Shining Light in Dark Places: Understanding the Tor Network

Damon McCoy[1], Kevin Bauer[1], Dirk Grunwald[1],
Tadayoshi Kohno[2], and Douglas Sicker[1]

[1] Department of Computer Science,
University of Colorado, Boulder, CO 80309-0430, USA
{mccoyd,bauerk,grunwald,sicker}@colorado.edu
[2] Department of Computer Science and Engineering,
University of Washington, Seattle, WA 98195-2969, USA
yoshi@cs.washington.edu

Abstract. To date, there has yet to be a study that characterizes the usage of a real deployed anonymity service. We present observations and analysis obtained by participating in the Tor network. Our primary goals are to better understand Tor as it is deployed and through this understanding, propose improvements. In particular, we are interested in answering the following questions: (1) How is Tor being used? (2) How is Tor being *mis*-used? (3) Who is using Tor?

To sample the results, we show that web traffic makes up the majority of the connections and bandwidth, but non-interactive protocols consume a disproportionately large amount of bandwidth when compared to interactive protocols. We provide a survey of how Tor is being misused, both by clients and by Tor router operators. In particular, we develop a method for detecting exit router logging (in certain cases). Finally, we present evidence that Tor is used throughout the world, but router participation is limited to only a few countries.

1 Introduction

Tor is a popular privacy enhancing system that is designed to protect the privacy of Internet users from traffic analysis attacks launched by a non-global adversary [1]. Because Tor provides an anonymity service on top of TCP while maintaining relatively low latency and high throughput, it is ideal for interactive applications such as web browsing, file sharing, and instant messaging. Since its initial development, researchers have analyzed the system's performance [2] and security properties [3,4,5,6,7]. However, there has yet to be a study aimed at understanding how a popular deployed privacy enhancing system is used in practice. In this work, we utilize observations made by running a Tor router to answer the following questions:

How is Tor being used?. We analyze application layer header data relayed through our router to determine the protocol distribution in the anonymous

N. Borisov and I. Goldberg (Eds.): PETS 2008, LNCS 5134, pp. 63–76, 2008.

network. Our results show the types of applications currently used over Tor, a substantial amount of which is non-interactive traffic. We discover that web traffic makes up the vast majority of the connections through Tor, but BitTorrent traffic consumes a disproportionately large amount of the network's bandwidth. Perhaps surprisingly, protocols that transmit passwords in plain-text are fairly common, and we propose simple techniques that attempt to protect users from unknowingly disclosing such sensitive information over Tor.

How is Tor being *mis*-used?. To explore how Tor is currently being misused, we examine both malicious router and client behaviors. Since insecure protocols are common in Tor, there is a potential for a malicious router to gather passwords by logging exit traffic. To understand this threat, we develop a method to detect when exit routers are logging traffic, under certain conditions. Using this method, we did, in fact, catch an exit router capturing POP3 traffic (a popular plain-text e-mail protocol) for the purpose of compromising accounts.

Running a router with the default exit policy provides insight into the variety of malicious activities that are tunneled trough Tor. For instance, hacking attempts, allegations of copyright infringement, and bot network control channels are fairly common forms of malicious traffic that can be observed through Tor.

Who is using Tor?. In order to understand who uses Tor, we present the geopolitical distribution of the clients that were observed. Germany, China, and the United States appear to use Tor the most, but clients from 126 different countries were observed, which demonstrates Tor's global appeal. In addition, we provide a geopolitical breakdown of who participates in Tor as a router. Most Tor routers are from Germany and the United States, but Germany alone contributes nearly half of the network's total bandwidth. This indicates that implementing location diversity in Tor's routing mechanism is not possible with the current distribution of router resources.

Outline. The remainder of this paper is organized as follows: In Section 2, we provide a brief overview of Tor and Section 3 describes our data collection methodology. In Section 4, we explore how Tor is used, and present the observed exit traffic protocol distribution. In Section 5, we discuss how Tor is commonly abused by routers, and describe a new technique for detecting routers that maliciously log exit traffic. Section 6 describes our first-hand experiences with misbehaving clients. Section 7 gives the geopolitical distributions of clients and routers. Finally, concluding remarks are given in Section 8.

2 Tor Network

Tor's system architecture attempts to provide a high degree of anonymity and strict performance standards simultaneously [1]. At present, Tor provides an anonymity layer for TCP by carefully constructing a three-hop path (by default), or *circuit*, through the network of *Tor routers* using a layered encryption strategy similar to *onion routing* [8]. Routing information is distributed by a set of authoritative directory servers. In general, all of a particular client's TCP connections are tunneled through a single circuit, which rotates over time. There are typically three hops in a circuit; the first node in the circuit is known as the *entrance Tor router*, the middle node is called the *middle Tor router*, and the final hop in the circuit is referred to as the *exit Tor router*. It is important to note that only the entrance router can directly observe the originator of a particular request through the Tor network. Also, only the exit node can directly examine the decrypted payload and learn the final destination server. It is infeasible for a single Tor router to infer the identities of both the initiating client and the destination server. To achieve its low-latency objective, Tor does not explicitly re-order or delay packets within the network.

3 Data Collection Methodology

To better understand real world Tor usage, we set up a Tor router on a 1 Gb/s network link.[1] This router joined the currently deployed network during December 2007 and January 2008. This configuration allowed us to record a large amount of Tor traffic in short periods of time. While running, our node was consistently among the top 5% of routers in terms of bandwidth of the roughly 1,500 routers flagged as `Running` by the directory servers at any single point in time.

We understand that there are serious privacy concerns that must be addressed when collecting statistics from an anonymous network [9]. Tor is designed to resist traffic analysis from any single Tor router [1]; thus, the information we log — which includes *at most* 20 bytes of application-level data — cannot be used to link a sender with a receiver, in most cases. We considered the privacy implications carefully when choosing what information to log and what was too sensitive to store. In the end, we chose to log information from two sources: First, we altered the Tor router to log information about circuits that were established though our node and cells routed through our node. Second, we logged only enough data to capture up to the application-level protocol headers from the exit traffic that was relayed through our node.

In order to maximize the number of entry and exit connections that our router observed, it was necessary to run the router *twice*, with two distinct exit policies:[2] (1) Running with an *open exit policy* (the default exit policy[3]) enabled our

[1] Our router used `Tor` software version 0.1.2.18.

[2] Due to the relatively limited exit bandwidth that exists within Tor, when we ran the default exit policy, our node was chosen as the exit router most frequently on established circuits. As a result, in order to observe a large number of clients, it became necessary to collect data a second time with a completely restricted exit policy so that we would not be an exit router.

[3] The default exit policy blocks ports commonly associated with SMTP, peer-to-peer file sharing protocols, and ports with a high security risk.

router to observe numerous exit connections, and (2) *Prohibiting all exit traffic* allowed the router to observe a large number of clients.

Entrance/Middle Traffic Logging. To collect data regarding Tor clients, we ran our router with a completely restricted exit policy (all exit traffic was blocked). We ran our Tor router in this configuration for 15 days from January 15–30, 2008. The router was compiled with minor modifications to support additional logging. Specifically, for every cell routed through our node, the time that it was received, the previous hop's IP address and TCP port number, the next hop's IP address and TCP port number, and the circuit identifier associated with the cell is logged.

Exit Traffic Logging. To collect data regarding traffic exiting the Tor network, we ran the Tor router for four days from December 15–19, 2007 with the default exit policy. For routers that allow exit traffic, the default policy is the most common. During this time, our router relayed approximately 709 GB of TCP traffic exiting the Tor network.

In order to gather statistics about traffic leaving the network, we ran `tcpdump` on the same physical machine as our Tor router. Tcpdump was configured to capture only the first 150 bytes of a packet using the "snap length" option (`-s`). This limit was selected so that we could capture up to the application-level headers for protocol identification purposes. At most, we captured 96 bytes of application header data, since an Ethernet frame is 14 bytes long, an IP header is 20 bytes long, and a TCP header with no options is 20 bytes long. We used `ethereal` [10], another tool for protocol analysis and stateful packet inspection, in order to identify application-layer protocols. As a post-processing step, we filtered out packets with a source or destination IP address of any active router published during our collection period. This left only exit traffic.

4 Protocol Distribution

As part of this study, we observe and analyze the application-level protocols that exit our Tor node. We show in Table 1 that interactive protocols like HTTP make up the majority of the traffic, but non-interactive traffic consumes a disproportionate amount of the network's bandwidth. Finally, the data indicates that insecure protocols, such as those that transmit login credentials in plain-text, are used over Tor.

4.1 Interactive vs. Non-interactive Web Traffic

While HTTP traffic comprises an overwhelming majority of the connections observed, it is unclear whether this traffic is interactive web browsing or non-interactive downloading. In order to determine how much of the web traffic is non-interactive, we counted the number of HTTP connections that transferred over 1 MB of data. Only 3.5% of the connections observed were bulk transfers. The vast majority of web traffic is interactive.

Table 1. Exit traffic protocol distribution by number of TCP connections, size, and number of unique destination hosts

Protocol	**Connections**	**Bytes**	**Destinations**
HTTP	12,160,437 (92.45%)	411 GB (57.97%)	173,701 (46.01%)
SSL	534,666 (4.06%)	11 GB (1.55%)	7,247 (1.91%)
BitTorrent	438,395 (3.33%)	285 GB (40.20%)	194,675 (51.58%)
Instant Messaging	10,506 (0.08%)	735 MB (0.10%)	880 (0.23%)
E-Mail	7,611 (0.06%)	291 MB (0.04%)	389 (0.10%)
FTP	1,338 (0.01%)	792 MB (0.11%)	395 (0.10%)
Telnet	1,045 (0.01%)	110 MB (0.02%)	162 (0.04%)
Total	13,154,115	709 GB	377,449

4.2 Is Non-interactive Traffic Hurting Performance?

The designers of the Tor network have placed a great deal of emphasis on achieving low latency and reasonable throughput in order to allow interactive applications, such as web browsing, to take place within the network [1]. However, the most significant difference between viewing the protocol breakdown measured by the number of bytes in contrast to the number of TCP connections is that while HTTP accounted for an overwhelming majority of TCP connections, the BitTorrent protocol uses a disproportionately high amount of bandwidth.[4] This is not shocking, since BitTorrent is a peer-to-peer (P2P) protocol used to download large files.

Since the number of TCP connections shows that the majority of connections are HTTP requests, one might be led to believe that most clients are using the network as an anonymous HTTP proxy. However, the few clients that do use the network for P2P applications such as BitTorrent consume a significant amount of bandwidth. The designers of the network consider P2P traffic harmful, not for ethical or legal reasons, but simply because it makes the network less useful to those for whom it was designed. In an attempt to prevent the use of P2P programs within the network, the default exit policy blocks the standard file sharing TCP ports. But clearly, our observations show that port-based blocking strategies are easy to evade, as these protocols can be run on non-standard ports.

4.3 Insecure Protocols

Another surprising observation from the protocol statistics is that insecure protocols, or those that transmit login credentials in plain-text, are fairly common. While comprising a relatively low percentage of the total exit traffic observed, protocols such as POP, IMAP, Telnet, and FTP are particularly dangerous due

[4] Recall that our router's default exit policy does not favor any particular type of traffic. So the likelihood of observing any particular protocol is proportional to the usage of that protocol within the network and the number of other nodes supporting the default or a similar exit policy.

to the ease at which an eavesdropping exit router can capture identifying information (i.e., user names and passwords). For example, during our observations, we saw 389 unique e-mail servers, which indicates that there were at least 389 clients using insecure e-mail protocols. In fact, only 7,247 total destination servers providing SSL/TLS were observed.

The ability to observe a significant number of user names and passwords is potentially devastating, but it gets worse: Tor multiplexes several TCP connections over the same circuit. Having observed identifying information, a malicious exit router can trace all traffic on the same circuit back to the client whose identifying information had been observed on that circuit. For instance, suppose that a client initiates both an SSL connection and an AIM connection at the same time. Since both connections use the same circuit (and consequently exit at the same router), the SSL connection can be easily associated with the client's identity leaked by the AIM protocol. Thus, tunneling insecure protocols over Tor presents a significant risk to the initiating client's anonymity.

To address this threat, a reasonable countermeasure is for Tor to explicitly block protocols such as POP, IMAP, Telnet, and FTP[5] using a simple port-based blocking strategy at the client's local socks proxy.[6] In response to these observations, Tor now supports two configuration options to (1) warn the user about the dangers of using Telnet, POP2/3, and IMAP over Tor, and (2) block these insecure protocols using a port-based strategy [11].

However, this same type of information leakage is certainly possible over HTTP, for instance, so additional effort must also be focused on enhancing Tor's HTTP proxy to mitigate the amount of sensitive information that can be exchanged over insecure HTTP. For instance, a rule-based system could be designed to filter common websites with insecure logins.

Finally, protocols that commonly leak identifying information should not be multiplexed over the same circuit with other non-identifying traffic. For example, HTTP and instant messaging protocols should use separate and dedicated circuits so that any identifying information disclosed through these protocols is not linked with other circuits transporting more secure protocols.

5 Malicious Router Behavior

Given the relatively large amount of insecure traffic that can be observed through Tor, there is great incentive for malicious parties to attempt to log sensitive information as it exits the network. In fact, others have used Tor to collect a large number of user names and passwords, some of which provided access to the computer systems of embassies and large corporations [12].

[5] Anonymous FTP may account for a significant portion of FTP exit traffic and does not reveal any information about the initiating client. Therefore, blocking FTP may be unnecessary.

[6] Port-based blocking is easy to evade, but it would protect naive users from mistakenly disclosing their sensitive information.

In addition to capturing sensitive exit traffic, a Tor router can modify the decrypted contents of a message entering or leaving the network. Indeed, in the past, routers have been caught modifying traffic (i.e., injecting advertisements or performing man-in-the-middle attacks) in transit, and techniques have been developed to detect this behavior [13].

We present a simple method for detecting exit router logging under certain conditions. We suspect — and confirm this suspicion using our logging detection technique — that insecure protocols are targeted for the specific purpose of capturing user names and passwords.

5.1 Detection Methodology

At a high level, the malicious exit router logging detection technique relies upon the assumption that the exit router is running a packet sniffer on its local network. Since packet sniffers such as `tcpdump` are often configured to perform reverse DNS queries on the IP addresses that they observe, if one controls the authoritative DNS server for a specific set of IP addresses, it is possible to trace reverse DNS queries back to the exit node that issued the query.

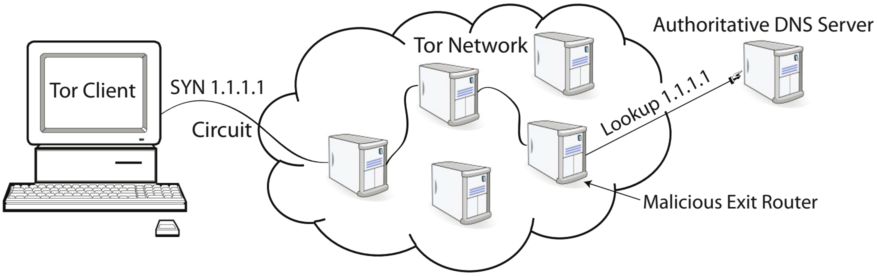


Fig. 1. Malicious exit router logging detection technique

More specifically, the detection method works as follows:

1. We run an authoritative domain name server (DNS) that maps domain names to a vacant block of IP addresses that we control.
2. Using a Tor client, a circuit is established using each individual exit router.
3. Having established a circuit, a `SYN` ping is sent to one of the IP addresses for which we provide domain name resolution.

This procedure (shown in Figure 1) is repeated for each exit router. Since the IP address does not actually exist, then it is very unlikely that there will be any transient reverse DNS queries. However, if one of the exit routers we used is logging this traffic, they may perform a reverse DNS look-up of the IP address that was contacted. In particular, we made an effort to direct the `SYN` ping at ports where insecure protocols typically run (ports 21, 23, 110, and 143).

5.2 Results

Using the procedure described above, over the course of only one day, we found one exit router that issued a reverse DNS query immediately after transporting our client's traffic. Upon further inspection, by `SYN` ping scanning all low ports (1-1024), we found that only port 110 triggered the reverse DNS query. Thus, this router only logged traffic on this port, which is the default port for POP3, a plain-text e-mail protocol. We suspect that this port was targeted for the specific purpose of capturing user names and passwords.

Further improvements on this logging detection could be made by using a honeypot approach and sending unique user name and password pairs through each exit router. The honeypot could detect any login attempts that may occur. This method would find the most malicious variety of exit router logging. In fact, upon detecting the logging exit router (using the method described above), we also used this honeypot technique and observed failed login attempts from the malicious IP address shortly after observing the logging.

These results reinforce the need to mitigate the use of protocols that provide login credentials in plain-text over Tor. Given the ease at which insecure protocols can be captured and the relative ease at which they could be blocked, it is a reasonable solution to block their default ports.

5.3 Discussion

This approach to detecting exit router logging has limitations. First, it can only trace the reverse DNS query back to the exit router's DNS server, not to the router itself. To complicate matters more, there exist free domain name resolution services (such as OpenDNS [14]) that provide somewhat anonymous name resolution for any host on the Internet. If one assumes that the exit router is logging and performing reverse DNS queries in real-time, then it is easy to correlate reverse DNS queries with exit routers using timing information.

If reverse DNS is *not* performed in real-time, then more sophisticated techniques for finding the malicious exit router are required. For instance, if one controls the domain name resolution for several IP addresses, then it is possible to embed a unique pattern in the order of the `SYN` pings to different IPs through each exit router. This order will be preserved in the exit router's queries and can be used to determine the exit router that logged the traffic. Here we can leverage many of the same principles as explored in [15,16].

The detection method presented makes the key assumption that the logging process will trigger reverse-DNS queries. However, this is not always the case. For example, exit routers that transport traffic at high bandwidth cannot feasibly perform reverse DNS queries in real-time. Also, this technique can be evaded simply by not performing reverse DNS when logging.

6 Misbehaving Clients

While Tor provides an invaluable service to protecting online privacy, over the course of operating a Tor router with the default exit policy, we learned about a wide variety of malicious client behavior. Since we are forwarding traffic on behalf of Tor users, our router's IP address appears to be the source of sometimes malicious traffic. The large amount of exit bandwidth that we provided caused us to receive a large number of complaints ranging from DMCA §512 notices related to allegations of copyright infringement, reported hacking attempts, IRC bot network controls, and web page defacement. However, an enormous amount of malicious client activity was likely unreported.

As a consequence of this malicious client behavior, it becomes more difficult to operate exit routers. For instance, our institution's administration requested that we stop running our node shortly after the data for this paper was collected. Similar accounts of administrative and law enforcement attempts to prevent Tor use are becoming more common as Tor becomes more popular to the masses [17]. The Electronic Frontier Foundation (EFF), a group that works to protect online rights, has provided template letters [18] and offered to provide assistance [19] to Tor router operators that have received DMCA take-down notices.

One solution to our problems could have been to change our router's exit policy to reject all exit traffic, or specific ports (such as port 80) that generate a large portion of the complaints. However, this is not practical, since Tor requires a certain amount of exit bandwidth to function correctly. Another solution is to provide a mechanism for anonymous IP address blocking, such as Nymble [20]. Our first-hand observations with misbehaving clients reinforces the need to further study anonymous IP address blocking mechanisms.

7 Geopolitical Client and Router Distributions

As part of this study, we investigate where Tor clients and routers are located geopolitically. Recall that a client's IP address is visible to a router when that router is used as the entrance node on the client's circuit through the Tor network. In the current Tor implementation, only particular routers, called *entry guards*, may be used for the first hop of a client's circuit. A router is labeled as an entry guard by the authoritative directory servers. All Tor router IP addresses are maintained by the directory servers, and we keep track of the router IP addresses by simply polling the directory servers periodically.

In order to map an IP address to its corresponding country of origin, we query the authoritative bodies responsible for assigning IP blocks to individual countries [21,22,23,24,25]. In order to determine the geopolitical distribution of Tor usage throughout the world, we aggregate IP addresses by country, and present the client and router location distributions observed during the January 2008 data collection period.

7.1 Observations

In this section, we present our observations regarding the client and router location distributions.

Table 2. Geopolitical client distributions, router distributions, and the ratio of Tor users relative to Internet users

Client Distribution		**Router Distribution**	
Country	*Total*	*Country*	*Total*
Germany	2,304	Germany	374
China	988	United States	326
United States	864	France	69
Italy	254	China	40
Turkey	221	Italy	36
United Kingdom	170	Netherlands	35
Japan	155	Sweden	35
France	150	Finland	25
Russia	146	Austria	24
Brazil	134	United Kingdom	24

Relative Tor Usage	
Country	*Ratio*
Germany	7.73
Turkey	2.47
Italy	1.37
Russia	0.89
China	0.84
France	0.77
United Kingdom	0.75
United States	0.62
Brazil	0.56
Japan	0.32

Client Distribution. During a one day period when our Tor router was marked as an entry guard by the authoritative directory servers, it observed 7,571 unique clients[7] As depicted in Table 2, the vast majority of clients originated in Germany, with China and the United States providing the next largest number of clients. Perhaps the most interesting observation about the client distribution is that Tor has a global user base. While most of the clients are from three countries, during the course of the entire 15 day observation period, clients were observed from 126 countries around the world, many of which have well-known policies of Internet censorship.

To put these raw geopolitical client distributions into perspective, Table 2 includes a ratio of the percentage of Tor users to the percentage of Internet users by country, using data on the distribution of broadband Internet users by country [26]. These percentages were computed by dividing the total number of Tor clients located in each country by the total number of Tor clients we observed, which provides the percentage of Tor users located in each country. For example, the relative Tor usage for Germany is computed as follows: The percentage of the total Internet users who are from Germany is 3.9% and according to our client observations, Germany makes up 2,304 of the 7,571 total Tor clients, which is 30.4%. Thus, the ratio of Tor users to Internet users in Germany is 7.73.

These ratios show that Tor is disproportionately popular in Germany, Turkey, and Italy with respect the the number of broadband Internet users located in these countries. It is unclear why there is such a large scale adoption of Tor in these specific countries, relative to Tor usage in other countries. An investigation of the possible technological, sociological, and political factors in these countries that are causing this might be an enlightening area of research.

Examining the number of clients that utilized our router as their entry router when *it was not marked as an entry guard* provides insight into the approximate

[7] We assume that each unique IP address is a unique client. However, dynamic IP addresses or network address translators (NATs) may be used in some places.

number of clients that are using a significantly old version of the Tor client software. Specifically, this indicates that these clients are using a version *before* entry guards were introduced in `Tor` version 0.1.1.20 (May 2006). Over four days, only 206 clients were observed to be using Tor software that is older than this version.

Incidentally, entry guards were added to prevent routers from profiling clients, and indeed the reliance on entry guards prevented us from profiling a large number of clients beyond what we describe above. Before entry guards were widely adopted, a strong diurnal usage pattern had been observed [27]. Since entry guards are now widely adopted, utilizing multiple entry guard perspectives gives a larger snapshot of the clients' locations and usage patterns. We informally compared our geopolitical client distribution to that which was observed from other high bandwidth entry guard routers. The distribution was consistent across each entry guard. However, we attempted to observe the current client usage patterns, but this required a more global perspective than we were able to obtain.

Tor Router Distribution. During our data collection, we monitored the authoritative directory servers to determine the total number and geopolitical distribution of Tor routers. Over the course of 7 days, we took hourly snapshots of the authoritative directory servers, noting each router's IP address and bandwidth advertisements. During this time, on average 1,188 Tor routers were observed in each snapshot. As shown in Table 2, Germany and the United States together contribute nearly 59% of the running routers. However, in terms of total bandwidth, as depicted in Figure 2, Germany provides 45% of the bandwidth and the United States only provides 23% of the bandwidth.

It has been suggested that location diversity is a desirable characteristic of a privacy enhancing system [28]. However, given the current bandwidth distribution, location diversity while maintaining adequate load balancing of traffic is difficult to guarantee. It is currently possible to build circuits with at least one router from Germany and the remaining routers from other countries. However, if a location-aware routing mechanism mandated that a user's traffic should exit in a specific country, such as the Netherlands, then it is necessary to ensure that there is sufficient exit bandwidth in that country. Incentive programs to encourage volunteers to run routers in under-represented countries should be investigated. In addition, mitigating malicious client behavior (as noted in Section 6) can consequently attract more Tor routers.

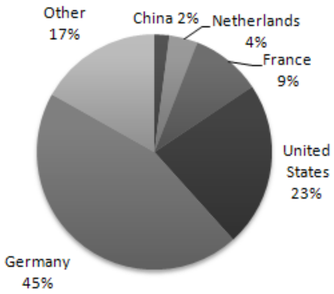
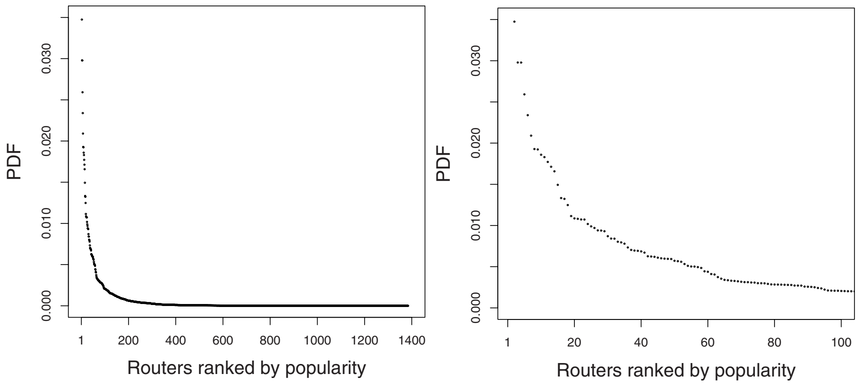


Fig. 2. Distribution of Tor router bandwidth around the world



(a) PDF of all routers. (b) PDF of the top 100 routers.

Fig. 3. PDFs of Tor's traffic distribution over its routers during a one hour snapshot

7.2 Modeling Router Utilization

Understanding the distribution with which different routers are utilized on circuits can provide valuable insights regarding the system's vulnerability to traffic analysis. In addition, a probability distribution can be used to build more realistic analytical models and simulations.

By counting the number of times that each router appears on a circuit with our router, we provide probability density functions (PDFs) to model the probability of each router forwarding a particular packet (shown in Figure 3). In a one hour snapshot during the January data collection period, the top 2% of all routers transported about 50% of traffic from the perspective of our router. Within this top 2%, 14 routers are hosted in Germany, 6 are hosted in the United States, 4 are in France, and Switzerland, the Netherlands, and Finland each host a single router. These numbers are consistent with the bandwidth distributions given in Figure 2, and further highlight the difficulty of providing strict location diversity in Tor's routing mechanism. The PDF curve drops sharply; the bottom 75% of the routers together transported about 2% of the total traffic. The most traffic that any single router transported was 4.1% of the total traffic. This indicates that the vast majority of Tor traffic is handled by a very small set of routers. Consequently, if an adversary is able to control a set of the highest performing routers, then its ability to conduct traffic analysis increases dramatically. Finally, the PDFs calculated from our router's observations are very similar to the router distribution based on routers' bandwidth advertisements, as reported by Tor's directory servers.

8 Conclusion

This study is aimed at understanding Tor usage. In particular, we provided observations that help understand how Tor is being used, how Tor is being mis-used, and who participates in the network as clients and routers. Through our observations, we have made several suggestions to improve Tor's current design and implementation. First, in response to the fairly large amount of insecure protocol traffic, we proposed that Tor provide a mechanism to block the ports associated with protocols such as POP3, IMAP, and Telnet. Given the ease at which an eavesdropping exit router can log sensitive user information (such as user names and passwords), we developed a method for detecting malicious logging exit routers, and provided evidence that there are such routers that specifically log insecure protocol exit traffic. As a final avenue of study, we show the disparity in geopolitical diversity between Tor clients and routers, and argue that location diversity is currently impossible to guarantee unless steps are taken to attract a more diverse set of routers.

Due to its popularity, Tor provides insight into the challenges of deploying a real anonymity service, and our hope is that this work will encourage additional research aimed at (1) providing tools to enforce accountability while preserving strong anonymity properties, (2) protecting users from unknowingly disclosing sensitive/identifying information, and (3) fostering participation from a highly diverse set of routers.

Acknowledgements. We thank Roger Dingledine, Parisa Tabriz, and the anonymous PETS 2008 reviewers whose comments greatly improved the quality of this paper. This research was partially supported by the National Science Foundation under grant ITR-0430593.

References

1. Dingledine, R., Mathewson, N., Syverson, P.: Tor: The second-generation onion router. In: Proceedings of the 13th USENIX Security Symposium (August 2004)
2. Wendolsky, R., Herrmann, D., Federrath, H.: Performance comparison of low-latency anonymisation services from a user perspective. In: Borisov, N., Golle, P. (eds.) PET 2007. Springer, Heidelberg (2007)
3. Goldberg, I.: On the security of the Tor authentication protocol. In: Danezis, G., Golle, P. (eds.) PET 2006. LNCS, vol. 4258. Springer, Heidelberg (2006)
4. Murdoch, S.J.: Hot or not: Revealing hidden services by their clock skew. In: 13th ACM Conference on Computer and Communications Security (CCS 2006), Alexandria, VA (November 2006)
5. Murdoch, S.J., Danezis, G.: Low-cost traffic analysis of Tor. In: Proceedings of the 2005 IEEE Symposium on Security and Privacy. IEEE Computer Society Press, Los Alamitos (2005)
6. Øverlier, L., Syverson, P.: Locating hidden servers. In: Proceedings of the 2006 IEEE Symposium on Security and Privacy. IEEE Computer Society Press, Los Alamitos (2006)
7. Bauer, K., McCoy, D., Grunwald, D., Kohno, T., Sicker, D.: Low-resource routing attacks against Tor. In: Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2007), Washington, DC, USA (October 2007)
8. Goldschlag, D.M., Reed, M.G., Syverson, P.F.: Hiding routing information. In: Anderson, R. (ed.) IH 1996. LNCS, vol. 1174. Springer, Heidelberg (1996)

9. Sicker, D.C., Ohm, P., Grunwald, D.: Legal issues surrounding monitoring during network research. In: IMC 2007: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement. ACM Press, New York (2007)
10. Ethereal: A network protocol analyzer, `http://www.ethereal.com`
11. Bauer, K., McCoy, D.: Block insecure protocols by default (January 2008), `https://tor-svn.freehaven.net/svn/tor/trunk/doc/spec/proposals/129-reject-plaintext-ports.txt`
12. Zetter, K.: Tor researcher who exposed embassy e-mail passwords gets raided by Swedish FBI and CIA (November 2007), `http://blog.wired.com/27bstroke6/2007/11/swedish-researc.html`
13. Perry, M.: Torflow, `https://www.torproject.org/svn/torflow/README`
14. OpenDNS, `http://www.opendns.com`
15. Bethencourt, J., Franklin, J., Vernon, M.: Mapping Internet sensors with probe response attacks. In: Proceedings of the 14th conference on USENIX Security Symposium, Baltimore, MD. USENIX Association (2005)
16. Shinoda, Y., Ikai, K., Itoh, M.: Vulnerabilities of passive Internet threat monitors. In: Proceedings of the 14th conference on USENIX Security Symposium, Baltimore, MD. USENIX Association (2005)
17. Cesarini, P.: Caught in the Network. In: The Chronicle of Higher Education, Washington, D.C, vol. 53 (February 2007)
18. Tor: Response template for Tor node maintainer to ISP, `http://www.torproject.org/eff/tor-dmca-response.html`
19. Dingledine, R.: EFF is looking for Tor DMCA test case volunteers, `http://archives.seul.org/or/talk/Oct-2005/msg00208.html`
20. Johnson, P.C., Kapadia, A., Tsang, P.P., Smith, S.W.: Nymble: Anonymous IP-address blocking. In: Borisov, N., Golle, P. (eds.) PET 2007. Springer, Heidelberg (2007)
21. American Registry for Internet Numbers, `http://www.arin.net/index.shtml`
22. Asia Pacific Network Information Centre, `http://www.apnic.net`
23. Latin American & Caribbean Internet Addresses Registry, `http://lacnic.net/en`
24. Ripe Network Coordination Centre, `http://www.ripe.net`
25. African Network Information Centre, `http://www.afrinic.net`
26. Inernet World Stats, `http://www.internetworldstats.com`
27. McCoy, D., Bauer, K., Grunwald, D., Tabriz, P., Sicker, D.: Shining light in dark places: A study of anonymous network usage. University of Colorado Technical Report CU-CS-1032-07 (2007)
28. Feamster, N., Dingledine, R.: Location diversity in anonymity networks. In: Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2004), Washington, DC, USA (October 2004)