First Glance: An Introductory Analysis of Network Forensics of Tor

Raymond Hansen
Department of Computer and Information Technology, Purdue University, hansenr@purdue.edu

Scholarly Commons Citation

FIRST GLANCE: AN INTRODUCTORY ANALYSIS OF NETWORK FORENSICS OF TOR

Raymond Hansen
Department of Computer and Information Technology
Purdue University
401 N. Grant Street
West Lafayette, IN 47907-2021
Phone: (765) 796-9482
Fax: (765) 496-1212
hansenr@purdue.edu

ABSTRACT

The Tor network is a low-latency overlay network for TCP flows that is designed to provide privacy and anonymity to its users. It is currently in use by many as a means to avoid censorship of both information to be shared and information to be retrieved. This paper details the architecture of the Tor network as a platform for evaluating the current state of forensic analysis of the Tor network. Specific attempts to block access to the Tor network are examined to identify (a) the processes utilized to identify Tor nodes, and (b) the resulting exposure of potentially inculpatory evidence. Additional known, but yet to be perpetrated, attacks are examined for a more holistic view of the state of forensics of the Tor network. Based on the combination of these studies, there is some evidence that a specific, individual flow of traffic over the Tor network is attributable to a single entity. However, the content of that flow has not been compromised within the Tor network. As such, the inculpatory evidence required for legal action is limited at this time.

Keywords: Tor, Forensic Analysis, Privacy & Anonymity

1. INTRODUCTION

Tor is a popular 2nd generation implementation of the onion routing topology (Tor, 2012). This topology was developed in work for the U.S. Navy Research Lab in the mid-1990s (Goldschlage, Reed, Syverson, 1996). This is an overlay network of the public Internet that is designed to provide online privacy and anonymity to its users through two different mechanisms. The first mechanism of privacy relies on multiple encryption iterations in order to obfuscate the entirety of traditional IP packets. The second mechanism relies on seemingly random network ingress points, routing hops, and egress points to diminish an external observer's ability to identify the end-to-end path of a traffic flow through traffic analysis or network surveillance.

Since Tor is designed to provide anonymity, it would be beneficial to have at least a rudimentary definition of anonymity. One could simply say it is the state of being anonymous. But that seems circular in definition and is less than useful for any rigorous discussion. So, instead, anonymity could better be thought of as "the state of not being identifiable within a set of subjects" (Pfitzmann & Kohotopp, 2001, p.1). Even using this definition, there are still murky waters concerning attribution of the Tor flows to a particular entity.

The anonymity provided by this network is intended to allow users to send and receive data across the network with little fear of being identified by an external observer, regardless of friend or foe status. Since this tool can be utilized by both the innocuous and nefarious with no immediate mechanism to distinguish between them, and the anonymity and privacy provided may afford the impetus for illicit online behavior, many governments and law enforcement agencies have become increasingly concerned with the operations of this network. Additionally, many nations wish to censor

communications both within and across their borders, including: Belarus, China, Cuba, North Korea, Syria, and more (Ho, 2009). In fact, recently, all encrypted traffic was blocked within Palestine (Ma'an News, 2012; York, 2012), all HTTPS traffic was blocked in Iran (Rezale, 2012), and access to Tor Directories and Bridges have been blocked by State-sponsored service providers (Tor Project, 2009; Tor Project, 2010, etc.). For many citizens within these countries, access to Tor was a potential solution to this censorship, if they were able to access the network. As such, shortly after these denials of entrance to the Tor network, access to the nodes was restored by the semi-anonymous operators of the Tor network, much to the dismay of the State.

Discussions of Tor Hidden Service nodes and client-side views of Tor are left for future work. This paper focuses on the network-side attributes of Tor as well as the inculpatory and exculpatory evidence from the data communications and networking aspects of its operations. This view uses cybersecurity as the lens to focus this view. As such, this view is driven by the understanding that cybersecurity and digital forensic analysis are inextricably linked. These two are complementary, supplementary, and co-dependent. An analysis of the security processes gives way to the mechanisms that are useful for performing forensic analysis.

This paper identifies types of uses and users of the Tor network which includes highlighting the history and operations of the Tor network while differentiating its operation from that of the traditional routing mechanisms in use on the public Internet. It then identifies known and executed "attacks" against the Tor network and provides details of their execution process(es). Utilizing previous State-sponsored efforts, earlier successful forensic approaches are identified for the purpose of determining if unique individual flows were identified, or merely particular nodes and their aggregate traffics were determined to be participatory. It will be shown in this paper that even though it might be possible to identify a particular flow of traffic, attribution to any individual is not provable through Tor network attributes. Yet, that correlation of traffic flow to a particular host device may be inculpatory enough for some jurisdictions. Next, known *potential* attacks against the Tor network are detailed. It should be noted that while these are known potential attacks, there is not yet any proof that they have been. These are intended to provide a foundation to identify potential forensic analysis of operations of the Tor network. Lastly, we detail the potential evidentiary findings of a generic Tor node and any applicable inculpatory indicators therein.

2. THE TOR ARCHITECTURE

Tor is a 2nd generation implementation of the onion routing topology initially developed in work for the U.S. Navy Research Lab in the mid-1990s (Goldschlage, Reed, Syverson, 1996; Tor, 2012). This implementation of the onion routing topology is intended to be a low-latency overlay network for TCP flows over the public Internet that intends to provide privacy and anonymity to its users. Specifically, Tor provides the functionality that "prevents a user from being linked with their communication partners" (Loesing, Murdoch, Dingledine, 204, 2010).

While the original design goal of the Tor network was to provide significantly more privacy to a user than provided in default Internet communications, Tor has recently been used by evade state-sponsored censorship attempts (Loesing, Murdoch, Dingledine, 2010). Dingledine has noted that there is an "ongoing trend in law, policy, and technology" that "threaten anonymity… (and) undermine our ability to speak and read freely" on the public Internet (n.d.). For example, in early 2012, Iran disallowed access to any sites that utilized HTTPS (Kabir News, 2012). Also, in May, 2012, the Palestinian government shut down eight news websites for posting critical opinion pieces of the president (Hale; OONI, 2012). In mid-2012, the Ethiopian Telecommunications Corporation began performing deep packet inspection on all ingress and egress traffic coming in to the country. As Ethiopia's only service provider, they had direct access to all such traffic (Runa, 2012). York has provided multiple cases where there have been calls for additional censorship, the creation of a

censorship body, and even examples where citizens have been arrested for political or religious reasons (2012b). And so then, Tor is intended to provide not only privacy in these types of scenarios, but anonymity by providing protection against eavesdropping and man-in-the-middle attacks. Additionally, by using multiple iterations of encryption and intermediating nodes defining a radically different path from a client to its ultimate destination, deep packet inspection, traffic analysis, and timing analysis attempts are mitigated.

2.1 Traditional Internetwork Routing

In a traditional routing environment, each packet from a source is addressed with the ultimate address of the destination. A router will then examine the destination address, choose the best neighboring router based on the longest-match algorithm, and then forward the packet to that destination. Each router along the path from source to destination performs this action on every packet in a flow. This process, while resilient to the dynamic topology of the Internet, is inherently insecure. This process potentially leaks information about both the source and destination of the flow, the content of the packets, as well as the path that is used to deliver these packet flows. As such, there is little inherent privacy, and no anonymity in the default routing process.

A pair of options that were defined in the original IP standard for use in networks was loose source and record routing. These options allowed the source node of an IP packet to explicitly specify a partial, or complete, path for the packet to follow (RFC 791, 1981). Source-routing overrides the traditional routing process using the longest-match algorithm to search the routing table that occurs on each router in a path. Instead, the packet is routed along the destinations listed in the source-route contained in the IP packet header. This approach to routing a packet through the network relies on the assumption that the source of the IP packet has a complete view of the network topology to the ultimate destination and can provide explicit path information in the packet. By specifying each hop from the source to destination, or minimally at least one intervening hop, a "trusted" path can be used to deliver packets. Yet, even though the path may be "trusted", this IP options approach provides no additional security of the packet; neither is there additional privacy, confidentiality, or integrity of the packet using this approach. Further, the source and destination node addresses are still directly readable by any passive monitor in this path. Extending this concept, the use of this IP option could potentially be used to reduce privacy by exposing known "trusted" nodes in the path to additional scrutiny by any interested party. Because of this, most routers on the public Internet ignore the path specified by a host with this option or discard the packet altogether (Zwicky, Cooper, and Chapman, 2000).

Enhancements and additions have been made to attempt to add privacy to this original routing process. Encryption, in the form of SSL and TLS, has been added to secure the payload of each packet from inspection (intrusion) by outside parties. However, this approach does nothing to obfuscate the source or destination of the flow. Also, simply encrypting the payload still allows traffic analysis and timing analysis to occur and potentially identify the type of payload being transacted even though the exact payload is not identified. Other encryption approaches have been added to this process to further obfuscate the packet payload through the use of IPsec, which encrypts the IP packet payload. This approach hides the TCP or UDP port numbers, which makes protocol identification more difficult, but not impossible. Traffic analysis and timing analysis are still possible, as well as the immediate identification of the source and destination IP address information for the source and destination nodes. This is perhaps better recognized as providing integrity and confidentiality of the payload, not privacy or anonymity.

An additional implementation of IPsec allows for full encapsulation and encryption of the entire packet. This is referred to as Tunnel-Mode. In this implementation of IPsec, the packet is fully encrypted and then encapsulated within another packet. This new packet typically will have different source and destination IP addresses than that of the encapsulated packet. These new IP addresses are

the end points of a VPN tunnel. Each router along the path from the new source address to the new destination address passes the encapsulated packet with no knowledge of the original source and destination IP address. This approach provides some privacy for the packet flow, as an external observer will have difficulty identifying the payload information. However, the VPN end points are still fully capable of examining the payload and identifying the communicating participants.

2.2 Routing in the Tor Network

The Tor network is based on the original onion routing architecture described in Hiding Routing Information by Goldschlag, Reed, and Syverson (1996). The Tor network, while being based on the original onion routing architecture, has deviated from and enhanced significantly in operation. There are three defined connection-establishment processes for Tor (Dingledine, Mathewson, n.d.). This paper focuses on the most recent of those processes (v3).

A rudimentary understanding of the architectural components is critical to the discussion of the Tor network. So, a short description of the major components of the Tor network is included here:

- Bridge (Bridge Relay) – A Tor Relay that does not report to the Tor Directories and can only be reached through knowledge of their location on existing networks. These nodes are used to circumvent filtering techniques against known Relays.

- Circuit – The onion router path from a Host through a public internetwork to a subset of Relays that requires a unique set of encryption between the host and each Relay in the path. Routing of packets between the Host and each ordered Relay is handled with the traditional packet forwarding mechanism by the routers within the public internetwork.

- Directory – One of a set of Tor nodes that collect, verify, format, and disseminate Relay status information throughout the Tor network.

- Host – The source node in the Tor network that runs the Tor software proxy, initiates circuit construction, and passes data through a local SOCKS proxy into circuits for traversal over the encrypted paths.

- Mix – The combination of Relays provided by a Directory that a Host will use to establish a Circuit.

- Relay – A node in the Tor network that acts as a circuit routing device. This node responds to circuit construction requests by establishing an encrypted channel between itself and host. These node forwards packets based on the Circuit ID contained in the Tor packet header. Availability and capacity of this node are reported to the Tor Directories.

In order for the Tor network to be able to process any connections, each of the Relays must report their availability to a set of Directories. This is shown in Figure 1 below. Each of the Relays reports their name, exit policy, available capacity, and other defined information to the Directories, which then verify the reported capacity and utilize a voting process to establish a network consensus document for distribution to Hosts (Tor Directory Protocol, version 3, 2012).
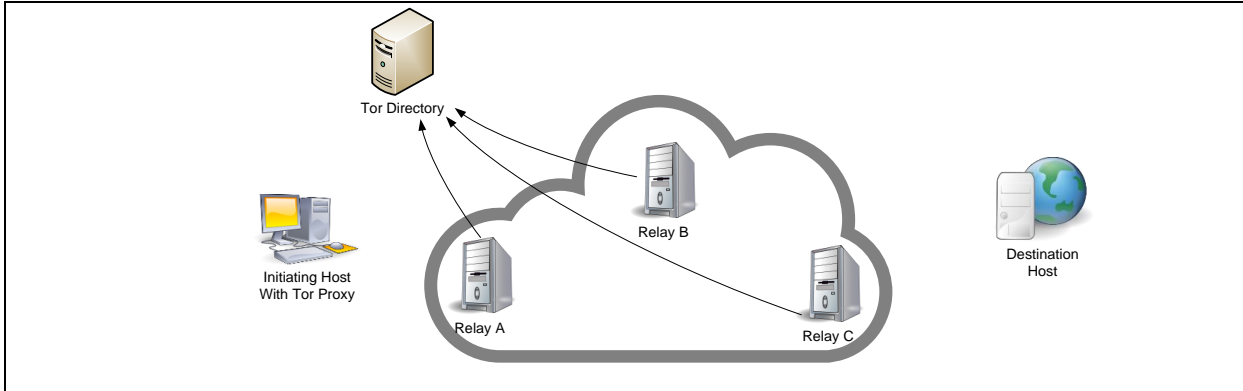
Figure 1 Relay Registration

Hosts must learn of the available Relays by periodically contacting a hard-coded Directory from a set of Directories, as shown in Figure 2. This connection is actually established over a single-hop onion route to obfuscate the source Host from the Directory. The Directory will respond with a consensus document to the requesting Host, which will then determine the mix of Relays to be used to establish the onion route.
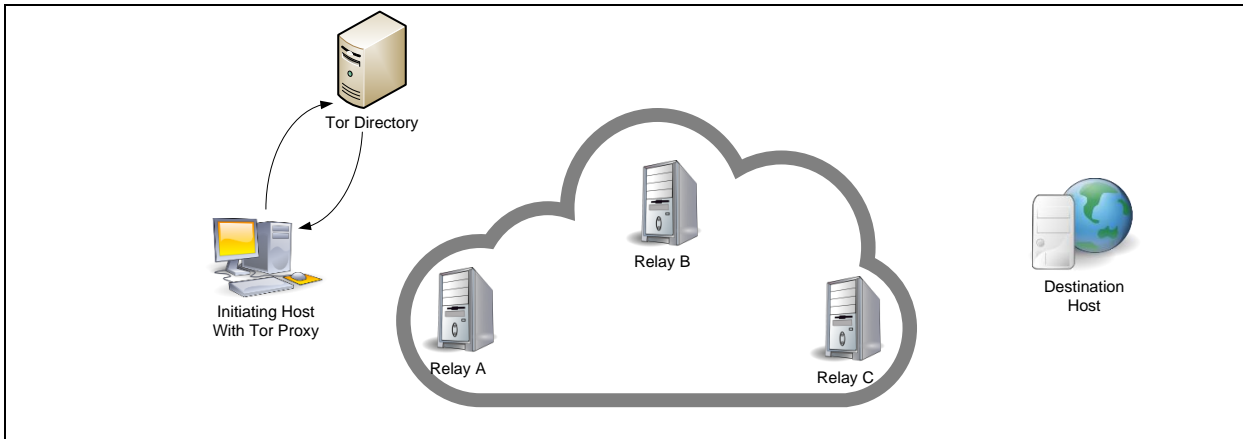

Figure 2 Mix Request & Response

Based on the mix derived by the Host from the previous steps, the Host will iteratively build an onion route a single Relay at a time, choosing the exit Relay first. This Relay is selected based on it having an acceptable exit policy and then the remaining Relays are pseudorandomly chosen (Dingledine & Matthewson, 2012). The host will initiate the first layer in the onion route to the initial Relay using the Diffie-Hellman Ephemeral Key Exchange to establish a TLS connection based on those derived keys, as seen in Figure 3, using preferred ciphersuites such as TLS_DHE_RSA_WITH_AES_256_CBC_SHA, TLS_DHE-RSA_WITH_AES_128_CBC_SHA, and SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA (Dingledine & Matthewson, 2012). Then, the Host sends a Tor specific command (CREATE) to indicate a new circuit is needed. The Relay (Relay A) will respond with a CREATED command if successful. The first layer of the onion route has then been created and is uniquely identified by a CircuitID that is meaningful only between the Host and Relay. Traditional routing processes will still occur to determine the path between the Host and the first Relay, and vice versa, in the onion route.
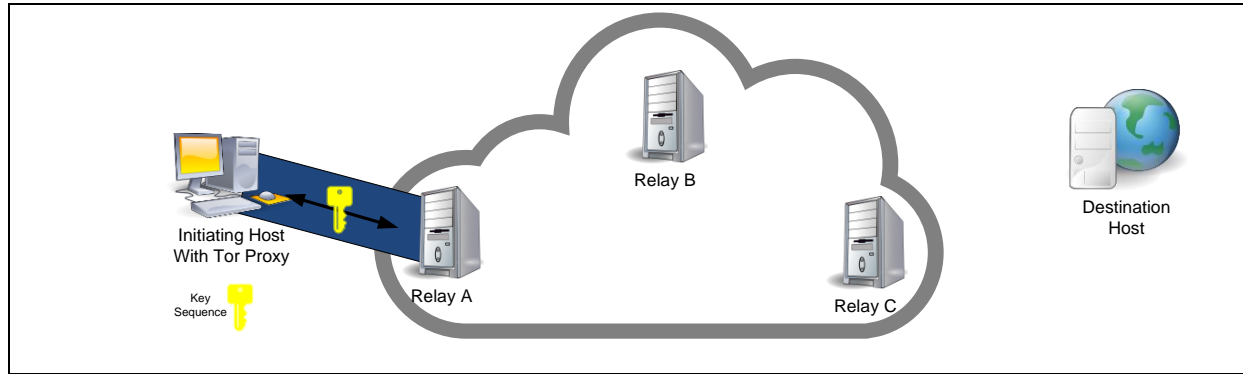
Figure 3 Building Channel to Relay A

The Host will then extend the circuit established in the previous step by negotiating the second layer of the onion route with the second Relay (Relay B). Again, the Diffie-Hellman Key Exchange is used to establish shared keys for the TLS connection, as seen in Figure 4. The Host sends an EXTEND command to Relay A, encrypted using the shared keys of Host-Relay A, which then forwards the content to Relay B. That content is a CREATE command for Relay B and is encrypted with the Host-Relay B keys. This then establishes the second layer of the onion route circuit. So, the host now manages two shared keys, one for each of the Relays that it has established connections with to build a circuit. The first layer of the circuit (Relay A) is unable to observe the content of the cells that flow over it from the Host to Relay B.
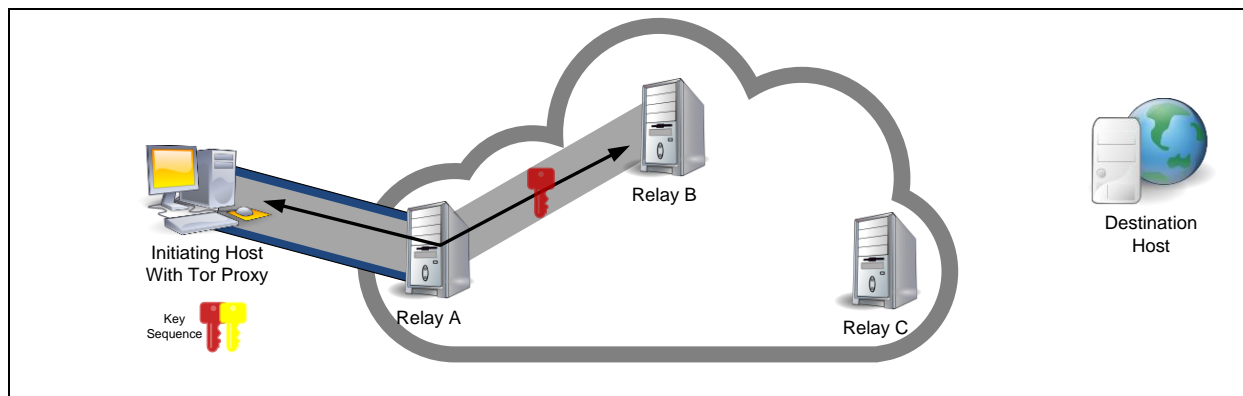
Figure 4 Extending Channel to Relay B

The onion route has one additional layer added to it through the use of a third Relay (Relay C). The same process is used to extend the circuit as in the previous steps, with an additional round of encryption occurring for the newly established connection between the Host and Relay C that traverses both Relay A and Relay B, as shown in Figure 5. Relay A and Relay B both process an EXTEND command, while Relay C processes a CREATE command. At this point, the onion route is complete. A Host now has an anonymized connection through the public Internet.

Figure 5 Extending Channel to Relay C

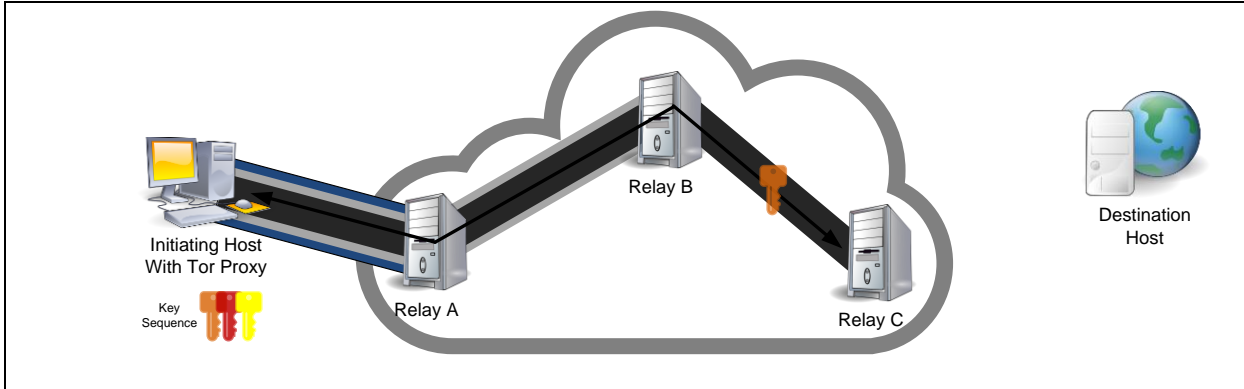Using this established onion route, the Host can interact with any public resource as seen in Figure 6 by using the RELAY commands within the cell to be routed. Relay C will perform all DNS queries on behalf of the Host so as to limit any monitoring of the Host in an attempt to identify specific destinations or resources that are being accessed. In this way, each node in the path (initiating client, Relays, and destination device) only knows the identity of its immediately adjacent neighbors (Dingledine et al, 2004). It should be note, however, that simply using the onion route doesn't provide any protection of the data flowing between Relay C and the destination. In fact, monitoring traffic at an exit node is one of the viable methods for attempting to identify users of the Tor network. So, any unencrypted traffic between the exit node (Relay C) and the destination can be observed, recorded, and examined for specific content.
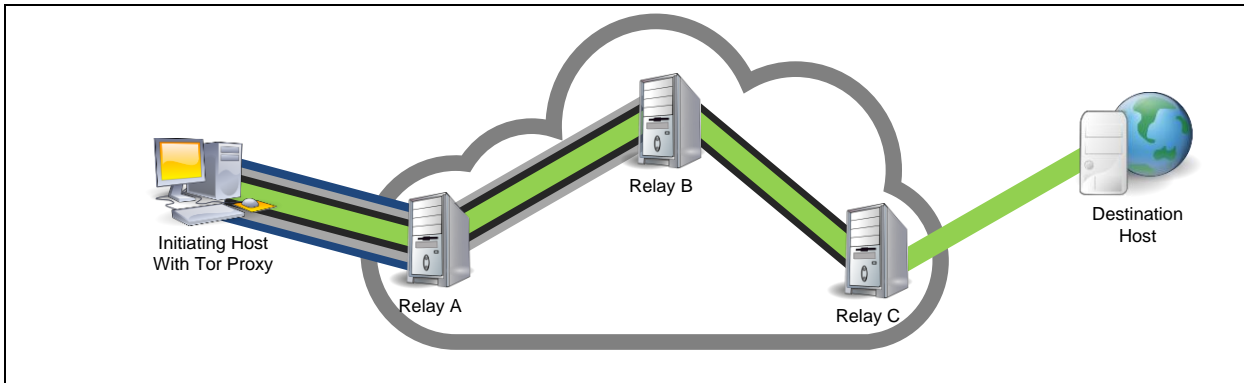
Figure 6 Anonymous Communication via Tor

A Host does not use, or maintain only a single onion route. In fact, it regularly constructs and terminates circuits in a timely manner. That is, a Tor Host may initiate a new circuit a frequently as once per minute, and will terminate an unused circuit every five minutes. Most circuits are terminated after less than ten minutes of use as a mechanism to reduce traffic analysis attacks against the network and those using the network. A simple example of the diversity of connections maintained by a Host can be seen in Figure 7.

Figure 7 Multiple Distinct Concurrent Channels from Host

So then, this approach to obfuscating data provides anonymity of the source, as only the Directory and first Relay know the identity of the initiating client and not the identity of the destination. The remaining Relays will not know the identity of the initiating client. Only the exit Relay (Relay C) will know the identity of the destination.

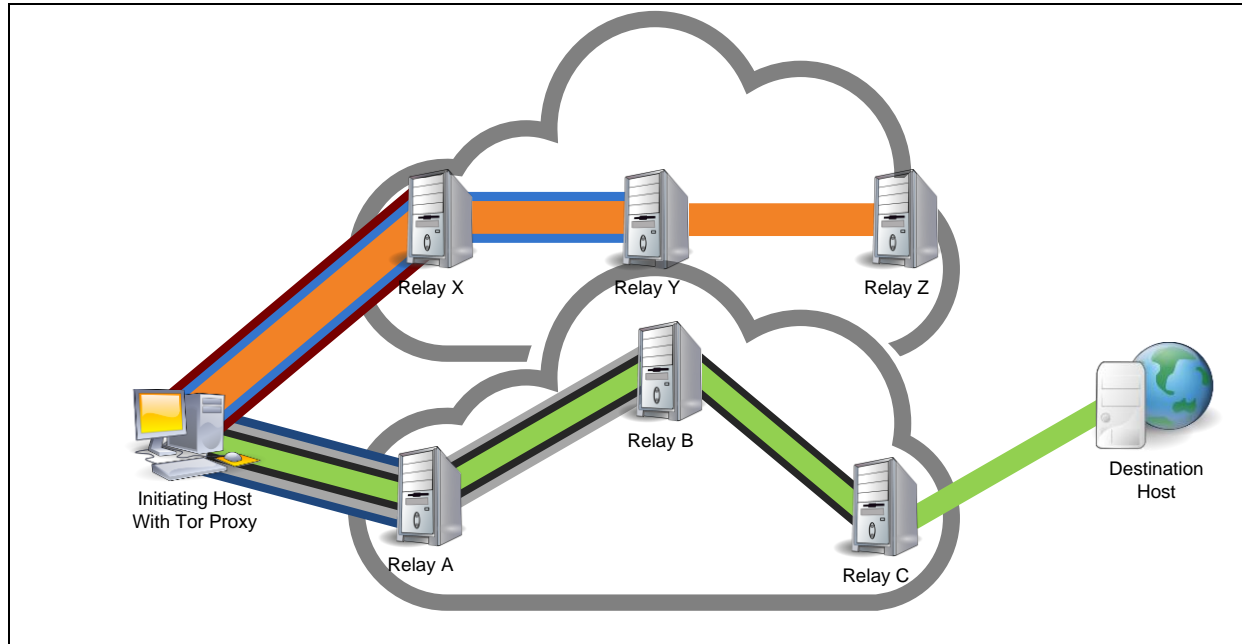# 3. CURRENT TOR NETWORK ATTACKS

There are many well-known mechanisms that can be used to censor materials on the Internet. These have ranged from simple source/destination IP address filtering, DNS filtering/injection/hijacking, as well as content filtering (Ho, 2009). However, these approaches assume a user with little expectation of privacy or anonymity. If a user utilizes encryption or other obfuscation approaches, then these censoring tasks become more difficult. This is the intent of those utilizing the Tor network for access to resources on the Internet – make difficult the analysis of their traffic content and its characteristics through encryption and anonymizing services. So then, for authorities to keep pace with the censorship of content flowing over the Tor network, they must also attempt to censor the utilization of this network.

Previously, performing deep packet inspection by a state-owned service provider would provide complete access to the traffic being sent or received by any user. At any ingress/egress point of the network, analysis could take place through a legal (or illegal) wiretap or other intercept process. The network could even be designed to have several "chokepoints" where this analysis could be expedited. Even if non-typical protocols were implemented by privacy- and anonymity-seeking users, digital fingerprinting and traffic analysis would typically be sufficient to identify potentially "dangerous" network traffic. However, the Tor network significantly, and sometimes dramatically, complicates this process for state-sponsored filtering.

As nation-states attempt to quell access to the Tor network, opposing actions have been taken to restore anonymous capabilities. The following is a partial listing that highlights a subset of current "attacks" against Tor (the service), the Tor network components, and the corresponding responses by Tor maintainers to circumvent these censorship attempts.

In September, 2011, Iran blocked access to the Tor network by adding a filtering rule to their national border routers. This rule specifically identified Tor traffic and filtered this traffic while still passing all other traffic that was not previously blocked. This filter identified a specific component of the traffic flow that established the Tor connections. Tor is designed to make all network traffic appear that of a client accessing an HTTPS web server. As such, this requires a secure handshake process. It was this handshake process that was uniquely identified by the Iranian government. The SSL session certificate expiration time on Tor Relays was set to two hours, which is uncharacteristic of certificates issued from a real, public certificate authority. The resolution, which was implemented the same day by Tor engineers, was to lengthen the expiration time of the Relay certificates to more closely emulate that of a true SSL certificate implementation (Tor, 2012).

The standard connection initiation process involves a client requesting a listing of Relays from a Directory. These Directories are statically coded in the Tor client, meaning they are directly known by any party using, or monitoring, Tor. Because of this, the Chinese government simply blocked access to the IP addresses of each of those known Directories, effectively eliminating a client's ability to establish a connection into the Tor network (Lewman, 2009). Prior to this event in 2009, the Tor developers added functionality to the Tor architecture to resolve the potential of this vulnerability being exploited. The resolution to this vulnerability was the addition of Bridges into the Tor architecture. A Bridge functions just like a Tor Relay, but is not registered in the public Directory servers and is used as the first (entry) or last (exit) Relay in the path So then, a Tor client must learn of these Bridge locations (addresses) by some other means. This is accomplished by using the graphical interface to Tor, Vidalia, or by manually configuring the Bridge IP addresses into the Tor client after retrieving the information from the Tor website. Considering that both of these approaches are likely filtered by any entity blocking access to Directories and Relays, an email option is also available. (Tor Projects, 2012). This architectural adaptation effectively circumvented the blocking of Directories and Relays.

In late-2011, the Chinese government began blocking access to Bridges within China. Analysis of the blocking actions revealed that connections were initially allowed, but were terminated within a matter of minutes (Wilde, 2012). Further analysis showed that filtering process was performing a passive fingerprint analysis of the communications between a host and the Bridge and then attempted to establish an active connection to the Bridge. It was determined that an identifiable client-side parameter of the SSL negotiation was unique to the communications of hosts to Tor Bridges. The intensity of the active scans suggested near line rate deep packet inspection, which requires significant processing capabilities (Wilde, 2012). This type of scanning lasted for only a few short weeks before it was abruptly terminated (Wilde, 2012). But, was apparently active again in March, 2012 when additional evaluation was performed (Winter and Lindskog, 2012). This attack is further mitigated in the most current handshake negotiation process, as TLS has supplanted SSL, and the parameter in the SSL/TLS cipher list that was uniquely identifiable has thus been removed (Wilde, 2012).

Not long after the Chinese government identified the client-side cipher list issue to block access to Bridges, the Kazakh government also began blocking access to Bridges within Kazakhstan (Lewman, 2012). Again, unpublished Bridges were being identified through continual deep packet inspection. Until this time, Relays were not blocked within the country and Tor was used somewhat extensively. Analysis of traffic arriving at the Bridges suggested no active scans, as the Chinese government had undertaken. However, it was determined that the passive scans of deep packet inspection have identified a unique parameter of the server-side hello message in the TLS negotiation. An additional tool, called obfsproxy, while not directly part of the Tor architecture, continued to function within the country during this time (Lewman, 2012). Thus, allowing citizens anonymized access to the Internet. As of this writing, no resolution to this filtering approach was found.

In the Bad Apple Attack an insecure application, such as a web browser, used over Tor is capable of revealing the Host's IP address thorough information leaking via Flash, Java, JavaScript, etc. In this

case, they exploited a BitTorrent application. By leaking this information, there is the potential to be able to attribute particular quantities and patterns of packets to the source as member packets of this flow. By leaking this information, all other flows from that host could be considered suspect (Le Blond et. al., 2011).

Many more attacks against Tor can be found by browsing the Tor blogs and brief Internet searches. With the recent political unrest in Egypt, Syria, Libya, and more, there are moribund examples of State-sponsored censorship attempts. Interestingly, censorship is not necessarily only occurring locations of political upheaval. Access to the Tor website is blocked by non-government controlled cellular providers in both the UK and US (Tor, 2012).

4. PROPOSED TOR NETWORK ATTACKS

In addition to the many existing and perpetrated attacks that have already transpired against the Tor network, there are multiple theoretical attacks that have been described by the Tor developers. This section briefly describes some of those attacks as a means to discover Bridges and Relays and subsequently use them an attack launching point to (a) discover the topology of the Tor network; (b) break anonymity of the Tor user (profiling, timing, or traffic analysis); or (c) block access to the Tor network (Tor, 2011a; Tor, 2011b).

Many of these described attacks by the Tor developers and maintainers center on a nefarious party passively monitoring connections through an observation point within the Tor path. A more active version of this is to have the attacker actually participate in the Tor network as a Bridge, Relay, Guard, or even client. Both of these approaches allow the timing and traffic analysis attacks to be performed against a Tor user in such a manner that most Tor users would not recognize any analysis was occurring. Other approaches are more active in their style; including port scans, issuing malformed connection requests, and spoofing messages between Tor nodes. Combinations of these attacks may be performed to be more discriminating in actions against network users. That is, they may be concentrated in a more targeted manner to reduce collateral damage within the network.

Additional attacks have been proposed by Feamster and Dingledine (2004); Murdoch and Danezis (2005); and Dingledine and Murdoch (2009). Each of the scenarios described in these papers seek to perform traffic analysis and timing attacks. That is, attribution of a flow of traffic to a particular location within the network at a specific moment in time. The content of these flows were not exposed in these scenarios.

Other attacks over the Tor network have been performed that have specifically identified the actual source's IP address, hostname, time zone settings, and Internet browser type and version by exposing weaknesses in the applications that use Tor (Christensen, 2006a; Christensen, 2006b). Alternatively, an additional side-channel attack has been proposed by Shebaro (2012) in which a unique, identifiable, binary string is written to a client from a controlled or compromised destination; leaving the client potentially identifiable–eliminating privacy and anonymity upon investigation. That is, users have expected Tor to provide privacy and anonymity of their web browsing, yet their browsing habits allowed their anonymity to be compromised.

5. EVIDENTIARY ANALYSIS

Tor is not a panacea for all network related censorship issues. Tor can't solve complete network blackouts or shutdowns, as in Syria in late 2012 (Renesys, 2012). There must be connectivity in place for Tor to utilize. Without a network path, Tor is as powerless as any other connection tool; anonymizing or otherwise. With that being said, this section will analyze the potential for identifying inculpatory evidence within the Tor network for the purpose of some action being taken by law enforcement agencies (LEA).

Each of the perpetrated and proposed attacks has little forensic value at first pass. The value of these attacks, when successful, is in the transparent monitoring capabilities that a State or LEA may have that would subsequently allow direct correlation of inculpatory evidence to a specific network host. This approach from a state-sponsored attack is highly reliant on their ability to track or trace a particular data flow. This ability to correlate a flow to a particular host is dependent on the ability to actually capture those flows. This is directly related to the number of egress (and by extension, ingress) points to those networks. An analysis of the number of connection points into and out of a nation has been performed by Renesys in order to determine the likelihood of a complete blackout occurring similar to the Syria 2012 incident (Forbes, 2012). An alternate view of this data could be taken that a state may not wish to fully disconnect their citizens from the Internet. Instead, these connection points can become the capture, and correlation points for their monitoring systems; thereby, providing a State an evidence-gathering facility directly in the network path.

5.1 Analysis

To date, there are no known attempts to break the encryption algorithms used specifically attributable to the Tor network. That is, all known attempts at obstructing access to the Tor network, or identifying a user of the Tor network rely on attacking the architectural components or passive observation of traffic over the Tor network. As seen in Lewman (2009, 2012); Tor (2011a, 2011b); and Wilde (2012), it is possible to identify the role(s) a particular node is performing in the Tor network. When a connection is established, it is possible to determine which device is the client by identifying the TLS "Client Hello" portion of the encryption exchange. Likewise, Wilde showed that Bridges are identified in China and Kazakhstan through the TLS "Server Hello" messages within the encryption exchange. So then, we know directly that there are certain identifiable characteristics attributable to each role within the Tor architecture. This has been discussed further in Tor (2011a) as a code implementation and auditing issue.

It may be possible to identify traffic flows through the Tor network without knowing the location of Tor Relays, Bridges, or hosts. Tor specifies a cell size of 512 bytes. As such, it may be possible to examine flows for multiple consecutive packets around this size, as it may indicate a Tor flow is present. This pattern of packets will differ from that of a typical web transaction, where many consecutive packets are sized at the MTU and only the last packet will be smaller than the MTU. This is one example of the many different traffic and timing analysis attacks that could be utilized to identify Tor traffic. Yet, this analysis doesn't reveal the exact contents of the packets traversing the Tor network. Specific patterns of Tor cells may reveal the obfuscated protocol(s) and thereby types of traffic. However, no content is directly leaked out of the Tor network in any of the approaches. Anonymity is compromised via correlation of these flows to potential traffic patterns of known flows, or templates of flows.

5.2 Inculpatory Approaches

What, if any, evidence is available to prove participation with the Tor network? Many nations that are actively pursuing censorship of their population typically wish to identify those that are evading the systems in place that block or otherwise restrict access to the censored content. In identifying those persons, there is the potential for legal action, if there is inculpatory evidence. We have shown that many current nations that are blocking access to the Tor network are not yet actively pursuing the participants for legal action. Yet, that is not to say that they will not do so in the future.

It has been suggested that a LEA wishing to identify inculpatory evidence should host its own exit Relay within the Tor network and then actively perform traffic and timing analysis on the traffic as it exits and any response traffic that is generated (Schneier, 2008). Additionally, they could perform deep packet inspection on the traffic as it leaves the network, as the traffic will no longer be encrypted using the Tor network's layers of encryption. In this way, any unencrypted traffic would reveal the

actual payload sent by the anonymous client. The difficulty with this approach is that a Tor client will use a single channel for a single flow for no more than five minutes, by default. So, if the flow is longstanding, hosting an exit node will only account for a portion of that flow.

As seen in the prior section, there are many proposed attack methods intended to circumvent anonymity in Tor. As of this writing, there are no known successful attacks on the actual underlying encryption standards used within Tor. However, it has been known for some time that the vulnerabilities of encryption reside in potentially poor implementation of the encryption protocols and standards, not the protocols themselves (Schneier, 1998). As such, this paper will not spend any additional efforts to describe encryption circumvention attempts.

So then, based on the known executed attacks and potential attacks, what is the state of inculpatory discovery attempts? There are many denial of service attacks, as seen by the Chinese, Kazakh, and Iranian examples (Lewman, 2009; Lewman, 2012; Wilde, 2012). However, these do not provide any inculpatory material for an investigator as the client's identity has not been determined in any of these cases. That is, only access to the Tor Directory Service has been limited (Winter and Lindskog, 2012). Additional efforts are required to specifically identify the location and identity of the client in order to gain inculpatory evidence, depending on jurisdiction.

Other investigative efforts have been performed and identified that a direct attack of the Tor protocol and architecture is not the best means of identifying the users of the anonymizing service (Christensen, 2006a). Instead, attacks and manipulation of the application layer services being delivered over the Tor network are a much better means of determining Tor users (Christensen, 2006b; Fleischer, 2009).

Additionally, a LEA may wish to establish a mechanism for additional inculpatory evidence to elimination potentially indefensible scenarios. Exploiting the clients and services within this network could allow a LEA to place a unique "key" on a host under investigation, assuming they control part of the network, or a service requested by the client. Shebaro (2010) has proposed a mechanism by which a unique and recoverable bit pattern can be placed onto a host over the period of ~30 minutes with little possibility of timing analysis revealing its transmission.

Is Exculpatory evidence present or even feasible? As evidence may be present from other "regular" web browsing and file handling, specifics about exculpatory evidence are not explicit in this study. Understanding that there are differing thresholds for inculpatory evidence inclusion in any examination is critical (Loesing, Murdoch, Dingledine, 2010). In fact, there are hints of mere proposition of inculpatory evidence being sufficient to incarcerate or otherwise persecute individuals. (Hale, 2012).

6. CONCLUSIONS

The Tor network was devise and deployed as a low-latency anonymity-providing overlay network for TCP flows. It is implemented in a manner so that no node within the network can identify a complete flow through the network. A node is only capable of communicating with directly adjacent Tor nodes, even though they require potentially significant standard routing processes to connect those two adjacent nodes. This anonymity-providing network is now in use by those simply wishing to obfuscate their traffic from any potential observers as well as by those actively attempting to circumvent censorship. Because of this, those entities advocating censorship actively wish to maintain the censorship by eliminating the means used to evade it.

Traffic analysis of Tor nodes operating in a known infrastructure currently provides little inculpatory evidence without significant efforts to capture traffic at ingress and egress of multiple points of the network. Correlating this analysis to individual hosts on the Internet in a reliable and defensible manner poses a daunting challenge. Yet, this is, to date, the only executable attack against the Tor

network that attempt to determine "what" an individual flow *might* contain. The encryption methods used within Tor have not been circumvented. So, the raw payload of the traffic is not viewable.

There are potential attack points with the Tor architecture and the protocols implemented. Some, of which, have already been executed. However, these are not trivial exercises to perform or leverage these attacks. The more likely approach to breach the Tor network is to attempt to expose the identity of the users via exploitation of the weak (in terms of security) implementations of application-layer protocols and services that directly interact with those protocols.

Inculpatory evidence from the Tor network is difficult to obtain for most nation-states and their law enforcement agencies. Simple monitoring of existing Tor nodes will not directly reveal the types of traffic contained in the flows that passed, nor will any Tor node know the complete path that the flow is taking. Even if an agency were to implement a Tor Relay or Bridge and monitor connections to it, significant man-power and technical expertise is required to maintain and monitor these nodes for the purpose of ultimately reporting analyses on traffic behaviors. This is an implausible scenario for all but the largest of LEAs… typically those that are state-sponsored.

By looking at each individual component of the Tor network, the significance and amount of effort required to begin these attacks or analysis of the attacks can be seen. Beginning with inculpatory Analysis at a Directory–Since no user application data is sent to, passed through, or retrieved from the Directory there is little chance of inculpatory evidence. Also, since no single directory is authoritative for the state of all Relays in the Tor network there is no single point of attack, meaning also that there is no single point for investigation. As there is a voting process between all Tor Directories that establishes an agreed upon state of each Relay and publishes to all requesting Hosts as a network consensus document there could be the potential for learning some capability information of each Relay. But, this process is handled over encrypted links between each Directory and the Directory and Host. Additionally, there is no Directory discovery process for Hosts or Relays. Instead, the Directories are coded within the Tor toolset currently. So, all communications with a directory are encrypted. However, if this information is needed, it is not private information to the Tor network. A simple query to a Directory will result in the current network consensus document. This may have little inculpatory (or exculpatory) value unless it can be obtained during a specific monitoring period, as the document is updated on an ongoing basis.

Inculpatory Analysis at Entry node (Relay) – As shown in many of the insecure application attacks, the Onion Proxy (Host) can potentially be identified as the source of a flow of traffic. But, after the secure onion router circuit is established, traffic & timing analysis are the only viable methods, which is not significant in many cases, as attribution is significantly difficult.

Inculpatory Analysis at Intermediate relay – An intermediate (second) Relay in the onion routed path cannot identify source or destination without significant effort. This would require capabilities to monitor all relays in the onion route path in addition to performing traffic and timing analysis. This only provides the capabilities to perform attribution of ingress flows to egress flows with an uncertain level of probability. So then, only directly adjacent relays are known, and all communications along that channel segment are encrypted multiple times.

7. FUTURE WORK

While the forensic analysis that was briefly surveyed here shows little current inculpatory evidence is available by monitoring, and even participating in, the Tor network, there is significant interest in further evading detection of the Tor network. Some have proposed a censorship-detection add-on to the Tor network, while others are concentrating on continuing to push for the maturation of Tor. Extending the idea that Tor is a privacy- and anonymity-enhancing tool, and that those that block and

monitor Tor do so out of a desire to limit access to some resource, there are efforts required to define anti-forensic mechanisms within the Tor network.

Lastly, the mixes of Relays that are provide to the Tor client to establish circuits and connections are in need of more research. First, a thorough evaluation of the allocation policy and its potential to leak location or other identity information could be performed. Next, it has been suggested that even though multiple Relays are returned in a mix and are thought to be geographically diverse, they may still be part of the same administrative domain (BGP ASNs). There has been some effort to alleviate this, as no two Relays within the same /16 CIDR block will be selected in the same circuit. But, there has been no definitive proof that this is sufficient. So, it should be determined if this is indeed the case en masse, or as the outlier. To protect against such a situation, it is suggested that a module be added to the Tor Directory that documents the correlation between destination IP addresses and the BGP autonomous system in which those destinations reside. The ultimate purpose of this module will be to ensure that a mix provided by the Directory does not contain Relays from the same, or adjacent, ASes.

Further, there are additional architectural implementations of Tor that provide hidden services, where dual-party anonymity is possible. Additional efforts are needed to determine the state of forensics on those particular architectures and the services available over Tor network in this hidden nature. Identification of services offered and how they are processed through the Tor network with respect to the "standard" delivery of those services is of some interest.

REFERENCES

Arma. (2012). Iran blocks Tor; Tor releases same-day fix. Retrieved from https://blog.torproject.org/blog/iran-blocks-tor-tor-releases-same-day-fix on Dec. 7, 2012.

Christiansen, A. (2006a). *Peeling the onion: Unmasking Tor users*. FortConsult's Security Research Team.

Christiansen, A. (2006b). *Practical onion hacking: Finding the real address of Tor clients*. FortConsult's Security Research Team.

Dingledine, R. (n.d.). *Tor: An anonymous Internet communication system*.

Dingledine, R., Mathewson, N., and Syverson, P. (2004). Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, August 2004.

Dingledine, R., and Mathewson, N. (n.d.). Tor Protocol Specification. Retrieved from http://gitweb.torproject.org/torspec.git/blob/HEAD:/tor-spec.txt

Dingledine, R., and Murdoch, S.J. (2009). Performance improvements for Tor or, Why is Tor slow and what we're going to do about it. *DEFCON 17*.

Feamster, N., and Dingledine, R. (2004). Location diversity in anonymous networks. *Proceedings of WPES 2004 (ACM)*, Washington D.C.

Fleischer, G. (2009). Attacking Tor and the application layer. *DEFCON 17*.

Goldschlage, D. M., Reed, M. G., and Syverson, P.F. (1996). Hiding routing information. *Workshop on Information Hiding*, Cambridge, UK. May 1996.

Ho, S. (2009). F.O.E: Feed over Email–A proxy-less RSS Reader. *DEFCON 17*.

Le Blond, S., Manils, P., Chaabane, A., Kaafar, M., Castelluccia, C., Legout, A., and Dabbous, W. (2011). One bad apple spoils the bunch: Exploiting P2P applications to trace and profile Tor users. *LEET 2011: 4th USENIX Workshop on Large-Scale Exploits and Emergent Threats*. Retrieved from http://static.usenix.org/event/leet11/tech/full_papers/LeBlond.pdf on March 15, 2013.

Lewman, A. (2009). Tor partially blocked in China. Retrieved from https://blog.torproject.org/blog/tor-partially-blocked-china on Dec. 7, 2012.

Lewman, A. (2012). Updates on Kazakhstan Internet censorship. Retrieved from https://blog.torproject.org/blog/updates-kazakhstan-internet-censorship on Dec. 7, 2012.

Loesing, K., Murdoch, S. J., and Dingledine, R. (2010). A case study on measuring statistical data in the Tor anonymity network. In R. Sion et al. (Eds), *FC 2010 Workshops, LNCS* (pp. 203-215). Springer-Verlag Berlin.

Murdoch, S. J., and Danezis, G. (2005). Low-cost traffic analysis of Tor. *The 2005 IEEE Symposium on Security and Privacy*, May 8–11 2005, Oakland, California, USA.

Pfitzmann, A., and Kohntopp, M. (2001). Anonymity, unobservability and pseudoymity – A proposal for terminology. In Hannes Federath (Ed.), *Designing Privacy Enhancing Technologies, Lecture Notes in Computer Science, LNCS 2009* (pp. 1-9). Springer-Verlag.

Rezale, A. (2012). Iran shut down Gmail, Google, Yahoo and sites using HTTPS Protocol. *KAbir News*. Retrieved from http://kabirnews.com/iran-shut-down-gmail-google-yahoo-and-sites-using-https-protocol/202/

RFC 791. (1981). Internet Protocol: DARPA Internet program protocol specification. Retrieved from http://www.ietf.org/rfc/rfc791.txt on Dec. 7, 2012.

Schneier, B. (1998). Security pitfalls in cryptography. Retrieved from http://www.schneier.com/essay-028.html on Dec. 7, 2012.

Shebaro, B. (2012). *Privacy-Preserving techniques for computer and network forensics* (Doctoral Dissertation). Available from ProQuest (UMI No. 3517591).

Tor Project. (2012). Retrieved from http://www.torproject.com/?asdf on Dec. 7, 2012.

Tor. (2011a). Research problems: Ten ways to discover Tor bridges. Retrieved from https://blog.torproject.org/blog/research-problems-ten-ways-discover-tor-bridges on Dec. 7, 2012.

Tor. (2011b). Research problem: Five ways to test bridge reachability. Retrieved from https://blog.torproject.org/blog/research-problem-five-ways-test-bridge-reachability on Dec. 7, 2012.

Tor. (2012). A tale of new censors - Vodafone UK, T-Mobile UK, O2 UK, and T-Mobile USA. Retrieved from https://blog.torproject.org/blog/tale-new-censors-vodafone-uk-t-mobile-uk-o2-uk-and-t-mobile-usa on Dec. 7, 2012.

York, J. C. (2012a). Palestinian authority found to block critical news sites. Retrieved from https://www.eff.org/deeplinks/2012/04/palestinian-authority-found-block-critical-news-sites on Dec. 7, 2012.

York, J. C. (2012b). This Week in Internet censorship: Crackdowns in Nigeria, Tajikistan and Morocco, immolation in Vietnam. Retrieved from https://www.eff.org/deeplinks/2012/07/week-internet-censorship-nigeria-tajikistan-morocco on Dec. 7, 2012.

Wilde, T. (2012). Knock knock knockin' on Bridges' Doors. Retrieved from https://blog.torproject.org/blog/knock-knock-knockin-bridges-doors on Dec. 7, 2012.

Winter, P., and Lindskog, S. (2012). How the great firewall of China is Blocking Tor. *FOCI 2012: 2nd USENIX Workshop on Free and Open Communications on the Internet*. Retrieved from http://www.cs.kau.se/philwint/foci2012

Zwicky, E. D., Cooper, S., and Chapman, D. B. (2000). *Building Internet firewalls*. 2ⁿᵈ ed. O'Reilly Media.