# Barriers to Tor Research at UC Berkeley*

Karl Chen
EECS, UC Berkeley

Joseph Lorenzo Hall
SoI, UC Berkeley

Matthew Rothenberg
SoI, UC Berkeley

May 18, 2006

## 1 Introduction

There is increasing interest in anonymizing communication tools [2, 7]. Onion-routing, where secure packets are passed around a number of anonymizing network nodes, is at the frontier of the academic research on this topic. Tor [3, 4] is software that creates an onion-routing network where Tor clients can anonymously initiate communication through the Tor network. Although Tor has been shown to be vulnerable against attacks it was designed against, like timing analysis [8], it has garnered support from digital rights groups such as the Electronic Frontier Foundation (EFF).[1]

One area of Tor-related research that has been conspicuously absent from the literature are investigations concerning social questions surrounding its use. While Tor-related research to date has concentrated on attacks against the network [8] or the performance of the network [5], there has been no research into profiling the traffic on the Tor network. Our original project goal was to take steps in this direction. Specifically, we aimed to profile the traffic of the Tor network by looking at the services people use, the domains they visit and the geographical distribution of desired connections. The questions we intended to answer, included: "What do people use Tor for?", "Is Tor something that our institution should support?" and "Are there uses of the network that should be further disincentivized?"

Due to impediments we were unable to achieve our original goals by the end of the semester. Instead we report barriers to doing anonymous networking research, such as on Tor, at an institution such as UC Berkeley: legal barriers, bureaucratic barriers, technical barriers to running Tor in an enterprise environment, and possible solutions.

## 2 Technical Work

We planned to set up a Tor exit node at South Hall in the School of Information.[2] Per an agreement brokered with the University (see §3.2.2), the traffic on this node would be

*This is a project paper for Prof. Doug Tygar's IS219 course, *Privacy, Security, and Cryptography*, at the UC Berkeley School of Information.

[1] See: `http://tor.eff.org/`.

[2] The hardware for this project was generously donated by Prof. Tygar.

limited to a steady state of 1Mbps (125KB/s) and it would have had the default Tor exit policy[3] plus some enhancements necessary to operate the node on the UC Berkeley network. In addition, we would operate a software firewall on the machine that mimics the exit policy in the Tor configuration file.

We planned to log the destination IP address/port and time of each packet exiting our node. We spent some time thinking about logging methods that allowed us to capture all Tor-specific exit traffic but would not necessitate modifying the Tor source or adversely affecting the performance of the Tor network. We settled on the following scheme: the Tor application would have its own virtual network interface[4] and all traffic out of that interface (*accepts* through IPtables) would be logged to a MySQL database.

### 2.1 Performance and Storage Estimates

Storage in a MySQL database is quite efficient. Based on outside opinion, we expect MySQL on a typical server to be able to handle approximately 15,000 transactions per second. Assuming our node operates at 100% capacity, we anticipated about 100 packets per second.

We estimate total storage needs of about 40GB; our node has 120GB of disk space. Assuming the following database schema:

```
CREATE TABLE `tor_log` (
 `ip` int(10) unsigned NOT NULL,
 `host` varchar(255) NULL,
 `port` smallint(5) unsigned NOT NULL,
 `timestamp` datetime NOT NULL
)
```

We expected 14 bytes for each record[5]: 4 bytes for `ip`, 2 bytes for `port`, 8 bytes for `timestamp` and $(L + 1)$ bytes for `host`, based on length (however, since we won't be filling in the `host` field in real-time, we ignore it temporarily). With about 2 bytes of MyISAM overhead, we get a rough total of 16 bytes per record for integer values only (with `host` left NULL during logging). At 100/s, that amounts to 138 MB/day. At max capacity, that would be only 4.3 GB/month. With hostname information added later, we expect to have a 40GB table (or set of smaller tables, if necessary) over 2 months.

## 3 Barriers to experimenting with Tor

### 3.1 Legal barriers

Before we decided to undertake this project and design the particular experiment described above, we had to do an initial assessment of the legal implications of profiling traffic. We

[3]For a list of the default exit ports, see "Is there a list of default exit ports?" at `http://wiki.noreply.org/noreply/TheOnionRouter/TorFAQ`

[4]Since IPtables doesn't distinguish between applications and we want to log all traffic, this seemed wise.

[5]We obtain storage estimates per item using The MySQL Reference Manual 5.0, "Data Type Storage Requirements" available at: `http://dev.mysql.com/doc/refman/5.0/en/storage-requirements.html`

spoke with our legal colleagues here on campus associated with the Samuelson Law, Technology and Public Policy Clinic (SLTPPC) and had brief discussions with legal staff at the Electronic Frontier Foundation (EFF). From these discussions and our own research, we determined that merely aggregating TCP header information for research purposes holds little risk.

### 3.1.1 Federal Wiretapping Law

The most strict of federal legislation that would apply to traffic profiling is the Federal wiretapping laws.[6] The contents of real-time communications are given a very high level of protection in federal law [6]. The wiretap legislation makes it illegal to intercept the contents of electronic communications.[7] There are also high barriers, usually involving a court order, that agents of the government have to meet to engage in communications content monitoring. Further teeth of this law include statutory penalties as well as a civil cause of action and statutory damages.[8]

Besides the legal question, our team considered content monitoring to be highly unethical as people would have a reasonable expectation of privacy when exiting from a Tor node. Even if they did not have this expectation, it would be difficult to get such a research design through UC Berkeley's Committee for the Protection of Human Subjects (CPHS). After examining content-based restrictions on traffic profiling, it was clear that we would have to do something different or scrap the experiment altogether.

### 3.1.2 Federal Pen Register Law

Federal law treats the real-time capture of communication *attributes* entirely differently from content capture.[9] The "pen register"[10] statutes generally prohibit the real-time capture of "dialing, routing, addressing, or signaling information"[11] used in wire and electronic communications. However, the protection that the law gives communication attributes is substantially less than the content of communications.[12] Those seeking to install such devices (agents of the government) have a lower bar to meet: they simply have to apply for a court order authorizing the installation of a pen register device and the court "shall" issue the order as long as the information gained is likely to be relevant to an ongoing investigation.[13]

Aside of the lower protection for communication attributes in the pen register statutes, it was also important that there were some relevant exceptions to the interception of com-

[6] 18 USC §2510-2522.

[7] Note that there is another body of law, The Stored Communications Act (18 USC §2701), that covers access to communications that are not conducted in real-time.

[8] 18 USC §2511(4) (fine and up to five years in prison), §2513 (confiscation of equipment used) and §2520 (recovery of civil damages).

[9] 18 USC §3121-3127.

[10] Note that a "pen register" was originally a device used in association with telephonic communications. A pen register would record all the phone numbers calling into or out of a phone line. The definition of a pen register was extended with the USA Patriot Act to include any device that captures communications attributes.

[11] 18 USC §3121(3).

[12] See [6] at 48.

[13] 18 USC §3123(a).

munication attributes and that not just anyone could file suit against us. First, there is an exception outlined in 18 USC §3121(b)(1) that states the prohibition does not apply "[...] to the protection of users of that [communications] service from abuse of service or unlawful use of service". As discussed below in §3.2.3, one of the conditions we had to agree to in our negotiations with the University was to determine for what uses people were using Tor so that we could recommend disincentives to illegal and abusive uses of the Berkeley exit node. This was entirely within the scope of our experiment.

Second, the pen register statutes do not include a civil cause of action like the wiretapping statutes. In order for our project to attract the attention of a legal complaint, it would have to be interesting enough for a state or federal prosecutor to determine it was worth their effort and resources to pursue us. As we had no malicious intent and are squarely aiming our experiment at answering questions fundamental to the research around anonymous proxies and anonymizing communications tools, we estimated that the risk of attracting attention from a prosecutor would be very low. From our examination of the Federal laws relating to real-time information capture of electronic communications, it made the most sense to only capture attributes of Tor traffic.

#### 3.1.3 State Law

We also looked at California State Law. Like the Federal statutes noted above, the California Penal Code has provisions for the protection of real-time communications.[14] We noted that the protection of the content of communications was much higher in California state law. However, California law does not contemplate pen registers. While the higher level of protection for content in State law would have made any content-based experiment even more difficult, the lack of protection for communication attributes was further encouragement that our experiment – which was designed only to capture TCP header information – carried low legal liability.

### 3.2 Institutional Barriers

In addition to the specter of legal liability that we had to take into account while designing and running our experiment, there were also a variety of institutional barriers that we had to overcome before we could begin work. Unfortunately for our research, this proved to be intractable in the current institutional environment given the current technical limitations of Tor.

#### 3.2.1 Departmental Approval

The first step in running an experiment like this was to get local departmental approval. We first got the approval of our Dean, AnnaLee Saxenian, by explaining what Tor is and how it would be useful in research, pedagogical and administrative uses of anonymizing communications tools. Dean Saxenian required that we get the approval of our computer system administration staff and further recommended that we get a faculty sponsor or two who would be more familiar with the technical implications of running a Tor exit node in the

[14] California Penal Code §629.50-629.98, available at: `http://leginfo.ca.gov/calaw.html`.

department. After explaining the technical details to our senior system administrator, Kevin Heard, we were able to convince him that our Tor node would play nice on our local network. We also contacted two faculty members, Doug Tygar and John Chuang, as potential faculty sponsors for this project. Both faculty agreed that this would be a useful feature to have active on our network and to be available for other researchers to use.[15]

### 3.2.2 Berkeley's Minimum Security Standards and CISC

The Berkeley campus has a number of policies that are relevant to information technology, computing and network resources. After examining them, we were able to determine that the policy that we would most likely run afoul of was Berkeley's Minimum Standards for Security of Berkeley Campus Networked Devices (MSSBNCD).[16] Item seven of the MSSBNCD is entitled "No unauthenticated proxy services" and reads, in part:

> Although properly configured unauthenticated proxy servers may be used for valid purposes, such services commonly exist only as a result of inappropriate device configuration. Unauthenticated proxy servers may enable an attacker to execute malicious programs on the server in the context of an anonymous user account. Therefore, unless an unauthenticated proxy server has been reviewed by SNS and approved by the CISC as to configuration and appropriate use, it is not allowed on the campus network.

Tor is definitely within any definition of an unauthenticated proxy server as that is exactly its purpose, to provide anonymous communications that cannot be traced back to an individual or location.

In order to operate a Tor exit node and to run our experiment, we had to get an exception to the MSSBNCD by presenting our rationale for allowing a Tor exit node on campus to the Campus Information Security Committee (CISC), made up of IT and system security directors on campus and the CIO of UC Berkeley, Shelton Waggener.[17] After an involved discussion, CISC conditionally approved a Tor exit node at the School of Information with the following conditions:

1. Approval was for one year (expires 2 Feb 2007);
2. CISC approval was conditional on the approval of the CIO;
3. We had to block all traffic to UC Berkeley IP addresses;
4. We had to explain the proposal to Barbara VanCleave Smith at risk management and get her approval;

[15] For example, a postdoctoral researcher here at the School of Information, Sonja Buchegger, has some interest in conducting research with Tor to allow for limited notions of reputation. This would allow Tor users to, for example, be able to edit Wikipedia, which currently bans Tor users do to the prevalence of abuse. If Tor users could demonstrate that they had behaved in the past through some sort of basic signal, then services like Wikipedia could only block users that had bad reputations.

[16] See: `http://security.berkeley.edu/MinStds/AppA.min.htm`.

[17] See `http://security.berkeley.edu/CISC/`.

5. We would need to operate a software firewall that mimics the exit policy with which we run Tor. SNS reserved the right to require a hardware firewall;

6. SNS may shut it off (ban the IP address from making any connections outside and inside of the Berkeley network) immediately, temporarily and/or permanently if it gives them too many headaches,[18] and;

7. We would have to include in the exit policy the IP-authenticated addresses for services that Berkeley subscribes to so that Tor users could not specify the Berkeley Tor node as a preferred exit node and thereby get access to library services only licensed for student, faculty and on-campus uses.

Few of the conditions above seemed to be show-stoppers, at first.

#### 3.2.3 Risk Management

We first tackled the requirement to meet with Risk Management staff and explain the proposal and Tor. A common theme in this process has been the difficulty of explaining to non-technical audiences what Tor does and especially why anonymous communication is desirable. When we met with Risk Management they were concerned with legal liability in general, how the UC Berkeley name might be "tarnished" and, specifically, "how could Tor be used by terrorists?". We explained that anonymous communication tools were useful in general in any case where there might be a potential risk involved with exposing who or where the communication originated. The example that usually gets the most traction with administrators is to describe how it can be counter-productive to expose one's identity or location during competitive analyses of firms or in faculty or other high-profiling hiring processes.

In the end, the main concern of Risk management was that they couldn't grasp what Tor was going to be used for. Where most computing and network services have specific functions, Tor's function of anonymizing TCP/IP traffic in general is unique. We brokered an agreement with Risk Management that turned on providing them a description, in non-technical terms, of what people use Tor for. This turned out to be very convenient as that is the experiment that we had designed.

#### 3.2.4 Library Services Licensing

Up to this point, all the conditions that we had agreed upon with the University to run a Tor exit node and conduct our experiment were manageable. However, the requirement to block all exit traffic to services with which the UC Berkeley library subscribes proved to be intractable.

After contacting those responsible for managing library subscription services at UC Berkeley, it became clear that no one has an exhaustive list of all the subscription services. The system for keeping track of web-based library services seems to be the following:

[18] A corollary to this was: We agreed to work with SNS to mitigate any issues as we went forward.

- When UC Berkeley signs a subscription agreement with a service, the domain name (say, `portal.acm.org`) for the service is added to the library's `proxy.pac` file.[19]
- When an agreement expires or is terminated, the entry is deleted.
- When a second service is added that has a similar domain name to a previous entry, (say, `blogs.acm.org`) instead of adding another line to the proxy file, a simple regular expression (e.g., `*.acm.org`) is added in place of the previous entry to cover both services.

For our purposes, this posed a number of problems. First, we would need to translate the long ($\sim$ 3000) list of domain names in the proxy file into IP addresses. Second, we would need to use a service like Netcraft's SearchDNS[20] for entries that used wildcards (`*`) in order to get an exhaustive list of third and forth level domain names. Third, we would need to update this list periodically as the IP addresses associated with domain names may change.

Finally, this posed direct issues with Tor itself. Tor's directory distribution protocol was not designed to handle large exit policies. In our case, our exit policy would be anywhere from 3,000 to 150,000 entries long in order to block exits to subscription services.[21] This would quite literally break Tor. It would result in a sharp increase in the size of the directory information that each exit node would have to download.[22] Also, since Tor exit nodes process the exit policies of other nodes in serial to determine how to route traffic, an enormous exit policy would dramatically reduce the performance of slower exit nodes.[23]

To summarize, through design of our experiment and negotiation within our institutional environment, we successfully surmounted almost all the barriers with which we were presented. We were unable to reconcile the demands of blocking exits to library subscription services and the technical capabilities of Tor.

# 4 Options for Tor Research on the Berkeley Campus

With the preceding discussion in mind, we now turn to an analysis of the options available to Tor researchers in the institutional environment at Berkeley and given the current technical limitations of Tor. We do this to highlight how restricted the options are for Tor research (and in some cases with anonymous proxies in general).

## 4.1 Operate in Middleman Mode

One option is to operate a Tor node in middleman mode. This would mean that the node does not allow any exit traffic at all, the exit policy would be minimal and no infrastructure

[19]Note that you can see the contents of this file by accessing the following url: `http://proxy.lib.berkeley.edu:7777/proxy.pac`

[20]See `http://searchdns.netcraft.com/`.

[21]These figures come from assuming one IP address per proxy list entry (which is not realistic but a quick lower bound) and then taking the number of domains in `*.acm.org`, which is 52 according to SearchDNS, and multiplying by the number of entries in the proxy list to get 156,000.

[22]Tor exit nodes refresh their directory cache every 20 minutes by default.

[23]Note that many exit nodes are run on older hardware.

for dealing with the library's proxy list would be needed. While this would be beneficial for the Tor network – high-bandwidth middleman nodes decrease latency in the network – it would prohibit any research that relied upon exit traffic or running a modified exit node of some sort.

## 4.2 Operating in a Whitelist Mode

A Tor exit node could be run in a "whitelist" mode where only a certain set of IP addresses are allowed to receive exit traffic from the exit node. This would be useful, for example, for running an experiment that would only deal with traffic, say, to the top 10 or 100 websites on the internet. The advantages of this mode would be running a very small exit policy and not needing to worry about the library's proxy list. However, this would provide a very narrow view of the internet and would be biased to certain types of traffic.[24] Also, if this whitelist was changed periodically, in a sampling fashion, the experiment would be limited by the amount of time that it takes an exit node's changed exit policy to propagate through the network.[25]

## 4.3 Blocking All Exits to Library Services

Another option is to operate in the mode that our experiment had designed; by blocking all traffic to identifiable, domain-name registered library services. The main advantage of this mode is that it is highly precise and minimizes blocked traffic. The disadvantages of this mode are those we've listed above: it results in a very long exit policy that doesn't work with Tor, it blocks traffic to non-subscription sites,[26] and doesn't block traffic to services that don't have domain names (unnamed services).

## 4.4 Blocking Whole Netblocks

Finally, as a variation on the last option, the exit policy list could be made even shorter by blocking whole netblocks instead of trying to be so precise. While this would result in a drastically smaller exit policy when compared to the last option, it would still be on the order of thousands of items long. It is unclear if an exit policy on the order of thousands of entries rather than hundreds of thousands would make much of a difference in the impact on the Tor network. Of course, this would block even more legitimate traffic than the previous option.

[24] For example, only whitelisting the top web domains would bias towards web traffic.

[25] A Tor developer, Roger Dingledine, has told us that this takes about 2-3 hours.

[26] Note that all of `*.princeton.edu` is matched by the proxy list. It is undeniable that there is non-subscription content elsewhere in that domain.

# 5 Solutions

## 5.1 Remove IP Authentication

This entire experience has underscored the problem with Berkeley's use of IP authentication in a large, diverse, network. For example, during the recent ASUC online election, ballots were allowed to be cast from any Berkeley IP except AirBears wireless IPs. ASUC was able to do this because ASUC controlled the IP authentication on their voting servers, and AirBears is a single subnet, so easy to exclude. With the border between hosts internal to the Berkeley network and those outside increasingly blurred, it is no longer useful to use this distinction. IP authentication has been known to be broken in the security community for many years [1]. While policies like the MSSBNCD formally discourage IP-based authentication *on campus*, it is important that we champion the rationale behind this policy with our connections to services that we do business with off campus. Ideally, library services should use a more robust form of authentication, for example Kerberos or with a hook into to CalNet. However, this requires non-trivial modification of each library service and will make access harder, so this would be difficult to deploy.

A medium-term solution is to separate the Berkeley IP address space into two or more tiers of trustedness, grouped by subnets. AirBears IP addresses should be considered untrusted. Authorized open proxies such as a Tor node should also have untrusted IP addresses. Library proxy servers should be trusted. Then, we could tell services to which we subscribe to trust the trusted subnet of our network. This would be a good solution in the sense that it would require little effort on behalf of library services and the only side-effect that on-campus users would notice is that they would now have to CalNet authenticate via the library automatic proxy configuration file to use library services from both on and off campus.

## 5.2 Fix Tor's exit policy distribution algorithm

In current implementation, Tor synchronizes the exit policy of all nodes with a push-broadcast – that is, a Tor node sends its exit policy to a central directory and then downloads all other nodes' policies from the same place. This has worked so far because Tor developers frowned at anyone running a Tor node with a large exit policy. However, even with small exit policies, this will not scale with the number of nodes increasing as interest in anonymous communication increases as well. There are known solutions to this data distribution problem in Computer Science research. For example, Tor nodes can use a Distributed Hash Table system [9], send periodic diffs of the exit policy, or use a hybrid publish-subscribe policy. To deal with performance problems related to exit nodes processing long exit policies in serial, there are a variety of tree-based algorithms that would be much more efficient.

# 6 Conclusion

Both aspects of the institutional environment here at UC Berkeley and the technical limitations of Tor have frustrated our research efforts this semester. We did look into taking our equipment to a collocation facility that had Tor-friendly Service Level Agreements and Terms of Service. The number of Tor-friendly collocation facilities near us is small, but

non-zero and we've estimated the costs for the level of service that we would need to be in the range of $150-250 per month. We hope to actually run our experiment at an off-campus collocation facility sometime in the near future. However, it is unfortunate that we were not able do this research on the Berkeley campus.

We'd like to leave open the option of doing something further than the above proposed work in the future, and we would hope it would be back in our Berkeley laboratory facilities. One idea we've had to extend some of this work would be to measure Tor's performance under load. However, we'd like further work to be compelled from the traffic profiling experiment outlined above – for example, if we see evidence of peer-to-peer traffic bypassing the current default exit policy blocking, we'd like to measure how useful Tor is for different types of shared media. As an additional example, if we find that 99% of the traffic is web surfing, we might want to do latency measurements and attempt to improve the latency through incorporation of additional features like those in next-generation onion-routing networks like Cashmere [10]. Finally, experimentally verified information about the type of traffic that a Tor exit node introduces to a typical network would be valuable information to administrators attempting to set network policy requirements.

# 7 References


[1] S. M. Bellovin. Security problems in the TCP/IP protocol suite. *SIGCOMM Comput. Commun. Rev.*, 19(2):32–48, 1989.

[2] H. Bray. Beating censorship on the internet; tools mask user IDs, give alternative routes to sites. *Boston Globe*, February 20, 2006.

[3] R. Dingledine and N. Mathewson. Tor protocol specification. Technical report, The Free Haven Project, February 2006.

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

[5] R. Dingledine, N. Mathewson, and P. Syverson. Challenges in deploying low-latency anonymity (unpubhlished draft). 2005.

[6] S. Freiwald. Online surveillance: Remembering the lessons of the wiretap act. *Alabama Law Review*, 56(9), 2004.

[7] J. D. Glater. Privacy for people who don't show their navels. *New York Times*, January 25, 2006.

[8] S. J. Murdoch and G. Danezis. Low-cost traffic analysis of Tor. In *Proceedings of the 2005 IEEE Symposium on Security and Privacy*. IEEE CS, May 2005.

[9] S. Rhea, B. Godfrey, B. Karp, J. Kubiatowicz, S. Ratnasamy, S. Shenker, I. Stoica, , and H. Yu. OpenDHT: A public DHT service and its uses. ACM SIGCOMM, 2005.

[10] L. Zhuang, F. Zhou, B. Y. Zhao, and A. Rowstron. Cashmere: Resilient anonymous routing. In *The 2nd Symposium on Networked Systems Design and Implementation (NSDI)*. USENIX, May 2005.