A Case Study on Measuring Statistical Data in the Tor Anonymity Network

Karsten Loesing[1], Steven J. Murdoch[1,2], and Roger Dingledine[1]

[1] The Tor Project
[2] Computer Laboratory, University of Cambridge, UK

Abstract. The Tor network is one of the largest deployed anonymity networks, consisting of 1500+ volunteer-run relays and probably hundreds of thousands of clients connecting every day. Its large user-base has made it attractive for researchers to analyze usage of a real deployed anonymity network. The recent growth of the network has also led to performance problems, as well as attempts by some governments to block access to the Tor network. Investigating these performance problems and learning about network blocking is best done by measuring usage data of the Tor network. However, analyzing a live anonymity system must be performed with great care, so that the users' privacy is not put at risk. In this paper we present a case study of measuring two different types of sensitive data in the Tor network: countries of connecting clients, and exiting traffic by port. Based on these examples we derive general guidelines for safely measuring potentially sensitive data, both in the Tor network and in other anonymity networks.

1 Introduction

Tor [1] is an anonymous communication system that permits its users to surf on the Net without revealing their identity or location. Tor is used by private citizens, corporations, and governments to protect their online communications, as well as by users trying to circumvent censorship. Its basic principle is to redirect traffic over virtual tunnels through three independent Tor nodes, to make it hard for an attacker to link origin to destination.

The scale of the Tor network makes it attractive for researchers who want to study real deployed anonymity networks. McCoy *et al.* published a study [8] that characterizes the usage of Tor; they tried to answer how Tor is used and mis-used, as well as discover what types of users are using Tor. We have talked to other researchers who have performed similar studies in the Tor network (or would like to), but they have not published their results because of technical or legal concerns around safe data collection. From a technical point of view, measuring data in the Tor network can easily be performed by setting up a Tor relay and logging all relayed user traffic. However, this approach raises ethical questions ranging from legal issues over hurting users' privacy to lack of community acceptance. The big threat is that an adversary could make use of this data to correlate Tor users with traffic exiting the Tor network. If researchers measure the live Tor network in a way that does not protect the users' privacy, and the underlying data of these studies are leaked, the protection that Tor aims to provide might be in danger. Worse, if the conservative researchers choose not to publish in case their data or process is not safe enough, then the only groups that do publish will be ones that are confident (whether rightly or wrongly) that they got every detail right.

In this paper we describe a case study of safely measuring two types of sensitive data in the Tor network: client IP addresses and exiting traffic. We consider this data to be necessary to make Tor better by making it faster, giving us a better sense of the level of anonymity Tor can provide, and making it harder for censors to block the Tor network. At the same time, both types of data could help an adversary de-anonymize Tor users if measured without caution. We identify possible problems with measuring this data and present our measurement approach which avoids putting the Tor users at risk. At the end of the paper we derive general guidelines for measuring potentially sensitive data that could be used by other researchers and in other anonymity networks.

The next section gives a brief background on Tor. Section 3 describes the goals of statistical analysis in the Tor network. Section 4 discusses the potential ethical problems when doing so. In Section 5 we present our case study of measuring client IP addresses and exiting traffic, and summarize general guidelines for similar cases in the future. Section 6 concludes the paper.

2 Background on Tor

Tor aims to prevent users from being linked with their communication partners; i.e. someone monitoring a client should be unable to find out which servers he is accessing, and a server (or someone monitoring the server) should be unable to find out the identity of clients using Tor to access it. While the original goal of Tor was to enhance privacy, recently Tor has become popular amongst users who wish to circumvent national censorship systems, such as those in countries like Iran and China. Tor's primary security property (an attacker cannot find out which websites a user is visiting) also makes it useful for circumvention because the censor is not able to selectively block access to blacklisted sites.

Tor users download and install the Tor client software, which acts as a SOCKS proxy interfacing their client software (typically a web browser) with the Tor network. This software first connects to one of the *directory authorities*, which are operated by (currently seven) individuals trusted by the Tor Project. From these authorities the software downloads a list of available Tor *nodes* which are relays run by volunteers. The Tor client then selects three of these nodes, and builds an encrypted channel to the first one (called the entry node). Over this encrypted channel, the Tor client builds an encrypted channel to the middle node, and then via this channel, connects to the third node (the exit node).

In this way, the client has a connection to the exit node, but the exit node is not aware of who the entry node or client is; similarly the entry node does not know which exit node the client has selected. The client can then request that the exit node connects to a particular destination server, such as a website accessed by the user. Messages to the server are encrypted multiple times: first to the exit node, then the middle node, and finally to the entry node. As a message is relayed by each node, one layer of encryption is removed. Thus the original message is known only to the exit node. Replies from the server are encrypted by each node along the path, and then decrypted by the client. Therefore messages coming into a node cannot be matched, based on content, to the corresponding message leaving the node.

However, Tor does not prevent an attacker from using traffic analysis to de-anonymize users. Here, the timing of packets in streams leaving the Tor network is recorded. Then, a target stream which is coming into the network is correlated with each candidate output stream. Because nodes do not significantly delay packets, it is likely that the output stream corresponding to the target incoming stream will become clear. Experiments have shown that this conclusion can be reached after only a few packets [10].

Only very capable adversaries are likely to be able to simultaneously record network traffic across the entire Internet, so this attack is unlikely to be a concern to most Tor users. However, traffic analysis can still work even given incomplete information; it just takes more data to get the same level of confidence. For example, by only recording 1 in 2000 packets, it is still possible to de-anonymize streams [9]. In general, it is impossible to accurately estimate how much distortion must be applied to a data set before it is no longer useful to an attacker. This is primarily a consequence of *auxiliary information* [2] – data which is known by the attacker but not by the individual distorting the data set.

Even excluding the problem of auxiliary information, it is not possible to estimate whether a particular conclusion that could be reached by traffic analysis is sensitive, because we cannot accurately know the privacy requirements of users. For example, the mere fact that Tor is being used can be problematic, for example if there are only a small set of candidates for a particular action. Therefore the safe option is to not collect any information about an anonymous communication network. However this extreme approach can harm users too, e.g. data which could be used to detect attacks against the network would be unavailable. Instead, in this paper we discuss approaches that can be taken to allow useful data collection, while minimizing the potentially harmful consequences.

3 Goals of statistical analysis

The number of Tor relays has increased from 32 in May 2004 [1] to roughly 1500 in October 2009 carrying a total of 250 MiB/s. There are estimated to be hundreds of thousands Tor users every day routing their data through the Tor network. This volume and diversity makes the Tor network an interesting object of study, both to learn more about deployed anonymity networks and to improve Tor for its users.

Performing statistical analysis in the Tor network can serve various purposes. Statistics based on the list of publicly known relays [5] can help observe trends in the structure of the Tor network: which countries are contributing relays and bandwidth, what software versions are deployed, how many relays are running on dynamic IP addresses, etc. Statistics based on performance measurements [4, 6, 13] can help detect performance bottlenecks and evaluate the effect of performance improvements. These performance measurements are conducted with self-generated requests rather than by observing other users' requests.

The next step of statistical analysis in the Tor network is evaluating network data, i.e. data that is based on real user requests. The first thing we want to learn about usage of the Tor network is *who* uses Tor. Tor is meant to provide anonymity and censorship circumvention to people worldwide. In particular, one goal is to make Tor more useful for people in various possibly censoring countries around the world. Usage statistics can help in detecting in which of these countries Tor's efforts are succeeding and which ones need more work, e.g. by performing additional trainings.

As an example, the statistics shown in Figure 1 (a) indicate that Tor usage significantly increased from Iranian IP space in June 2009 after the Iranian elections. (Note that neither of the graphs contains actual user numbers, but rather data that might be used in the future to estimate user numbers; however, the relative increase in usage is already meaningful.) After publishing these statistics, more people were motivated to set up relays and help support the Tor network and Iranian Tor users, in turn improving the security and performance of the network.

Similarly, usage statistics can help discover attempts to block users from reaching the Tor network. Such a blocking event has been observed in late September 2009 when China blocked access to most Tor relays as shown in Figure 1 (b). At the same time, bridge usage from Chinese IP addresses increased significantly by a factor of 70 as compared to the time before the blocking. Bridges are Tor relays that are not listed in the public directory, making it harder for the censor to locate and block them; we deployed the bridge design preemptively as one of the steps in the arms race, so users would have another option ready when a government decided to block connections to the public Tor relays [11]. Statistics on usage by country can help build an automatic early warning system to detect country-wide blocking events.

Another motivation for statistics on usage of the Tor network is to make Tor faster by finding out *what* Tor is used for. These statistics include the observation of what kind of applications are used over the Tor network by looking at exiting traffic. Such statistics can help reveal what share of traffic is used for low-latency applications, like web browsing or IRC, or for bulk file transfers, like file sharing. While low-latency networks like Tor have been designed to support low-latency applications, applications like file sharing increase the load on the network and increase latencies for everyone. It would be desirable to know – and to track over time – what portion of Tor traffic is used for each application class.

Another type of statistics, related to the question of what Tor is used for, is the comparison of overall traffic volume per TCP port versus the advertised bandwidth capacity per port. Each Tor relay has an *exit policy* that specifies

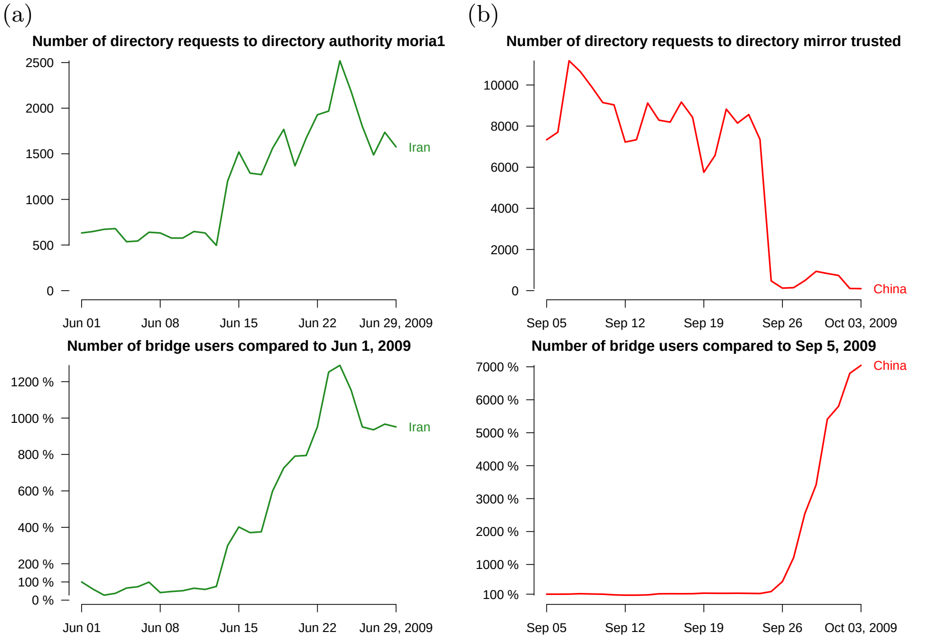

Fig. 1. Statistics related to the number of Tor users in Iran in June 2009 (a) and China in September 2009 (b)

what addresses and ports it is willing to connect to. When a client chooses its path for a given application request, it chooses a relay at random from those that permit the client's request. Selection is weighted by the relays' advertised bandwidths in order to achieve load balancing among relays. However, this approach has the drawback that relays with more permissive exit policies attract far more clients than relays that permit only a small number of addresses or ports. Statistics on exiting traffic per port can help improve load balancing by learning about the overall traffic volume per TCP port. Subsequently, clients could direct more traffic to relays with less permissive exit policies if possible.

Table 1 shows an example of the distribution of traffic to ports. Of these ports, port 80 is the one that has the largest share of read bytes (1.8 GiB) and opened streams (867896). We cannot say what fraction of this traffic can be amounted to web surfing, but the small amount of read bytes and the large number of opened streams speaks for the web surfing assumption and against file sharing. The measuring exit node permitted exiting to all ports, so is not representative for exit nodes in general. In particular, this exit node has seen a disproportional share of traffic on the non-default ports. For example, port 4662, which is typically used for file-sharing applications, sees the largest share of written bytes with a total of 6.3 GiB.

Questions like the ones described above can only be answered by performing statistical analysis on network data in the deployed Tor network. In some cases it may be sufficient to make assumptions about user behavior to build an ano-

Table 1. Statistics on traffic as seen by an exit node with unrestricted exit policy over one day distributed to TCP ports

Port	MiB written	MiB read	Streams opened in K	Default exit policy
80	666 (1.5 %)	1799 (31.7 %)	868 (16.4 %)	Yes
4661	756 (1.7 %)	10 (0.2 %)	25 (0.5 %)	No
4662	6432 (14.8 %)	75 (1.3 %)	176 (3.3 %)	No
6881	291 (0.7 %)	63 (1.1 %)	47 (0.9 %)	No
51413	387 (0.9 %)	40 (0.7 %)	46 (0.9 %)	Yes

nymity system. But with recent growth of the Tor network, these assumptions need to be questioned. Statistics can help make Tor more useful for censored users and improve performance for all Tor users.

4 Ethical problems

Performing statistical analysis in an anonymity network is problematic per se, not to mention statistics on network data. The problem is that statistics must not undermine the security properties that the anonymity system is designed to provide. There are several sets of guiding principles which can be followed when collecting statistics in an anonymity network. These include: legal requirements, user privacy, ethical approval, informed consent, and community acceptance.

Legal requirements. We cannot gather any statistical data in the Tor network that is against the law. This limitation becomes even more complicated because data collection needs to take place at multiple locations in the Tor network which are subject to different laws. Therefore, in order to be safe, data collection should be performed on the lowest common denominator of the various laws of countries with measuring nodes. These laws typically fall into two categories: laws specifically prohibiting wiretapping (common worldwide), and generic personal information data protection regulations (in the EU). However in both cases, how these apply to data collection in Tor is uncertain. Wiretapping legislation differentiates between traffic data (headers) and content, but on the Internet there are so many nested protocol layers it is difficult to point to a single boundary. Data protection regulations are even more vague, merely specifying general principles such as only collecting enough information necessary for business purposes, and ensuring that is is not improperly processed. But even though we are bound by laws, only following laws is insufficient from an ethical perspective anyway – especially in our case of an anonymity network. The constraints as described below force us to be even stricter than laws would require.

User privacy. The statistics that we gather must not harm Tor's security properties. In the simplest case, the gathered and subsequently published statistical data must not be useful for an adversary to de-anonymize users. In particular, an adversary that is running one or more Tor relays herself and thereby observing one side of a circuit must not learn any useful information from our statistics about the other side of the circuit. Further, the collection of possibly sensitive data must not make the measuring relays a more attractive target for hacking attempts. The measuring relays should therefore not store sensitive information that an adversary might learn about by hacking other Tor relays, as far as possible. Finally, the code that is used for measuring statistical data should not help an adversary to extend their own logging capabilities more than necessary. A less tech-savvy adversary should not be able to misuse the measurement code to find the places in the Tor source code that could be changed to log even more sensitive network data too easily. Obviously, some of these threats cannot be solved, but only mitigated. The goal of statistical analysis in the Tor network should be to sacrifice as little user privacy as necessary while making the impact of statistics as large as possible.

Ethical approval. For research performed in academic institutions it is sometimes necessary to gain ethics approval from the Institutional Review Board (IRB). However, while such committees are well established in medical research or psychology, they are not in computer science. Faculty-level boards may not have the necessary experience to decide whether a particular experiment is ethically justifiable. There is also significant variation between countries and even institutions on what types of activities require submission to such a committee. As an example, McCoy *et al.* responded to controversy over their PETS 2008 study [8] by asking their IRB whether their experiment would have needed approval. The committee's conclusion was that the research was not classified as using human subjects and was outside their remit [7].

Informed consent. A common principle for ethical approval is that researchers obtain informed consent from subjects. This approach is particularly difficult for an anonymity network where the identity of users is in itself sensitive information. In cases where this is not possible, for example psychology experiments where it is necessary to deceive subjects, stricter ethical rules must be applied and it is more common for IRB approval to be needed. While our data collection methodology will always be public information, we cannot be sure that users will read this documentation before using the system. We must therefore only carry out actions which we believe will cause no harm.

Community acceptance. Even if statistics are perfectly legal and do not harm any security properties, it is important to have the community of users, relay operators, and researchers accept them. An anonymity network like Tor depends to some extent on the trust in the other participants. The biggest threat is probably that we might fail to communicate our plans to gather statistics in the Tor network to our community. It is important that our community understands the need for gathering statistics and exactly how measurements take place. If our community starts thinking that we might not be honest in how we gather our statistics or might not be doing what is best for the Tor network, we lose their trust and the Tor network might lose their support. One approach to openness is to publish all the data we collect; but this may conflict with ethical or legal requirements that data is not improperly processed, and it requires that we apply very strict anonymization methods at the time of collection. In the networking research field, on the other hand, collected data is often only partially anonymized (so as to maximize their usefulness), but data sets are only available on signing a legal agreement to not attempt to de-anonymize users.

5 Case study

In the following case study we demonstrate the challenges of measuring statistical data in the live Tor network. We consider network data like countries of connecting clients and exiting traffic by port, both of which belong to the most sensitive types of data in an anonymity network. After all, the main purpose of an anonymity network is to keep the correlation between the users' IP addresses and the requests that they send to the network distinct. Measuring either client IP addresses or exiting traffic bears the risk of misuse by an adversary. Therefore, special caution must be taken when deciding how to measure these network data and how to process them in a way that they cannot aid an attacker. Subsequent to the two example cases, we derive a few general guidelines for measuring statistics in the Tor network that might be applied by other researchers studying the Tor network and in other anonymity systems as well.

5.1 Countries of connecting clients

The first question to answer is *who* uses the Tor network. This question can be answered by looking at IP addresses of connecting clients. In particular, we want to learn how Tor usage is distributed by countries and how this distribution changes over time. Similarly, statistics about Tor usage can be used to automatically detect blocking of the Tor network. Sudden changes in Tor usage by country would indicate country-wide blocking events.

There are various places at which clients "enter" the Tor network and where their IP addresses can be recognized. The first group is *entry nodes*, which are the first relays in the clients' circuits. Clients need to connect directly to entry nodes in order to hide their IP addresses from subsequent relays and the target they are connecting to. Hence, entry guards learn about the clients' IP addresses, but not what actions they perform over the Tor network. Relays can easily recognize whether a connecting IP address is a client or a relay from the directory of all relay IP addresses. If the connecting IP address is a known relay, they are acting as middle or exit node in a circuit. If not, the connecting IP address is a client.

The second group of places that can observe client IP addresses are *bridges*. Bridges are relays that are only known to a small set of clients that could otherwise not connect to the Tor network. Similar to entry nodes, bridges learn about client IP addresses from incoming connections. In contrast to entry nodes, bridges can be sure that every connection they see is from a client, so when making a list of clients they do not need to filter out relay IP addresses.

The third group of places at which clients connect directly to the Tor network are *directory nodes*, which are either the directory authorities or directory mirrors. Clients connect to the directory authorities during the bootstrapping process when they do not know about any relays other than the hard-coded directory authorities. Clients download the current network status (a list of relays through which they can build circuits), and then periodically connect to directory mirrors to update their view on the network. In most cases clients connect directly to the directory mirrors instead of building a circuit to fetch the information privately, because there are little or no privacy issues in downloading a network status. The requests that clients send to the directories are categorized into two versions of network status formats, one of them requested by clients up to Tor version 0.1.x and the other one by clients running Tor version 0.2.x. Only requests for network statuses are counted in the statistics.

It becomes immediately obvious that client IP addresses are highly sensitive information in an anonymity network. The mere fact that someone connects to the Tor network is not (and cannot be) protected by the Tor protocol. However, this information should not leak to an adversary easily. An adversary that is trying to break the anonymity properties of Tor tries to link a client's IP address to a request leaving the Tor network. If there were such a list of client IP addresses, an adversary could monitor the traffic exiting the Tor network and try to correlate clients to outgoing requests or incoming responses.

As a first step to protect client IP addresses from leaking to an adversary, they should be resolved to a country as soon as possible. Since analysis takes place on the country level, we do not need to keep the exact IP addresses of clients. This resolution can be done using a local GeoIP database that maps IP addresses to country codes. Tor versions since June 2008 include such a GeoIP database that is 2.5 MB in size. In the case of counting events per country, e.g. directory requests, this resolution can take place immediately. However, if the goal is to count unique IP addresses per country, IP addresses need to be stored in memory in some form in order to detect duplicates. In the process of writing this data to disk, IP addresses can be resolved to countries and the number of unique IP addresses per country can be summed.

The resolution of IP addresses to countries is an important first step, but it is not sufficient. The information that a client from a certain country has connected to the Tor network at a certain time might still be too sensitive to be published, especially for countries with only few Tor users. Therefore, as a second step, events are accumulated over an amount of time that makes the data less useful for an adversary. We assume that an accumulation of events over the course of one day is sufficient to prevent an adversary from learning too much. This accumulation means that statistics will not be able to discover changes in Tor usage by time of day, but this seems like a reasonable compromise.

Finally, the exact number of events from a certain country per day might still reveal sensitive information if that number is very low. In general, exact numbers pose a risk when the adversary can generate such events herself and observe how many other events have occurred in the same time. As a third step,

```
dirreq-stats-end 2009-08-20 17:16:35 (86400 s)
dirreq-v2-ips us=4136,de=3744,cn=3552,gb=1120,ir=1024,kr=952,it=848,
  fr=768,ru=768,??=688,ca=616,se=480,es=392,pl=392,au=368,[...]
dirreq-v3-ips us=6024,de=5176,cn=3384,fr=2208,kr=1328,it=1288,ru=1120,
  gb=1048,se=816,ca=808,pl=800,??=744,ir=728,jp=600,br=576,[...]
dirreq-v2-reqs us=7136,cn=5608,de=4728,kr=3816,gb=1568,ir=1464,ru=1136,
  it=1120,fr=1096,??=968,ca=936,tw=720,se=664,jp=576,au=552,[...]
dirreq-v3-reqs us=7800,de=5944,kr=4368,cn=4208,fr=2632,ru=1616,it=1576,
  gb=1272,ir=1096,ca=1024,??=1016,se=976,pl=944,tw=792,au=784,[...]
```

Fig. 2. Number of IP addresses and requests for network statuses as observed by a directory mirror

the exact number of events is concealed by introducing artificial imprecision. This is done by rounding up event numbers to the next multiple of 8.

All statistics based on client IP addresses are processed by the measuring entry node, bridge, or directory before publication as described above. Figure 2 shows an example of unique client IP addresses and number of requests for relay lists on a directory mirror. The first line indicates when the data were written and what time interval is covered. The remaining lines state how many unique IP addresses or directory requests have been observed from which country for the two possible network status versions. For example, this directory mirror has observed 7800 requests for version 3 network statuses from 6024 unique IP addresses from the United States. The country code ?? stands for IP addresses that could not be resolved to a country. The exact data format is described in the directory protocol specification document [12].

The statistics from entry nodes and bridges look similar, except that they only contain unique IP addresses and no requests of any kind. Directory mirrors and entry nodes upload their statistics to the directory authorities where they can be downloaded by anyone who is interested. Bridges upload their statistics to the bridge authority. Before publication of bridge statistics, all possibly identifying information about the bridge needs to be removed. Otherwise, bridge statistics might reveal to an adversary where bridges are located. Instead, bridges are assigned a unique bridge identifier, so that statistics of the same bridge can be observed over time.

5.2 Exiting traffic by port

The analogue of IP addresses of clients connecting to the Tor network is traffic exiting from the Tor network to the Internet. In contrast to the question *who* is using the Tor network that can be answered by looking at client IP addresses, exiting traffic can reveal more information about *what* the Tor network is used for. Statistics of exiting traffic include what kind of applications are used over the Tor network, or the comparison of overall traffic volume per TCP port versus the advertised bandwidth capacity per port.

```
exit-stats-end 2009-07-24 20:40:35 (86400 s)
exit-kibibytes-written 17=58902,23=9616,25=262579,40=9546,76=5789,
  80=681732,85=121859,143=7541,222=5133,300=9517,442=9634,443=12157,
  444=11692,690=5768,801=8100,850=9078,1000=6737,1015=57885,[...],
  other=15332199
exit-kibibytes-read 17=15,23=79,25=13221,40=7,76=2,80=1841879,85=926,
  143=1038,222=85,300=25,442=5,443=38435,444=94,690=8,801=9,850=12,
  1000=373,1015=68,[...],other=3035782
exit-streams-opened 17=12,23=88,25=141240,40=12,76=16,80=867896,
  85=2704,143=168,222=32,300=28,442=12,443=147348,444=92,690=4,
  801=16,850=16,1000=716,1015=56,[...],other=3165052
```

Fig. 3. Number of exiting bytes and opened streams as observed by an exit node

Statistics on traffic exiting the Tor network could be as sensitive as statistics on connecting client IP addresses. For one thing, the contents and targets of exiting traffic must not be disclosed, even without knowing which clients have sent or received these messages. After all, the majority of deployed application protocols do not encrypt traffic on the network. Similarly, the target address might reveal some information about the content and possible clients, especially if there are only few requests to that target. For another thing, exiting traffic, even in somewhat aggregated form, must not be usable to be combined with information on client IP addresses to correlate IP addresses to requests or responses. An adversary that runs an entry node or bridge should not gain additional information when combining her list of client IP addresses with exit traffic statistics.

Observations of exit traffic are processed in multiple steps to make them less useful for an adversary yet still useful for statistical analysis. In the first step, all information about the content of exiting traffic is discarded and only the meta data is preserved. Traffic content includes application headers and application content. While it is tempting from a statistical point of view to analyze at least the application headers, this analysis could cross the line from the pen register category (signaling and addressing) to the wiretap category (content) [3], so it is best avoided. Furthermore, the target address is discarded for statistics, as an adversary might draw conclusions about the content of requests. The remaining meta data that are used for statistical analysis are the target port and the number of outgoing and incoming bytes per connection.

In the next step, the exact times of observations are removed by accumulating observations over a measurement interval of 24 hours. Without this step, the information about an exiting connection including the target port number and number of transfered bytes might still give a hint on the content and/or client. Therefore, the number of written and read bytes as well as the number of opened streams are summed up per port. These sums not only make it impossible to restore timestamps, but they also hide single traffic patterns of incoming vs. outgoing bytes per connection. The intermediate result is a triple of written bytes, read bytes, and opened streams for every TCP port.

The third step of making these statistics less useful for an adversary is to report only the data for TCP ports that have seen a number of bytes exceeding a given threshold. All data for ports with data below this threshold are summed up and reported together.

Finally, in a fourth step, all observations are rounded up to conceal exact numbers of possibly only a few events. Bytes are rounded up to full KiB, and numbers of opened streams are rounded up to the next multiple of 4.

The results of the aggregation of exit traffic per port can be seen in Figure 3, which corresponds to the data shown in Table 1. The four lines describe when statistics were written and how long the measurement interval was, the number of written/read KiB, and the number of opened streams per port. For example, this exit node wrote 681732 KiB (666 MiB) and read 1841879 KiB (1.8 GiB) in 867896 streams on port 80. The threshold for a port being included in the statistics is 0.01% of all transfered bytes. All ports with fewer relayed bytes are summarized as port `other`. Again, the data format is described in the directory protocol specification document [12].

5.3 Guidelines

From these example cases as well as from earlier considerations we can derive a few guidelines. These guidelines shall apply to all future statistical analyses in the Tor network and hopefully to other anonymity systems as well.

Data minimalism. The first and most important guideline is that only the minimum amount of statistical data should be gathered to solve a given problem. The level of detail of measured data should be as small as possible.

Source aggregation. Possibly sensitive data should exist for as short a time as possible. Data should be aggregated at its source, including categorizing single events and memorizing category counts only, summing up event counts over large time frames, and being somewhat imprecise with respect to exact event counts.

Transparency. All algorithms to gather statistical data need to be documented and discussed publicly before deploying them. All measured statistical data should be made available to the public as a safeguard to not gather data that is too sensitive.

6 Discussion

This paper presents a case study of measuring two types of potentially sensitive data in the live Tor anonymity network: countries of connecting clients and exiting traffic by port. Both types of data have in common that they are sensitive in their raw form and need to be aggregated before being published and performing statistical analysis on them. We derived guidelines that can be useful for similar cases in the future when measuring sensitive data in anonymity networks. We hope that this paper starts a discussion on safely measuring network data in anonymity systems that serves both researchers studying anonymity networks and users relying on the protection that anonymity networks provide.

Acknowledgements

We thank Jacob Appelbaum and Jonathan Rippstein for measuring the presented statistics on their Tor relays.

References

1. Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, pages 303–320, August 2004.
2. Cynthia Dwork. Differential privacy. In *33rd International Colloquium on Automata, Languages and Programming (ICALP)*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12, Venice, Italy, July 2006. Springer.
3. Electronic Frontier Foundation. Tor: Legal FAQ for Tor server operators. `https://www.torproject.org/eff/tor-legal-faq.html`.
4. Jörg Lenhard, Karsten Loesing, and Guido Wirtz. Performance measurements of Tor hidden services in low-bandwidth access networks. In *Proceedings of the 7th International Conference on Applied Cryptography and Network Security (ACNS 09), Paris-Rocquencourt, France*, June 2009.
5. Karsten Loesing. Measuring the Tor network from public directory information. Technical report, 2nd Hot Topics in Privacy Enhancing Technologies (HotPETs 2009), Seattle, WA, USA, August 2009.
6. Karsten Loesing, Werner Sandmann, Christian Wilms, and Guido Wirtz. Performance measurements and statistics of Tor hidden services. In *Proceedings of the International Symposium on Applications and the Internet (SAINT 2008)*, Turku, Finland, July 2008. IEEE Computer Society.
7. Damon McCoy, Kevin Bauer, Dirk Grunwald, Tadayoshi Kohno, and Douglas Sicker. Response to tor study. `http://systems.cs.colorado.edu/mediawiki/index.php/Response_To_Tor_Study`.
8. Damon McCoy, Kevin Bauer, Dirk Grunwald, Tadayoshi Kohno, and Douglas Sicker. Shining light in dark places: Understanding the Tor network. In Nikita Borisov and Ian Goldberg, editors, *Proceedings of the Eighth Symposium on Privacy Enhancing Technologies (PETS 2008)*, volume 5134 of *Lecture Notes in Computer Science*, pages 63–76, Leuven, Belgium, July 2008. Springer.
9. Steven J. Murdoch and Piotr Zieliński. Sampled traffic analysis by internet-exchange-level adversaries. In Nikita Borisov and Philippe Golle, editors, *Proceedings of the Seventh Workshop on Privacy Enhancing Technologies (PET 2007)*, Ottawa, Canada, June 2007. Springer.
10. Lasse Øverlier and Paul Syverson. Locating hidden servers. In *Proceedings of the 2006 IEEE Symposium on Security and Privacy*. IEEE CS, May 2006.
11. The Tor Project. *Tor bridges specification*, 2009. `https://git.torproject.org/checkout/tor/master/doc/spec/bridges-spec.txt`.
12. The Tor Project. *Tor directory protocol, version 3*, 2009. `https://git.torproject.org/checkout/tor/master/doc/spec/dir-spec.txt`.
13. Rolf Wendolsky, Dominik Herrmann, and Hannes Federrath. Performance comparison of low-latency anonymisation services from a user perspective. In Nikita Borisov and Philippe Golle, editors, *Proceedings of the Seventh Symposium on Privacy Enhancing Technologies (PET 2007)*, volume 4776 of *Lecture Notes in Computer Science*, pages 233–253. Springer, June 2007.