Delft University of Technology, Bachelor Seminar of Computer Science and Engineering

Measuring Accessibility of Popular Websites While Using the I2P Anonymity Network

Ioana Paula Iacoban , Stefanie Roos
TU Delft

Abstract

Anonymity networks, such as The Invisible Internet Project, commonly known as I2P, enable privacy aware users to stay anonymous on the Internet and provide secure methods of communication, as well as multi-layered encryption. Despite the many innocent reasons users opt for online anonymity, these particular networks are censored at times, as they are associated with criminal activity. The goal of this paper is to measure to what extent I2P network users are being blocked by popular websites, and not, however, by governments or internet service providers. To establish this, we developed a web crawler which compares the responses to HTTP(S) GET requests sent anonymously, via I2P, and non-anonymously. Our results are based on the analysis of the received HTTP status codes, and on screenshots of the requested websites, to assess content blocking. This experiment shows that I2P users suffer from some form of blocking in 10.09% of cases. However, it should be noted that I2P faces certain bandwidth limitations and traffic congestion at the outproxy. This is a result of the fact that I2P was not designed with the intent of being a proxy to the Internet, but rather a self sustaining peer-to-peer network.

1 Introduction

Over the years, online user anonymity has become increasingly difficult to achieve, with the growth in popularity of web tracking techniques [1]. Tracking online user behaviour is heavily used in advertisement [1], law enforcement [2], compiling web analytics [3], and conducting user testing [4]. Having an anonymous identity over the Internet is desirable for various reasons, such as privacy concerns, bypassing censorship, or fear of retribution against whistle-blowers, unofficial leaks, and activists who do not believe in restrictions on information nor knowledge [5]. Anonymity networks, such as I2P (The Invisible Internet Project) and Tor (The Onion Router) [6], enable users to browse the web without being tracked and without revealing their identity to other network participants, such as a website provider or the internet service provider. However, anonymity networks are often being blocked when detected, due to their association with criminality. While blocking behaviour can originate from a specific website or service, in a significant amount of cases, censorship is imposed by governments and implemented by ISPs, in countries such as China [7, 8, 9], Iran [10], Pakistan [11], Russia [12], and Syria [13].

Measuring Internet censorship has been proved to be a challenging task, due to the fact that it varies over time and requires plenty of resources to assess. Burnett and Feamster [14] attempted a lightweight measurement of the Internet censorship by, having users request cross-origin resources while loading a web page. These requests were triggered by embedded HTML elements, such as images, scripts, hyperlinks or iframes, and establishing which requests get blocked, would account for the measurement. However, the study was discontinued, due to certain ethical concerns, since the experiment was deployed, through webmasters, on publicly available websites, and the clients (users) where not explicitly informed of the experiment taking place. Furthermore, Raman et al. [15] have created the Censored Planet platform to monitor censorship globally, by collecting and analyzing measurements from ongoing deployments of four remote measurement techniques (Augur, Satellite/Iris, Quack, and Hyperquack). Hence, network anomalies could be remotely detected, the most common censorship methods being shutdowns, DNS manipulation, IP-based blocking, and HTTP-layer interference [15]. Similar censorship observatory platforms include OONI [16], ICLab [17], UBICA [18], and CensMon [19].

Singh et al. [20] extensively studied the extent to which online service providers discriminate against Tor users, in their proceedings titled *Characterizing the Nature and Dynamics of Tor Exit Blocking*. According to the aforementioned research, the main problem Tor is facing is that users share their reputation. Thus, the malicious actions of a single user that lead to IP blacklisting, impact all the other users since Tor is a centralized network. The results showed that 88% of Tor relays are blacklisted, and 20% of all Alexa Top 500 website frontpage requests are discriminating against Tor users, with an increase of 3.9% and 7.5% in search and login functionalities respectively. The findings are based on email complaints sent to Tor relay operators,

blacklisting of Tor-related IP addresses, and measurements of server responses to Tor traffic, both synthetic (crawled) and user-driven.

Related research on the I2P network includes *Measuring I2P Censorship at a Global Scale* by Hoang, Doreen and Polychronakis [21], which focuses on how users are being impeded by censors to join the I2P network. The main techniques implemented to block new users from accessing the network, according to [21], are domain name blocking (DNS-based blocking, SNI-based blocking), as well as TCP packet injection. Measurements were made in 164 countries, and I2P blocking activities were detected in China, Iran, Oman, Qatar, and Kuwait, between the months of March and April, 2019 [21]. Moreover, Hoang et al. conducted an empirical study of the I2P network [22] by introducing new peers in the I2P network and crawling seed servers, in order to statistically approximate the size of the network, as well as its resistance to censorship. The study concluded that I2P can be censored by entering the network, identifying peers, and blocking them through means of blacklisting.

Nevertheless, our study does not focus on measuring the censorship established by governments, third parties, or by attacking the I2P network. The main questions this paper aims to answer are: *To what extent do websites block users accessing them using I2P? How frequent is blocking and which content does it affect?* To tackle the aforementioned questions, we developed a web crawler which identifies blocking behavior by comparing the responses to HTTP(S) GET requests sent anonymously, via I2P, and non-anonymously. Our findings show that, in 89.28% of instances, the websites were successfully retrieved, while 9.14% presented with partly inaccessible elements, and 1.58% were blocked. In addition, our experimental setup and results are extensively discussed in Section 4 and Section 5.

2 Background

2.1 The Invisible Internet Project

The Invisible Internet Project, commonly known as I2P, consists of a decentralized peer-to-peer network which aims to keep the identity of its users anonymous, by including additional layers of encryption to the sent messages. This communication technique is referred to as onion routing, where each layer of encryption is associated with a layer of an onion. However, I2P implements a variant of onion routing called garlic routing, the main difference being that garlic routing allows multiple messages to be encoded into one network packet (bundle), analogous to the bundling of garlic cloves into a garlic head. Additionally, unlike the onion routing implemented by Tor, which uses bidirectional communication channels, the garlic routing of I2P establishes unidirectional tunnels between the peers or routers in the network [22]. Currently I2P supports three transport layer protocols: NTCP (a Java New I/O (NIO) TCP transport), SSU (Secure Semireliable UDP), and NTCP2, a new version of NTCP. Each of them provide "a 'connection' paradigm, with authentication, flow control, acknowledgments and retransmission" [23].

The naming system, that allows peers (routers) to find each other and exchange messages, is integrated into the I2P distribution, external to the I2P router, while all hostnames are local [24]. The naming system consists of the following components: a local naming service, an HTTP proxy, HTTP host-add forms, HTTP jump services, the address book application, and the SusiDNS application [24]. The local naming service handles lookups and Base32 hostnames, while the HTTP proxy requests lookups from the router, and points the user to remote HTTP jump services, to assist with failed lookups. Additionally, HTTP host-add allows users to append hosts to their local hosts.txt file, and the address book application merges external host lists with the local list. Finally, SusiDNS is an application for address book configuration and viewing of the local host lists.

Upon entering the network, each router follows a bootstrap process in which it discovers other peers and creates inbound and outbound tunnels for the incoming and outgoing packets respectively. Each inbound tunnel consists of an inbound gateway which forwards incoming messages, through the inbound participants, to the inbound endpoint (the receiver situated at the end of the inbound tunnel). Similarly, the outbound tunnel carries outgoing messages from the outbound gateway (the sender), through the outbound participants, to the outbound endpoint, which ships the packets towards the inbound gateway of the recipient. Thus, if, for instance, Alice wants to communicate with Bob, the message will travel from Alice through her outbound tunnel, and the endpoint of the outbound tunnel will send the message to Bob's inbound tunnel, which will forward to message to Bob, as depicted in Figure 1. One important privacy related aspect to note is that Alice does not know Bob's address, but she does, however, know the address of Bob's inbound gateway from querying the network distributed database, netDb [25]. The netDb database implements a modified version of Kademlia distributed hash table, to securely distribute routing and contact information [26]. Hence, Alice and Bob can establish an end-to-end encrypted communication channel, consisting of four tunnels (one inbound, as well as one outbound tunnel, for both Alice and Bob), while staying anonymous. [22].

Moreover, if Alice wants to connect to the Internet, she could exit the I2P network via Bob, if Bob voluntarily runs an outproxy service. However, this is not encouraged by I2P due to privacy concerns: "I2P is primarily a hidden service network and outproxying is not an official function, nor is it advised. The privacy benefits you get from participating in the the I2P network come from remaining in the network and not accessing the internet." [26].

2.2 Web Crawlers

Web crawlers, sometimes referred to as spiders or spiderbots, are internet bots, generally used by search engines to index websites [27]. Crawling can be associated with a graph search problem, where each node is a website

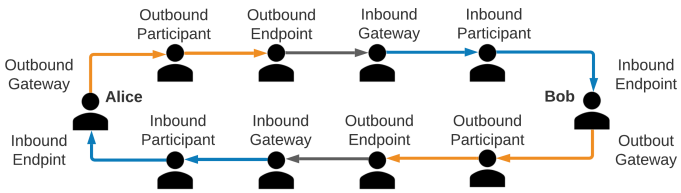

Figure 1: I2P Tunnel Routing

and each edge is a hyperlink, leading to another website [27]. At the start of a crawl, the spider is provided with an initial website frontier which constitutes of a set of URLs the crawler visits. Following a successful request, the spider extracts other hyperlinks from the current website and appends them to the frontier, to be retrieved subsequently.

One of the main challenges of web crawling is avoiding spider traps, which can be set intentionally by websites to block internet bots, or unintentionally [27]. A spider trap causes the crawler to get caught in an infinite loop, by re-requesting the same website indefinitely, and therefore terminating progress. To mitigate this issue, spiders are commonly given a limit of a specific number of addresses per domain. As mentioned by Ro, Han and Im in [28], other means of blocking crawlers include HTTP header information filtering, access pattern-based anticrawling, access frequency-based anticrawling, and CAPTCHAs (Completely Automated Public Turing test to tell Computers and Humans Apart). HTTP header information filtering can be used to differentiate users from bots by inspecting the user agent header, however, this field can be set to mimic a browser user agent [28]. Access pattern-based anticrawling recognizes crawler by assessing the pattern of the requested subpages of a website, and flags unusual behaviour [28]. Frequency-based anticrawling identifies crawlers by the frequency of requests, and blocks clients that exceed a specific threshold within a set time frame [28]. Another method that can exclude crawlers from specific addresses is the Robot Exclusion Protocol (robots.txt), through which webmasters can set explicit crawling rules, such as what sections, files, and subdomains should not be accessed by a spider [27]. Unlike the other blocking methods, robots.txt can be considered as an agreement between the website and the spider. In addition, the spider could technically disregard the rules specified in robots.txt, however this would be classified as malicious behaviour.

2.3 Parties implementing blocking techniques

We have identified four parties that could implement blocking techniques, namely governments, ISPs, websites, and third parties hosting website cross-origin resources. First of all, government authorities could impose web censorship, which is enforced by ISPs, as previously mentioned in the introduction. Moreover, ISPs could cause the occurrence of blocking without any governmental restriction in place, by simply dropping packets and connections, which is known as network bias or net bias [29]. Additionally, websites could be implementing blocking techniques, either against users of anonymity networks or against crawlers. In the aforementioned scenarios, the requested websites would be inaccessible, either by not responding to the request or by transmitting error messages, blank pages, blocked pages, or CAPTCHA challenges. Lastly, if a website is successfully retrieved, forms of blocking could still occur, originating form third parties hosting cross-origin resources used by the website, causing the inaccessibility of embedded elements and content.

3 Methodology

In order to evaluate to what extent websites block users accessing them using I2P, we first establish how frequently blocking behaviour is encountered and which content it affects by means of statistical data. Hence, we decide to conduct an experiment involving a web crawler that accesses popular websites and tracks successful, as well as failed requests. Each website is requested by the crawler through an anonymous request, via I2P, as well as a non-anonymous (control) request, and the responses are later compared, as illustrated in Figure 2. Moreover, since partial blocking can occur in the form of unavailability of specific subpages of certain websites, the crawler follows up to three hyperlinks on each successfully retrieved homepage of the specified website set, to account for this scenario. Following the crawl, we also investigate whether there are any specific categories of websites that are more prone to block I2P users. The technical implementation of the crawler and the experiment setup are explained in further detail in Section 4.

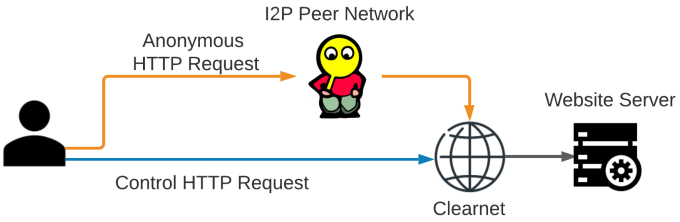

Figure 2: Experimental Setup

The frequency of blocking behaviour is assessed on HTTP status codes by running the crawler over a period of three to four weeks. Furthermore, evaluation of the blocking of website content is performed by comparing screenshots of websites requested anonymously, through I2P, and non-anonymously. In addition, analysing website screenshots reduces the number of false negative instances, since a website could respond with an HTTP 200 OK code, even though the received web page is inaccessible [20]. We consider a website to be blocked if the control request receives a successful HTTP status code, while the I2P request receives a non-successful HTTP status code. Additionally, if the rendered websites of the successful HTTP(S) GET requests show an HTTP error code, blank page, block page or a CAPTCHA challenge, the specific website would be considered as being blocked, unless the control request produces the same result.

In terms of content blocking, there are mainly two outcomes that could occur following an HTTP(S) GET request: the page is entirely blocked when a blank page, block page, CAPTCHA, or an error message is received, or partially blocked when certain website elements are not accessible. In the case of partial blocking, the content that are most expected to be impacted are media content, such as images and videos, as well as elements which require cross-origin resources including scripts, style sheets, or other media. This is due to the fact that the aforementioned content is often distributed in multiple locations, that might have other policies regarding information retrieval than the main website host. Furthermore, websites containing images and video require significantly more bandwidth to be successfully delivered and fully rendered. Hence, bandwidth is an important factor to this experiment, since I2P is restricted in this regard, as it was not designed to act as a proxy to the Internet, but rather as a self sustained, independent network. Therefore, the support for accessing content outside the network is limited [30].

The selected dataset for this experiment is the 500 most popular websites, ranked on domain authority by Moz [31]. Although most related research uses the Alexa Top Sites, we have selected the ranking made by Moz, because we expect the experimental results on the Alexa Top Sites to produce a higher false positive rate. This is due to the fact that Alexa Top Sites ranks web pages according to the average number of daily visitors [32]. As a result, the list contains websites that require authentication, which, when requested, trigger an HTTP 403 Forbidden error. Moreover, the Moz top 500 ranking would be a more fitting list for this experiment, since it mostly contains domains that are not bandwidth intensive.

4 Measuring Accessibility of Popular Websites

4.1 Crawling popular websites

To tackle the crawling challenges discussed in Section 2, namely HTTP header information filtering, access pattern-based anticrawling, access frequency-based anticrawling, and CAPTCHAs, we implemented a simple web crawler design, which starts with a frontier of 500 popular websites, from the Moz ranking [31], and extracts up to three hyperlinks form each successfully retrieved website. The selected hyperlinks are the first three that are encountered, with the condition that they are different from the one that lead to that respective main website. Hence, by limiting the the depth of the crawl, we avoid spider traps. It should be noted that the list of selected hyperlinks could change between different runs of the experiment, if the content of the respective homepage changes. Furthermore, to avoid bot detection while crawling, the HTTP user agent header is custom set, the number of concurrent requests per domain and per IP are set to 1, and the crawler is deployed responsibly to not overload servers with requests. In addition, the crawler is instructed to obey the robots.txt rules, and each request resulting in no response after 180 seconds is reattempted no more than two times, to avoid bot detection, as well as overloading the host server, which may respond in return with HTTP error code 429 Too Many Requests. Moreover, to ensure that the results are reliable, the HTTP cache was disabled and cookies where enabled during the crawl.

The crawler design is implemented in Python, within the Scrapy framework. Scrapy provides a particularly time and resource efficient way of accessing websites through asynchronous HTTP(S) GET requests, and supports adaptable settings. The initial development iteration of the spider covered requesting the 500 websites from the Moz ranking, and logging the timestamped HTTP status code responses to a file. Each website is requested twice: once through a control non-I2P request, and once through the I2P proxy. The crawler was later extended to follow up to three hyperlinks found on each of the websites from the initial frontier. The runtime of our Scrapy crawler, with an initial frontier 500 websites, requires approximately 15 minutes, and between 40 to 45 minutes if the crawler is programmed to follow three hyperlinks from the initial frontier of URLs. Therefore, with the low latency of the crawler, the effects of the I2P dynamic network topology, and of the network bias, caused by the ISP dropping network packets [29], on the results is minimized.

4.2 Evaluating blocking behaviour

Blocking behavior is evaluated twofold: based on frequency and on the affected content. The frequency of blocking will be assessed according to the received HTTP status code of the requests. Successful requests are confirmed at a later stage with screenshots of the respective websites. Additionally, with screenshots, any content blocking such as images, scripts and embedded elements, can be identified. The flowchart in Figure 3 depicts the how blocking is evaluated by this experiment.

Recording screenshots of websites poses a number of obstacles to the current Scrapy crawler setup. Firstly, Scrapy crawls by creating HTTP(S) GET requests without a browser, which is required in order to execute JavaScript code from the websites, especially dynamic websites. Although Scrapy supports Splash and Selenium headless browsers (browsers without graphical user interface used for automated scripts) as plugins, this creates a complication. In order to be able to screenshot websites, the full page must be retrieved and the code must be executed, which introduces a significant overhead for each request. Furthermore, Scrapy sends the requests asynchronously for efficiency purposes, however, not being able to synchronize the control and the I2P requests could affect the reliability of the results. If there is a notable difference in time between the two requests, several factors could interfere with the experiment. Among other considerations, the respective website could change in terms of content (prevalent in news websites), the host servers could become temporarily unavailable for one of the requests but not the other, and network issues or network bias could by encountered within the ISP network (locally or at the I2P outproxy). Furthermore, changes in the I2P network

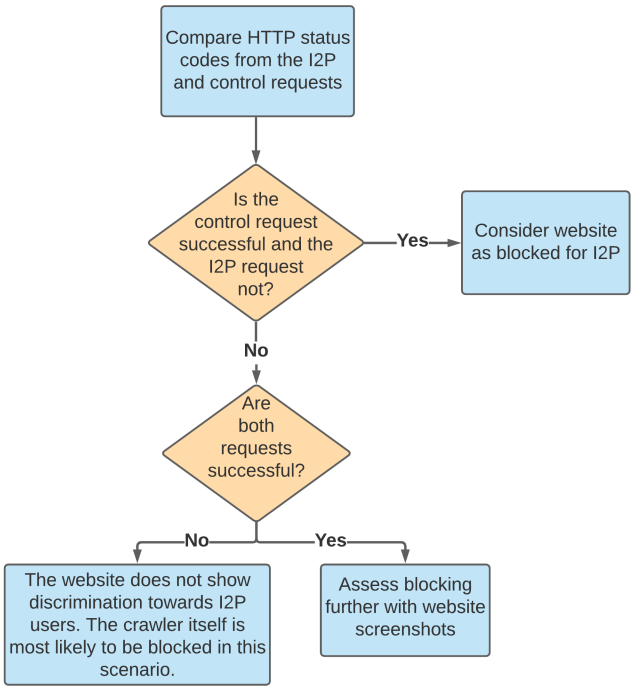

Figure 3: Assessment of Blocking

topology are highly likely to occur, which would influence packet routing, outproxy availability and, as a result, the I2P network performance. Thus, synchronization is a highly important aspect for lengthy crawls. Additionally, crawling while recording two screenshots of approximately 1500 hyperlinks is estimated to require several days, considering the fact that I2P trades performance for anonymity.

To ensure that the control and I2P requests are synchronized for the content blocking evaluation through screenshots, we decided on an approach similar to the one Singh et al. [20] used to study Tor exit blocking with a Selenium WebDriver, which is used for this stage of our experiment. The websites successfully retrieved by the Scrapy spider, through both the control and I2P requests, are the input dataset for the Chrome WebDriver. As, each website is rendered and screenshot, loading times differ significantly between the control request and the I2P request. A website requested through I2P could take several minutes to fully render. Hence, the request timeout is left as the default browser timeout (which is 300 seconds according to the Chromium codebase [33]), in order to not interrupt the loading process and bias the results.

Moreover, Singh et al. [20] employ pHashing (perceptual hashing) to assess the dissimilarity between pairs of images by calculating their pHash distance similarity score: "we classified distances < 0.40 as 'non-discrimination' and distances > 0.75 as 'discrimination'. Instances having pHash distances in the 0.40 to 0.75 range were manually inspected and tagged." [20, p. 334]. Essentially, the pHash is analogous to a fingerprint that is generally applied for image content authentication [34]. We make use of the pHashing technique for this experiment as well, to identify which pairs of images present with differences. We label the pairs, according to the type of blocking present, as follows: not blocked, partly blocked, blocked. The images that produce an identical pHash are labeled as not blocked, while the rest are manually classified accordingly.

5 Quantifying blocking behaviour

From May 10th until June 7th, 2021, we measured blocking behaviour on the top 500 websites from the Moz ranking [31], by means of HTTP status codes and screenshots, as presented in the previous section. Out of the 500 websites from our dataset, 7 web pages consistently caused DNS lookup errors at the time of measurement, and, as a result, they were excluded from the crawl. Since the DNS lookup errors occured on the I2P request, as well as the control request, and upon manual inspection, the most likely explanation is that the respective websites went offline. During the crawl of the Scrapy spider, the adjustable parameters were set as follows: the HTTP cache was disabled, cookies were enabled, the number of concurrent requests per domain and per IP was set to 1, the timeout period was 180 seconds, and each request that resulted in a timeout was reattempted at most two times. Furthermore, during the run of the Selenium Chrome WebDriver, cookies were automatically accepted using the *I don't care about cookies* Chrome extension. The measurements have been conducted in The Netherlands, and the selected I2P outproxy was false.i2p (Norway). Since outproxying is a service that is voluntarily offered by peers in the network, and it is not officially supported by I2P, there are hardly any choices to be made in terms of selecting the outproxy. As a result, false.i2p was set as outproxy, which is also I2P's default option.

In the following section, we discuss how frequently blocking occurred and what type of content it affected. Upon inspecting the preliminary results of the experiment, we estimated that the ratio of successful I2P requests, with no signs of blocking behaviour, is between 85% and 100% with a confidence interval of 95%. We consider an I2P request to be blocked with respect to the control request, when the website is successfully retrieved by the control, but not by the I2P request. The instances when the control request fails and the I2P request succeeds are classified as not blocked.

5.1 Frequency of blocking

The frequency of the HTTP status codes, recorded by our preliminary Scrapy crawl, can be visualized in Table 1 for the control requests, and in Table 2 for the I2P requests. The aforementioned tables encompass the received responses form 493 websites from our data set, excluding the data points causing DNS lookup errors. Thus far, results show that 96.35% of home pages and 97.25% of subpages were successfully retrieved through the control requests, while 93.1% of home pages and 97.1% of subpages were successful in the case of I2P requests. Hence, 3.59% of the Scrapy HTTP GET requests show discrimination towards I2P clients. Furthermore, the HTTP status codes accounting for the blocking behaviour towards I2P are 403 Forbidden, 404 Not found, 410

Gone, and 503 Service Unavailable. In addition to the HTTP error codes, 0.81% of I2P requests resulted in a timeout.

HTTP Status Code	Proportion
200 OK	96.35%
400 Bad Request	1.22%
403 Forbidden	1.01%
404 Not Found	0.81%
TCP Timeout	0.2%
Other	0.41%

Table 1: Control Requests Received HTTP Status Codes

HTTP Status Code	Proportion
200 OK	93.1%
400 Bad Request	1.22%
403 Forbidden	2.43%
404 Not Found	1.01%
410 Gone	0.2%
503 Service Unavailable	0.81%
Timeout Error	0.41%
Other	0.81%

Table 2: I2P Requests Received HTTP Status Codes

The successfully retrieved websites, which amount to 1502 data points, are further evaluated for blocking behaviour with screenshots using a Selenium Chrome WebDriver. For the remainder of this section, the statistics are compiled collectively, including the home pages, as well as the crawled subpages. Subsequently to the WebDriver crawl, 89.28% of requests are completely successful, while 9.14% present with some form of blocking, and 1.58% are blocked with respect to the control request as depicted in Figure 4.

We consider a request to be blocked when the received response contains an error code, an error message, a blocked page, a blank page, or a CAPTCHA challenge. Additionally, a request is considered to be partly blocked when the rendered website presents with missing items, such as images or interactive elements. It should be noted that Figure 4 excludes the case when both the I2P and the control requests are unsuccessful. Furthermore, we noticed some instances when the control request is blocked, but the I2P request is successful. This was mainly due to geo-blocking because of GDPR (General Data Protection Regulation) regulations as the location for control request is The Netherlands, while the I2P outproxy operates in Norway, which is not a member state of the European Union at the time of writing. This particular situation and all other cases where blocking or partial blocking was identified in the control request, but not the I2P request, were categorized as not blocked.

To sum up, we found that out of the 1520 data points, 1367 consist of successful I2P requests, that represents 89.28% of data points, which corresponds to our estimated interval of 85% - 100%.

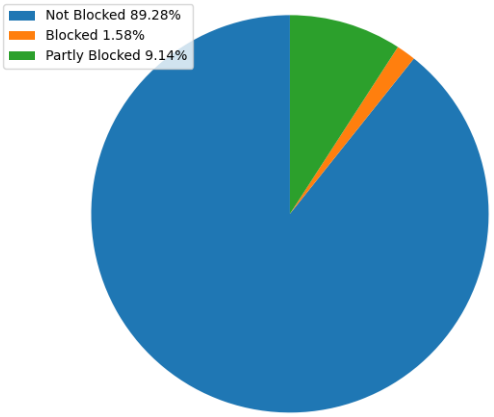

Figure 4: Proportion of successful, blocked, and partly blocked I2P requests with respect to the control requests

5.2 Content blocking

To answer our research sub-question regarding content blocking, we refer to Figure 5, which shows what type of content is blocked or inaccessible, when comparing screenshots of the I2P and control request. The analysis of the recorded screenshots indicates that, in the case of partial blocking, the most frequently affected type of content are images, which occurred in 43.65% of partial blocking instances. The second most common blocked type of content we defined as interactive elements, which was present in 26.98% of cases, and 15.08% of cases presented with both inaccessible images and interactive elements. Under the category of interactive elements we included buttons, forms or form fields, search bars, menus, calendars, and other components that can trigger an action, as a response to a user event. In addition, 9.52% of the partly blocked websites were rendered with missing scripts, such as CSS and JavaScript. Lastly, the other information slice of the pie chart from Figure 5, in proportion of 4.76%, consists of instances that presented with missing textual information and/or cross-origin embedded elements.

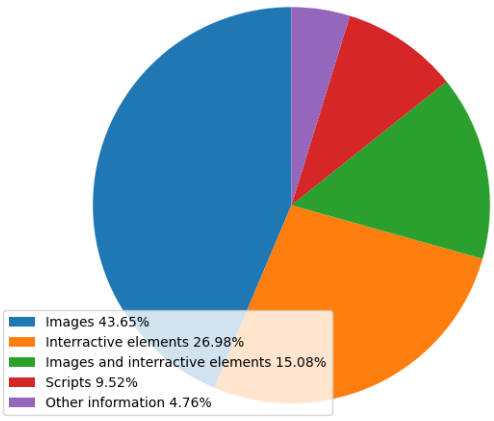

Figure 5: Web content affected by partial blocking

5.3 The reliability of the I2P network

One caveat has to be mentioned about the previously mentioned results, which relates to the reliability of the I2P network and outproxy service. While recording the website screenshots with the Selenium WebDrive, two error messages were encountered quite frequently: "taking too long to respond" (timeout) error and "Outproxy Not Found". As a result, a significant number of websites had to be re-requested several times, until they where successfully retrieved. The "Outproxy Not Found" error message makes it clear that our selected outproxy is temporarily not available, or it is experiencing traffic congestion, and therefore it drops connections. Hence, the inaccessibility is caused by I2P network technical issues. However, in the case of timeouts, it is not clear whether the error occurs as a result of a blocking behaviour instance, or an I2P network issue. Thus, we decided re-requesting timed out requests is necessary to minimize the false positive rate. Furthermore, some instances of partial blocking could be, in fact, caused by I2P network issues, as opposed to being an occurrence of blocking behaviour.

To get a sense of how frequently I2P network outages arise, we conducted an additional experiment. From our initial website dataset, we selected a subset of 15 websites which we requested through the Selenium Chrome WebDriver, via a control, as well as an I2P request, and recorded a screenshot of each, just as in the previous experiment. The procedure was repeated twice per day, over the course of five days. The selected websites for this short experiment were previously confirmed to not be completely blocked. Moreover, the selection was done with diversity in mind, therefore we included websites from the following categories: social networks, news, services, companies, finance, search engines, cloud storage, e-commerce, and education. Table 3 encapsulates the number of requests that resulted in a timeout or an "Outproxy Not Found" error, for each run of the procedure. On average 7.3 out of 15 requests time out, or fail due to an outproxy unavailability. Based on this average result, there is a 48.66% chance that a website requested via I2P will not be rendered. It should be noted that this is not the case for simple HTTP(S) GET request that only record status codes, since there is no content handling and therefore requires significantly less bandwidth. Furthermore, in the previous experiment, timeouts and outproxy availability issues were addressed by re-requesting the respective pages, to reduce the false positive rate as much as possible. Hence, the significant difference in ratio between the I2P network failures probability, and the blocking behaviour measured in the previous experiment.

Day	1	2	3	4	5
Run 1	9/15	8/15	7/15	8/15	10/15
Run 2	7/15	2/15	8/15	7/15	7/15

Table 3: Non-successful I2P requests, reliability experiment

5.4 Websites blocking I2P

The websites that showed blocking behaviour towards I2P users, either through blocking or partial blocking, are categorized in Table 4. The websites are categorized using the McAfee Customer URL Ticketing System service. The results indicate that General News is the most frequent website category to present blocking, followed by Blogs/Wiki, Business, Internet Services, Personal Network Storage, Education/Reference, and others. However, as discussed in the previous subsection, the results could be biased towards websites that encode large amounts of information, typically images, video, or other media, which is common among the General News and Blogs/Wiki, as well as other categories. Furthermore, General News sites also contain plenty of embedded elements, and their inaccessibility may be caused by either the news site, or the host site. These statistics could also indicate the categories of websites that are likely to cause loading issues due to I2P's limitations. Therefore, we cannot draw a precise conclusion as to whether a certain website category is more prone to implement blocking techniques.

Category	Percent
General News	23.36%
Blogs/Wiki	10.22%
Business	8.76%
Internet Services	8.03%
Personal Network Storage	5.11%
Education/Reference	5.11%
Interactive Web Applications	4.38%
Non-Profit/Advocacy/NGO	3.65%
Portal Sites	2.92%
Search Engines	2.92%
Finance/Banking	2.92%
Media Sharing	2.92%
Professional Networking	2.19%
Software/Hardware	2.19%
Sports	2.19%
Government/Military	2.19%
Travel	1.46%
Marketing/Merchandising	1.46%
Web Mail	1.46%
Entertainment	1.46%
Games	1.46%
Other	4.38%

Table 4: Categories of websites exhibiting blocking behaviour

6 Responsible Research

Since our research focuses on a network measurement achieved through web crawling while using the I2P anonymity network, several ethical aspects must be considered. Firstly, to ensure the privacy of the I2P peers, no sensitive data about the I2P network is recorded, such as IP addresses, established tunnels, or network statistics (stats.i2p). Moreover, we followed the I2P academic research

guidelines [35] to responsibly conduct the experiment.

Secondly, the experiment must not hinder the performance of the I2P network, and negatively affect the experience of other peers in the network. As previously discussed, I2P is mostly used for internal use such as hidden services, and less often as a proxy to the Internet. As a result, very few outproxy services are running and they are made available on a peer voluntary basis, not maintained by I2P staff. Thus, traffic congestion could occur at the outproxy, causing timeouts and website loading problems. To mitigate this issue, the experiment is performed in stages, in order to distribute the web traffic.

Thirdly, web crawling must be performed responsibly such that website servers are not overloaded with requests, and the Robots Exclusion Standard is respected. The Robots Exclusion Standard is a protocol through which webmasters provide the scraping and crawling rules of a particular website, encoded in the robots.txt file on the host server. Additionally, in case of timeouts, the crawler stops requesting a website after three attempts that result in no response.

Should the reader be interested in repeating the experiment and reproducing the results, the scripts used for crawling, image processing, as well as compiling the statistics from this paper, can be found in the *spiderbot* GitHub repository [36]. However, results may vary with time as websites could undergo changes such as their domain name, updates in their policies regarding crawling, or their blocking behaviour. Furthermore, geographical location is another factor which can influence the outcome of the experiment, because of geo-blocking techniques. The results presented in the previous section were achieved by crawling the web from The Netherlands.

7 Discussion and Future Work

From May 10th until June 7th, 2021, we measured blocking behaviour on 500 popular websites, by means of HTTP status codes and screenshots, using web crawling. We found that blocking behaviour towards I2P users is not prevalent among popular websites, since, in 89.28% of instances, the websites were successfully retrieved, while 9.14% presented with partly inaccessible elements, and 1.58% were blocked. In the case of partial blocking, the affected content included images, scripts (CSS and JavaScript), interactive elements (buttons, forms, menus, etc.), textual information, and embedded cross-origin elements. To draw a parallel, we present the blocking behaviour faced by the Tor network from the Alexa Top 1000 websites, measured in a similar experiment [37]. The findings show that 25.8% of homepages requested through Tor are blocked, with an increase of 8.3%, if three subpages per homepage are additionally requested.

However, the I2P network did not prove to be a reliable proxy to the Internet. As we showed in our follow-up experiment, on average 7.3 out of 15 HTTP GET requests, I2P network related issues caused the requests to fail. Hence, confirming I2P's claim that the network is intended to be used for internal services, and not as a proxy to browse the web: "I2P is primarily a hidden service network and outproxying is not an official function, nor is it advised. The privacy benefits you get from participating in the the I2P network come from remaining in the network and not accessing the internet." [26]. This aspect might have introduced a bias in our results, along with the dynamic I2P network topology, geo-blocking, network traffic, and other network related issues. Additionally, websites' temporal unavailability and the day of the week and time of day could influence the results as well, likely due to network traffic. Moreover, there is also the scenario that the I2P network is being attacked, which would result in network unavailability, however it is highly unlikely to have occurred during our experiment, since the crawler was deployed during a relatively short time frame. Furthermore, I2P is estimated to have 32K active users on a daily basis [22], compared to Tor's 8 million daily users [38], making I2P a less likely target for attacks.

One could obtain more reliable results by voluntarily running an outproxy as a secondary I2P router, and repeating the experiment with the custom set outproxy. Hence, mitigating I2P network traffic issues and outproxy outages. In addition, enlarging the website dataset would contribute to an improved and more reliable accessibility measurement. Furthermore, an interesting scenario worth exploring would be the impact of world-wide (political) events on the accessibility of popular websites requested through the I2P network.

Acknowledgements

We would like to thank Francine Biazin do Nascimento [39], Jurgen Mulder [40], Anant Pingle [37], and Willemijn Tutuarima [41], who conducted similar experiments, measuring the accessibility of popular websites accessed through Tor, the Java Anon Proxy, and VPNs. Related literature, experimental setups, common challenges, and potential solutions were discussed among the peer group.

References

[1] Tatiana Ermakova et al. "Web Tracking – A Literature Review on the State of Research". In: *Proceedings on Privacy Enhancing Technologies* (2018). URL: https : / / www . ftc . gov / system / files / documents / public_comments/2016/10/00057-129178.pdf.

[2] Andrea Tundis, Humayun Kaleem, and Max Muhlhauser. "Tracking Criminal Events through IoT Devices and an Edge Computing Approach". In: *2019 28th International Conference on Computer Communication and Networks (ICCCN)*. 2019, pp. 1–6.

[3] Samantha Kleinberg and Bud Mishra. "Psst: A Web-Based System for Tracking Political Statements". In: *Proceedings of the 17th International Conference on World Wide Web*. WWW '08. Beijing, China: Association for Computing Machinery, 2008, pp. 1143–1144.

URL: https://doi-org.tudelft.idm.oclc.org/10.1145/1367497.1367697.

[4] Richard Atterer, Monika Friedemann, and Albrecht Schmidt. "Knowing the user's every move: User activity tracking for website usability evaluation and implicit interaction". In: Jan. 2006, pp. 203–212.

[5] D. Serebryakov. "Anonymity networks". In: (2015). URL: http://cryptowiki.net/index.php?title=Anonymity_networks.

[6] Linda Lee et al. "Tor's Usability for Censorship Circumvention". In: *Proceedings on Privacy Enhancing Technologies* (2017). URL: https://www.ftc.gov/system/files/documents/public_comments/2016/10/00057-129178.pdf.

[7] Roya Ensafi et al. "Analyzing the Great Firewall of China Over Space and Time". In: *Proceedings on Privacy Enhancing Technologies* 1 (Apr. 2015).

[8] Xueyang Xu, Z. Morley Mao, and J. Alex Halderman. "Internet Censorship in China: Where Does the Filtering Occur?" In: *Passive and Active Measurement*. Ed. by Neil Spring and George F. Riley. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 133–142.

[9] "How the Great Firewall of China is Blocking Tor". In: *2nd USENIX Workshop on Free and Open Communications on the Internet (FOCI 12)*. Bellevue, WA: USENIX Association, Aug. 2012. URL: https://www.usenix.org/conference/foci12/workshop-program/presentation/Winter.

[10] Simurgh Aryan, Homa Aryan, and J. Alex Halderman. "Internet Censorship in Iran: A First Look". In: *3rd USENIX Workshop on Free and Open Communications on the Internet (FOCI 13)*. Washington, D.C.: USENIX Association, Aug. 2013. URL: https://www.usenix.org/conference/foci13/workshop-program/presentation/aryan.

[11] Zubair Nabi. "The Anatomy of Web Censorship in Pakistan". In: *3rd USENIX Workshop on Free and Open Communications on the Internet (FOCI 13)*. Washington, D.C.: USENIX Association, Aug. 2013. URL: https://www.usenix.org/conference/foci13/workshop-program/presentation/nabi.

[12] Reethika Ramesh et al. "Decentralized control: a case study of Russia". English. In: *Proceedings, 2020 Network and Distributed System Security Symposium*. Network and Distributed Systems Security Symposium 2020 , NDSS 2020 ; Conference date: 23-02-2020 Through 26-02-2020. Internet Society, 2020, pp. 1–18.

[13] Abdelberi Chaabane et al. *Censorship in the Wild: Analyzing Internet Filtering in Syria*. 2014. arXiv: 1402.3401 [cs.CY].

[14] Sam Burnett and Nick Feamster. "Encore: Lightweight Measurement of Web Censorship with Cross-Origin Requests". In: *ACM SIGCOMM Computer Communication Review* (2015). URL: https://dl.acm.org/doi/pdf/10.1145/2785956.2787485.

[15] Ram Sundara Raman et al. "Censored Planet: An Internet-wide, Longitudinal Censorship Observatory". In: (2020). URL: https://dl.acm.org/doi/pdf/10.1145/3372297.3417883.

[16] A. Filastò and J. Appelbaum. "OONI : Open Observatory of Network Interference". In: *2nd USENIX Workshop on Free and Open Communications on the Internet, FOCI '12, Bellevue, WA, USA, August 6, 2012*. 2012.

[17] Arian Akhavan Niaki et al. "ICLab: A Global, Longitudinal Internet Censorship Measurement Platform". In: *CoRR* abs/1907.04245 (2019). URL: http://arxiv.org/abs/1907.04245.

[18] Giuseppe Aceto et al. "Monitoring Internet Censorship with UBICA". In: *Traffic Monitoring and Analysis*. Ed. by Moritz Steiner, Pere Barlet-Ros, and Olivier Bonaventure. Cham: Springer International Publishing, 2015, pp. 143–157. ISBN: 978-3-319-17172-2.

[19] Elias Athanasopoulos, Sotiris Ioannidis, and Andreas Sfakianakis. "CensMon: A Web Censorship Monitor". In: *USENIX Workshop on Free and Open Communications on the Internet (FOCI 11)*. San Francisco, CA: USENIX Association, Aug. 2011. URL: https://www.usenix.org/conference/foci11/censmon-web-censorship-monitor.

[20] Rachee Singh et al. "Characterizing the Nature and Dynamics of Tor Exit Blocking". In: *Proceedings of the 26th USENIX Conference on Security Symposium*. SEC'17. Vancouver, BC, Canada: USENIX Association, 2017, pp. 325–341.

[21] Nguyen Phong Hoang, Sadie Doreen, and Michalis Polychronakis. "Measuring I2P Censorship at a Global Scale". In: Aug. 2019.

[22] Nguyen Phong Hoang et al. "An Empirical Study of the I2P Anonymity Network and its Censorship Resistance". In: *Internet Measurement Conference (IMC'18)* (2018). URL: https://arxiv.org/pdf/1809.09086.pdf.

[23] I2P Official Homepage. *Transport Overview*. URL: https://geti2p.net/en/docs/transport.

[24] I2P Official Homepage. *Naming and Address Book*. URL: https://geti2p.net/en/docs/naming#base32.

[25] I2P Official Homepage. *The Network Database*. URL: https://geti2p.net/en/docs/how/network-database.

[26] I2P Official Homepage. *Intro*. URL: https://geti2p.net/en/about/intro.

[27] Christopher Olston and Marc Najork. "Web Crawling". In: *Foundations and Trends® in Information Retrieval* 4.3 (2010), pp. 175–246. ISSN: 1554-0669. URL: http://dx.doi.org/10.1561/1500000017.

[28] Inwoo Ro, Joong Han, and Eul Gyu Im. "Detection Method for Distributed Web-Crawlers: A Long-Tail Threshold Model". In: *Security and Communication Networks* 2018 (Dec. 2018), pp. 1–7.

[29] Rob Frieden. "Network Neutrality or Bias? - Handicapping the Odds for a Tiered and Branded Internet". In: Sept. 2006.

[30] I2P Official Homepage. *The Frequently Asked Questions*. URL: https://geti2p.net/en/faq.

[31] Moz Official Homepage. *The Moz Top 500 Websites*. URL: https://moz.com/top500.

[32] Alexa Official Homepage. *The top 500 sites on the web*. URL: https://www.alexa.com/topsites.

[33] The Chromium Authors. *Chromium codebase*. 2012. URL: https : / / source . chromium . org / chromium / chromium/src/+/master:net/socket/client_socket_pool.cc;l=25;bpv=0;bpt=1.

[34] Li Weng and Bart Preneel. "A Secure Perceptual Hash Algorithm for Image Content Authentication". In: Oct. 2011, pp. 108–121.

[35] I2P Official Homepage. *I2P Academic Research Guidelines*. URL: https://geti2p.net/en/research.

[36] I.P. Iacoban. *spiderbot*. URL: https : / / github . com / iiacoban42/spiderbot.

[37] Anant Pingle. "Measuring Accessibility of Popular Websites While Using TOR". In: 2021.

[38] Akshaya Mani et al. "Understanding Tor Usage with Privacy-Preserving Measurement". In: *Proceedings of the Internet Measurement Conference 2018*. IMC '18. Boston, MA, USA: Association for Computing Machinery, 2018, pp. 175–187. ISBN: 9781450356190. URL: https://doi.org/10.1145/3278532.3278549.

[39] Francine Biazin do Nascimento. "Measuring the Accessibility of Popular Websites While Using Mullvad VPN". In: 2021.

[40] Jurgen Mulder. "Measuring the Blocking of AN.ON Users by Popular Websites Through Web Scraping". In: 2021.

[41] Willemijn Tutuarima. "Measuring Accessibility of Popular Websites When Using ProtonVPN". In: 2021.