfs2-bridge must be communicated to the user, along with the IP address. No attempt is currently made to hide packet timing or size. While obfs2 does not perform any authentication or integrity checks, it wraps the unmodified Tor protocol which does perform both.

Another system with similar goals is Dust [5]. Like obfs2, network traffic is indistinguishable from random, but it aims to resist both active and passive attack by relying on a password exchanged out-of-band. Dust also performs integrity checks and replay protection. While Dust can be used over both UDP and TCP (obfs2 is TCP only), it does not provide a reliable in-order transport, so an additional layer would be needed before a TCP-based protocol such as Tor could be used. Dust also does not itself hide packet timing or length.

The advantage of the "impersonating nothing" approach is that it has no static payload signature that could be programmed into DPI equipment. However, such traffic is also unlike almost anything else seen on the Internet, so if DPI equipment can detect it, the false positive rate would likely be very low. One such test would be to measure entropy of a communication stream, using one or more of the many tests for random number generators. Any traffic with a value that is higher than expected would be blocked. Performing such a test would be challenging because it cannot be expressed as a regular expression, which is commonly the interface exposed by DPI equipment for configuring new blocking rules. Additionally, some entropy tests have high RAM and CPU usage and so would be infeasible to run directly on high-bandwidth DPI equipment which only has a handful of CPU cycles for each packet, and store a few tens of bytes for each stream. Therefore a staged approach would be needed: efficient initial tests either for entropy or to exclude known protocols would be performed on all traffic, and only selected packets would be sent for subsequent processing.

Another way of blocking "impersonating nothing" protocols is through a whitelist. Only protocols which match a particular (perhaps port-dependent) signature would be permitted. As even encrypted traffic is commonly surrounded by an unencrypted header, it would be possible to find a set of DPI rules which would permit a substantial portion of network traffic. However, protocols which were not explicitly permitted would be blocked, and so there could be a substantial false-positive rate, especially for more obscure protocols. This false-positive rate would increase if more protocols became indistinguishable from random, so one way for protocol designers to help censorship resistance would be to hide any protocol characteristics, even if they have no need for censorship resistance themselves.

1.2.2 Impersonating something

As an alternative, the censorship-resistance scheme could impersonate a particular network protocol. Tor already does this to some extent, by using TLS for its outermost cryptography layer. Initial versions of Tor made no attempt to appear like web browsing, and so Tor TLS connections included a number of distinctive character-

istics such as static fields in certificates and an unusual set of ciphersuites. Later, the Tor TLS options were made closer to that of common web browsers and web servers, by randomizing certificate fields and selecting ciphersuites closer to those commonly seen on the Internet.

However, Tor differs from typical encrypted web browsing in one important way, which is that it depends on bidirectional authentication, rather than only server-to-client. In the original version of Tor, the client certificate was sent unencrypted and thus could be used to block traffic. In later versions of Tor, the client certificate is sent during a renegotiation phase, which is encrypted. Unfortunately, the fact that renegotiation is being performed can be inferred from the plaintext, because the type of a TLS record is not encrypted, and a client certificate is of a different type from the application data which would be expected after TLS key exchange.

Efforts are underway to make Tor traffic even closer to encrypted web browsing. So far these include using more commonly seen Diffie-Hellman parameters, and extending the expiry time of certificates to be more plausible. (Both these features were used by Iran to block Tor traffic, in January 2011 and September 2011[2] respectively.) The next step to be taken will be to disguise the renegotiation step, by implementing client-to-server authentication within the Tor protocol itself.

However, impersonating TLS is not a silver bullet. TLS is very common, and blocking TLS would cut off access to many useful Internet services, but countries have been willing to do so. Iran, in particular, has blocked TLS across much of the country for periods of time[3]. While partial, these blocks were at precisely the time that access to an anonymous communication network would be most important. For this reason, Tor supports "pluggable transports", which are external programs responsible for obfuscating the Tor traffic. obfs2 is one such pluggable transport, but it was always intended that there be many such available, taking a variety of approaches. Such approaches don't only include the obfuscation technique, but also development practices – while Tor is open source, there is no reason that an pluggable transport couldn't be distributed as an obfuscated binary if that was considered to make it hard to reverse engineering and block. Existing pluggable transports include impersonating HTTP, and there are others in development, so it is hoped that it be possible to disguise Tor traffic in one of a wide variety of network protocols.

[2] https://blog.torproject.org/blog/iran-blocks-tor-tor-releases-same-day-fix

[3] https://blog.torproject.org/blog/iran-partially-blocks-encrypted-network-traffic

1.3 Scanning resistance

Being accessible at a wide variety of IP addresses, and disguising payload, are sufficient for resisting passive blocking. However, more sophisticated adversaries may also perform active attacks, but scanning IP addresses and detecting whether it is an entry point for an anonymous communication network. One option would be pro-actively scan IP addresses, to check if anonymous communication software is listening; another is to target scanning based on network surveillance. China has taken this approach, by recording which IP addresses are contacted by computers in China over TLS where the ciphersuite list matches the one used by Tor [6]. Shortly after such a connection, another computer in China probes the IP address and attempts to establish a Tor circuit, and if successful, the IP address is blocked. This approach has allowed China to almost completely block access to non-obfuscated Tor bridges.

To resist such probing, there needs to be a way for anonymous communication nodes to distinguish between legitimate access and probing, and either fail to respond to probing or return content which the censor considers innocuous. This goal can be achieved by sharing a secret between the legitimate user of the access node, and designing an authentication scheme for which an authentication failure is indistinguishable from the node not being an access node. One such scheme is BridgeSPA [4], encodes an authentication token into the TCP initial sequence number and timestamp field, and simply rejects connections for which the authentication check fails.

2 Putting it all together

For each of the components of a censorship resistance system, there are a wide variety of options available.

IP address allocation and routing: Expose bridge's IP addresses directly, or use decoy routing? Should return traffic be sent using triangle routing?

IP address, port (and optionally authentication credential) distribution: Use email, web, social networks or some combination?

Payload-blocking resistance: Impersonate no protocol or some protocol. If the latter, which one(s)?

Scanning resistance: Which scanning resistance scheme should be used, if any?

Each of the options available have their own advantages in terms of overhead, implementation and deployment difficulty, and security. One common question,

however, is whether to put limited development effort into making one censorship resistance scheme which is highly resistant to blocking, or to spread effort over multiple, less robust methods. Which option is the best is more of a question of economic incentives rather than a purely technical decision. As such it depends on how both the censor and developer of the censorship resistance system value particular situations.

2.1 Censor costs

The costs of the censor are mainly in terms of political capital and financial capital. Political capital is spent by false-positives (by annoying users of services which are not intended to be blocked) and false-negatives (by being embarrassed by failing to block sites which they should). Financial capital is spent on blocking equipment and engineering time adding and testing new blocking rules. Neither cost function is necessarily a direct proportion to the underlying quantities; there could easily be discontinuities. For example, it may be almost as good to block a few users of the censorship resistance network, as it is to block most as then the censor could say "something has been done" in response to pressure. Alternatively it may be all or nothing – the censor will only get rewarded if all the network is blocked.

2.2 Censorship-resistant network operator costs

The operator of the censorship-resistant network also has to spend financial capital on developing anti-censorship schemes, and the value of the network is increased by it being available to more users. The operator obviously wants to have their network be unblocked, but it is not clear how useful it is for the network to be partially blocked. Some operators may view this as being adequate, because users can adapt their behaviour to use the unblocked mechanisms. Other operators may require an all-or-nothing approach to network blocking.

2.3 One system or several?

If the censorship-resistant network operator's priority is to put emphasis on avoiding any blocking of the network, or it is believed the censor needs only to block some of the users (but not necessarily all) the best solution is to choose one censorship-resistance system which impersonates the protocol which can be reliably impersonated and costs the most to the censor (in terms of political and financial capital). Hopefully this cost will be too high, and the system will remain unblocked, but if the censor is able to block the system all access to the anonymous communication system will be blocked.

Alternatively, if the network operator values some access to the network being preserved, or if the censor highly values blocking all of the network (and is uninterested in partial blocks), then it is better to impersonate several different popular protocols, and to distribute traffic between them. In order to block all users, the censor would have to block all protocols selected. On the other hand, this approach has reduced the cost of only blocking some users because the censor needs to only block the least-valuable network protocol.

In terms of conflict theory, achieving total blocking resistance is a best-effort problem, and achieving the network being mainly unblocked is a sum-of-efforts problem. Between these two extremes simulation could be used to select an optimum approach and where multiple obfuscation schemes are selected, how to optimally distributed traffic over the different transports.

3 Conclusions

This paper has discussed the various components of a censorship-resistant system for accessing an anonymous communication network. Options for each component have been described, but choosing between which composition depends not only on the technical features of each component, but of the subjective cost functions of both the network operator and the censor. The extent to which the censor and network operator value all-or-nothing censorship or censorship-resistance, respectively, is dominant in deciding whether to select one approach or several, and being able to quantify these functions would permit simulation and optimization. However, it remains unclear how to best estimate these functions, especially for the censor.

References

[1] DINGLEDINE, R., AND MATHEWSON, N. Design of a blocking-resistant anonymity system. Tech. Rep. 2006-1, The Tor Project, November 2006.

[2] HOUMANSADR, A., NGUYEN, G. T. K., CAESAR, M., AND BORISOV, N. Cirripede: Circumvention infrastructure using router redirection with plausible deniability. In *Proceedings of the 18th ACM conference on Computer and Communications Security (CCS 2011)* (October 2011).

[3] KADIANAKIS, G., AND MATHEWSON, N. obfs2 (the twobfuscator). Tech. rep. `https://gitweb.torproject.org/obfsproxy.git/blob/HEAD:/doc/obfs2/protocol-spec.txt`.

[4] SMITS, R., JAIN, D., PIDCOCK, S., GOLDBERG, I., AND HENGARTNER, U. Bridgespa: Improving tor bridges with single packet authorization. In *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2011)* (October 2011), ACM.

[5] WILEY, B. Dust: A blocking-resistant internet transport protocol. Tech. rep. `http://blanu.net/Dust.pdf`.

[6] WINTER, P., AND LINDSKOG, S. How china is blocking tor. Tech. rep., Karlstad University, 2012. `http://www.cs.kau.se/philwint/pdf/torblock2012.pdf`.

[7] WUSTROW, E., WOLCHOK, S., GOLDBERG, I., AND HALDERMAN, J. A. Telex: Anticensorship in the network infrastructure. In *Proceedings of the 20th USENIX Security Symposium* (August 2011).