

Sociotechnical Aspects of Tor Relay Rejection

Jules Dejaeghere*
University of Namur
Namur, Belgium
jules.dejaeghere@unamur.be

Lionel Goffaux*
University of Namur
Namur, Belgium
lionel.goffaux@unamur.be

Pierre Luycx
University of Namur
Namur, Belgium
pierre.luycx@unamur.be

Hosam Elkoulak
University of Namur
Namur, Belgium
hosam.elkoulak@unamur.be

Florentin Rochet
University of Namur
Namur, Belgium
florentin.rochet@unamur.be

Abstract

In 2019, the Tor Project enforced an end-of-life (EoL) policy for Tor versions, leading to the rejection of outdated relays, amounting to a notable fraction of consensus weight. While this policy aids network maintenance, reduces backporting efforts, and shortens vulnerability exposure, its sociotechnical implications remain unstudied.

A user study ($N = 26$) reveals that relay operators, though not universally aware of the EoL policy, generally view it favorably. Operational practices vary, occasionally excluding newly installed relays from the network.

Network simulations, grounded in historical data, assess the policy's immediate impact on Tor clients against common adversaries. Results indicate a marginal adversarial advantage, with network churn (i.e., relays entering and exiting) exerting a more pronounced effect on user anonymity. Security metrics are introduced to evaluate relay contributions against two adversary models, enabling ranking by individual utility and security. Analysis of four exclusion rounds shows that a minority of rejected relays typically account for over 50% of the security provided by all excluded relays.

Recommendations for EoL policy implementation are proposed to mitigate potential drawbacks.

CCS Concepts

• **Security and privacy** → Social aspects of security and privacy; • **Networks** → Network privacy and anonymity.

Keywords

Tor; Software release life cycle; Anonymity; Tor relay; Internet exchange point (IXP); Autonomous system (AS)

Reference Format:

Jules Dejaeghere, Lionel Goffaux, Pierre Luycx, Hosam Elkoulak, and Florentin Rochet. 2026. Sociotechnical Aspects of Tor Relay Rejection. arXiv version, 17 pages.

*Shared first author.

1 Introduction

The Tor network [14] is a critical privacy-enhancing technology currently used by millions of individuals [41]. The network is developed and open-sourced by a team largely funded through various sources [65], run by independent operators, and audited by a large community of academics and individuals. The Tor Project seeks operators with intrinsic motivation to run relays in the network, as there is no monetary benefit to running a relay. This governance model has contributed to establishing a high level of trust, especially since the developers do not select who may contribute by running relays. It contrasts with other networks, such as the one created by Mullvad for its multihop VPN [1, 2] or Apple Relays [3], for which both companies actively run relays themselves and do not open-source their code. It also contrasts with networks such as Nym [12], where nodes are run by various independent operators who are incentivized with NYM tokens for their contributions.

However, over the years, the openness and independence of the Tor developers have come at a price. The diversity of relay operators has resulted in varying rates of adoption of new Tor versions. This has forced the developers to allow old Tor versions to continue running in the network, which has led to various issues. Examples of discovered attacks [18, 29] by academics were eventually fixed but remained exploitable for several years in the live network, even on updated relays, due to backward-compatibility concerns with old versions. The Tor developers also struggled to deploy major changes to their network, often requiring compatibility code to interface with the behaviors of different live Tor versions. This further delayed improvements reaching Tor clients, since those improvements would occur only if the client happened to build circuits with the most up-to-date Tor relays.

Since 2019 [22], more than 15 years after Tor's initial deployment, the developers decided to become more restrictive toward operators. The Tor Project redesigned the lifecycle of its software, establishing an end-of-life (EoL) policy that involves no feature backporting to multiple older versions and introduced a rejection deadline [63]. This approach effectively strikes a compromise between the independence and freedom of operators and the requirements of the Tor network as seen by the Tor Project. Each major version is set to expire and be rejected from the network a few months after the next major version is officially released. This EoL policy led to the rejection of hundreds of relays and approximately 10% of relays in the worst event. This rejection does not necessarily occur at regular intervals, but over the last few years it has occurred about once a year.

Outdated relays are rejected from the network based on the Tor version they advertise to the authority servers. When creating the network consensus, the authorities will not include rejected relays in the final consensus, effectively excluding them from participating in the network. This means that Tor clients will never use excluded relays. An excluded relay can usually rejoin the network by updating to a supported version.

We aim to study the tension created by this policy change and are interested in the following questions:

RQ 1 What is the sociotechnical impact of relay rejections on the anonymity provided by Tor?

RQ 2 How can the individual security contribution provided by a Tor relay be measured?

To answer these research questions, we make the following contributions, in order of appearance:

C 1 A user study to understand how relay operators perceive the policy and to learn how they operate and what types of frictions are usually encountered during the lifecycles of the software used to contribute to the Tor network (Section 3).

C 2 A quantitative evaluation of the impact of rejecting relays, using simulations based on historical data from the Tor network and the Internet topology (Section 5).

C 3 Metrics to measure the contribution of individual Tor relays to the network (Section 6).

In the following sections, we present our findings and provide several contributions and recommendations to the community with a focus on social and technical aspects.

2 Background

2.1 Tor basics

The Tor network [14, 66] consists of approximately 9500 relays and nine directory authorities scattered across the globe and operated by volunteers. A directory authority is a special-purpose relay that maintains a list of currently connected relays and publishes an hourly *consensus* file together with the other directory authorities. Published consensus files contain information about all active relays in the network and are used by Tor clients to make informed decisions about which relays to select to build circuits.

A Tor circuit typically consists of three relays: guard, middle, and exit. Each relay in the circuit is aware only of the identity of the previous and the next relay. Thus, the guard relay knows the IP address of the client but not the destination, and likewise, the exit relay knows the IP address of the destination but is unaware of the source of the request.

Multiple constraints apply when selecting relays to build a circuit [66]¹. The same relay cannot be selected twice in a circuit. Additionally, any relay in the same family, administered by the same person or organization, will not be chosen as another hop in the same path. Moreover, it is not permitted to select more than one relay in a given network range, which defaults to /16 for IPv4 and /32 for IPv6.

¹ A recent proposal suggests relaxing those restrictions. The topic is under discussion at the time of writing. See spec.torproject.org/proposals/354-relaxed-restrictions.html.

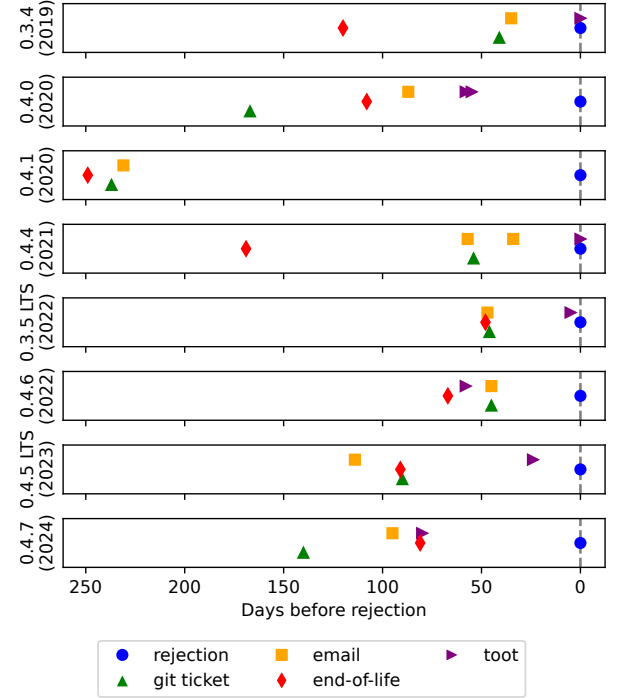


Figure 1: An overview of the timeline of events preceding the exclusion of outdated relays from the Tor network.

During path selection, when considering a relay for a specific position in a circuit, the choice is made based on the *flags* and *weights* of the relay, along with positional factors called *bandwidth weights*. These criteria enable clients to compute the *bandwidth* for a specific relay in a specific position. Complete details of these load-balancing equations and their calculations are documented in related works [30, 56].

2.2 Updating Tor relays

Before 2019, limited restrictions were enforced on the Tor versions accepted as relays in the network. Because of this, by late 2019 the Tor network comprised relays running 85 different Tor versions, with the oldest version released in December 2013 [22]. Some of those versions were still supported by the Tor Project, and some were no longer supported.

This situation posed two main challenges for the Tor Project. First, it was maintaining five version series (i.e., major versions) to provide fixes for major stability issues, security vulnerabilities, and portability regressions [22]. Second, operating the network with many versions made it more difficult to deploy certain security fixes and new features. Changes were generally available only for the supported versions, but those changes needed to account for the unsupported versions still operating in the network.

Because of these challenges, the Tor Project decided to stop accepting relays running unsupported versions. The project began contacting operators running such versions, asking them to update their relays to continue participating in the Tor network.

Currently, the Tor Project handles the process of rejecting EoL versions using public announcements (via its mailing list [67] and its social media accounts [68, 69]) and direct emails to relay operators who provided contact information when setting up their relays. Since the first rejection of relays in 2019, the Tor Project has proceeded as follows: once a version reaches EoL, the Network Health team sends an email to the Tor relay mailing list and opens a git ticket to track changes to be made in the source code to remove the rejected versions effectively. The time elapsed between a version becoming EoL and its actual exclusion was significantly longer in the first exclusion rounds. For the last exclusion round, the notification to the mailing list was sent before the version became EoL. Figure 1 provides an overview of the different events leading to the exclusion of eight Tor versions.

3 Social aspects

To address the social component of RQ 1, we collected feedback from volunteer relay operators regarding their experiences as Tor relay operators. Our objective is to determine whether relay operators are aware of the EoL policy enforced by the Tor Project and how they perceive the value of this policy. We also examine how relay operators keep their relays up to date (if they do) and how they react when, or if, their relays are rejected from the network.

3.1 Method

To investigate how Tor relay operators manage their relays and their views on the EoL policy and exclusion cycle, we conduct a qualitative study of current and past operators. Following Lazar et al. [39], our study has two parts.

First, we construct an interview guide for semi-structured interviews and conduct a pilot interview with a colleague running a Tor relay to identify areas of interest not covered by our initial interview guide. Based on this pilot interview, we augment the guide with additional topics of interest. As the colleague is aware of our research direction and thesis, we do not include the pilot interview in the results.

Second, we contact Tor relay operators via the tor-relays mailing list and via printed signs at FOSDEM² in Brussels. Interested Tor relay operators take the interview in one of the following settings: in person during FOSDEM, via an online voice call, via a synchronous text chat, or via asynchronous emails. In-person interviews and online voice calls are audio-recorded after obtaining consent from the participant. For text-based interviews, we save the transcript for analysis. We discuss ethical considerations in more detail in Section 8. The recruitment process and the interview guide are presented in Appendix C.

We acknowledge that there might be significant selection bias in how we recruit participants: we are interested in how relay operators manage their relays with a focus on the EoL policy. Finding participants via the mailing list and at FOSDEM may not yield a representative set, as the least dedicated relay operators are probably not following the mailing list or attending FOSDEM. For this reason, we adhere to a qualitative analysis and refrain from conducting a quantitative analysis (e.g., via an online form with closed questions) to avoid amplifying the selection bias.

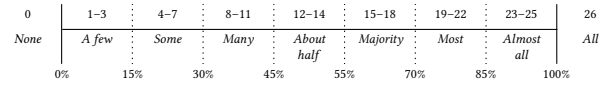


Figure 2: Quantifiers and percentages used to present the qualitative results. Each quantifier always refers to the same number of participants throughout the section. *Emphasis* is used when the quantifier appears in the text to remind the reader that it carries a specific meaning.

3.2 Analysis

Between January 31 and February 8, 2026, we conducted 26 interviews. We conducted the interviews as follows: 6 in-person interviews at FOSDEM, 12 online voice calls, 7 synchronous text interviews, and 1 asynchronous text interview. The same researcher conducted all interviews. All interviews but two were conducted in English, which is not the native language of the researcher and often not the native language of the participant either. The two remaining interviews were conducted in the native language of both the interviewer and the participant.

Using transcriptions of the audio interviews and saved transcripts of the text interviews, we develop a codebook by having two coders openly and independently code a set of three interviews. From the codes created by the two coders, we build a common codebook used to code all interviews. We augment this common codebook when new codes emerge. The resulting common codebook has 137 codes with 1511 coded segments in total.

Both coders regularly check the coded interviews to resolve potential points of disagreement and to align their interpretations of the data. We compute Cohen’s kappa [10, 39] score per code at the interview level. The average score for the two coders is 0.78 (SD 0.23), indicating satisfactory agreement [39].

Similar to previous work [17, 23, 35, 38, 75, 78], we report the prevalence of our findings using the quantifiers presented in Figure 2 instead of reporting exact counts.

3.3 Qualitative results

In total, participants manage 1379 relays located in 36 distinct countries. On average, participants manage 53.1 relays (SD 148.4) across 2.6 countries (SD 2.1). The *majority* of participants manage 5 relays or fewer; *some* participants manage a few dozen relays, and *some* participants manage hundreds of relays. *Many* participants have managed relays for less than 5 years, and the *majority* of participants have managed relays for 5 years or more. The *majority* of participants are solo operators, and *many* are part of 8 distinct groups or organizations. We choose not to give more precise profiles to limit information exposure and potential reidentification.

Although our participant sample might be biased toward committed relay operators, *some* participants are not aware of the EoL policy. The other participants are at least vaguely aware of the policy.

3.3.1 Update workflow. The *majority* update Tor manually: the process needs to be started by a human operator. However, the manual update is often triggered by a regular reminder or a message on the Tor Project mailing list announcing a new release. Among

² Fosdem.org/2026/

the participants stating that they manually update Tor, we find people operating large numbers of relays who still prefer to update manually. For example: “So we only update when it’s needed, for example, when a security patch is relevant to our situation or when the Tor Project actually marks a previous version as EoL or dangerous or bugged.” (P12) Participants mention various motivations to manually update: for a few participants, there are no automatic ways of updating on their operating system (OS); a few participants want to manually check the changelog; some do not need their update process to scale up; some feel that the process is simpler, simple enough, or familiar; a few want to evaluate whether the relay reputation hit is worth taking or want to test the update on a few relays before applying it to all relays. Finally, one participant mentions that they never update their relay: “as an extremely casual relay operator, I mostly don’t [update my relays]” (P25). Another participant performing manual updates says: “I honestly forget I run Tor nodes unless something makes me think about them” (P10).

Among participants updating automatically, several also manually update when requested by the Tor Project if their update schedule is infrequent. “Packer rebuilds the cloud images at least once per month on a schedule, but we can trigger immediate rebuilds in case of CVEs or other notices that require a quicker turnaround.” (P16) Motivations to automatically update include familiarity with this process, supporting the Tor developers, applying the latest security patches, and reducing maintenance burden. It is important to note that operators who manually update their relays are not necessarily slower to update than those with automatic updates: automatic updates might run on a monthly schedule, whereas manual updates can be performed upon notification on the Tor Project mailing list, for example. “I’m on the announcement mailing list, so when I see that there’s a new release then I usually update it certainly within a week, maybe within a few days.” (P08)

3.3.2 Tor package repository. Many participants report that they use the default package repository of their distribution for Tor. Most participants use Debian and/or Ubuntu. Some participants use other Linux distributions, such as Alpine or NixOS, for instance. There are also a few participants who use FreeBSD and/or OpenBSD with the goal of increasing OS diversity on the network. It is not unusual for some participants to run multiple distributions and/or OSes.

We analyze the version of Tor shipped by different OSes in relation to the Tor versions accepted in the network since 2019. Figure 3 provides a visual representation of Tor versions shipped by OS repositories. Whether we look from the Tor Project’s point of view or from package maintainers’ point of view, multiple versions are maintained simultaneously in both cases. The grayed part represents versions rejected from the network. We notice that Debian generally ends up shipping a Tor version that is not accepted anymore toward the end of its lifetime, even if the Tor version it ships is marked as long-term support (LTS). The same applies to Ubuntu LTS versions. In addition, however, Ubuntu LTS does not generally ship an LTS version of Tor. For example, if Ubuntu 22.04 (LTS) shipped Tor 0.4.5 (LTS) instead of 0.4.6, it would still have been excluded, but 5 months later. Even OSes with shorter release cycles, such as Alpine Linux or FreeBSD, sometimes ship outdated Tor versions. This is due to multiple release life cycles having different schedules, philosophies, and maintainers.

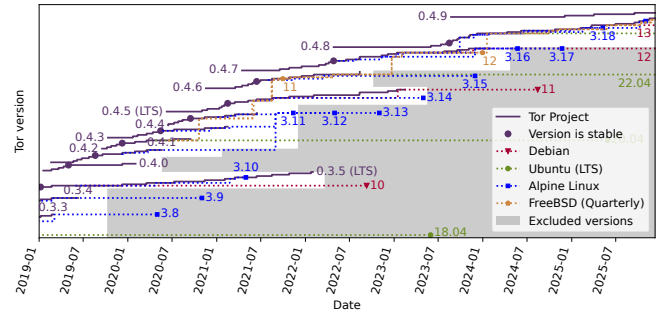


Figure 3: Tor binary packages provided by different OS software repositories (dotted lines) compared with the Tor Project’s official source code releases (full lines). Dotted lines stop when an OS version is no longer supported, annotated by a marker. Full lines stop when a Tor version series reaches EoL.

OS release cycles can affect the ability of volunteers to participate in the Tor network. Some operators are not always aware of this: “You might know that Debian has some very old packages in it. It took me a while before I learned that actually if you just enable the ORPort on Debian Tor, it’s rejected by the network because it’s too old.” (P25) Even if aware, one relay operator using Debian with the default Tor package mentions that “The relay was still seeing traffic at the time, so I thought I’ll just let it be and it continued to receive and process traffic, so that was fine.” (P03)

We note that the Tor Project is aware of the problem. In their installation guides, they recommend using their own package repositories for Debian, Ubuntu, RHEL, and Fedora [73]. Concerning FreeBSD, they recommend using “latest” package releases instead of the default “quarterly” option in order to get faster Tor updates [74].

Package maintainers providing fast updates help operators run the latest version without too much burden: “[FreeBSD and OpenBSD] both have high-quality Tor packages available, because there is Yuri who maintains high-quality packages for FreeBSD and OpenBSD. So we always have them fast, all the updates.” (P12) However, they recognize that this burden is instead on the package maintainer’s shoulders, who need free time to provide updates, and that it could be a single point of failure if the maintainer is absent or if the package repository is down.

3.3.3 Opinion about EoL policy. During the interviews, we provide a brief description of the EoL policy to the participants. We then ask for their opinions about this policy. Most participants are aware of the policy, and most are positive about it. This positive reaction is generally the first point they express when asked for their opinion. Some have either no strong opinion or understand the rationale without an explicitly positive sentiment in their wording. Almost all participants support the policy for the security benefits they perceive: new releases often fix potential vulnerabilities or introduce new security-related features. For some participants, updating is also important to help the developers deploy new features and remove the burden of maintaining multiple version series concurrently.

While participants are generally positive regarding the EoL policy, some nuanced aspects are mentioned. According to *a few* participants, the policy adds a maintenance burden on operators, who must keep systems up to date and track which version is about to be rejected. This burden is generally considered reasonable by the operators who mention it. However, one participant thinks that, while it is normal to expect this from “*middle- to large-scale operators*”, it can discourage small operators: “*There are also many relay operators that run like one relay or two relays [...], they didn’t sign up for having monthly maintenance.*” (P12) Another participant thinks the burden is reasonable only if operators are notified one or two months in advance (P18). *Some* participants are concerned that the policy might reduce the diversity of relays in the network. Indeed, less committed operators might run relays in otherwise uncovered areas and be removed from the network if they do not update frequently enough. We note that, except for two of them, these nuanced operators have managed relays for five years or more. Despite the nuances, participants value the Tor Project and its community for their voluntary support.

Finally, based on participants’ wording, it was not always clear whether they could distinguish between the EoL policy and the “not recommended” flag that was recently applied to some 0.4.8.x versions. Being clearer about the EoL policy when newcomers set up a Tor relay is suggested by *a few* participants. Clear and proactive communication from the Tor Project is important to *some* participants; they appreciate emails from the Tor Project notifying them that a relay they operate is down or has an issue. This was also discussed on the mailing list [50, 77].

3.3.4 Reaction if rejected. As mentioned, the participants we interviewed are likely among the most committed relay operators. Only two participants are certain that one of their relays was excluded from the network. Both of them updated the Tor package to continue participating in the network.

Some participants would be annoyed, *some* embarrassed, or *some* surprised at first if their relay were excluded from the network. *Some* would feel as though they had failed in their duty as relay operators. Given the results presented in the previous sections, multiple operators acknowledge that they would understand why their relay was rejected. Most importantly, *about half* of the participants mention that they would update their relay if they learned that it was excluded. *Many* also mention that it would prompt them to revise their update process.

In practice, however, most excluded relays never return. To obtain an actual number, we sample historical consensus from the directory authorities [71] at three different dates: one month before, just before, and one month after an exclusion date. Figure 4 presents our results regarding the April 2024 exclusion. Out of the 7765 relays in the network, 553 had a to-be-excluded Tor version one month before. 75 of them updated before the exclusion deadline, while 440 did not and were excluded. Finally, by looking one month later, we observe that 68 (15%) excluded relays returned with the same fingerprint, while 372 (85%) excluded relays did not return (or not with the same fingerprint, at least). In addition, the chart shows that several hundred up-to-date relays enter and leave the network over the course of one month. This flow is what we call the natural churn of relays.

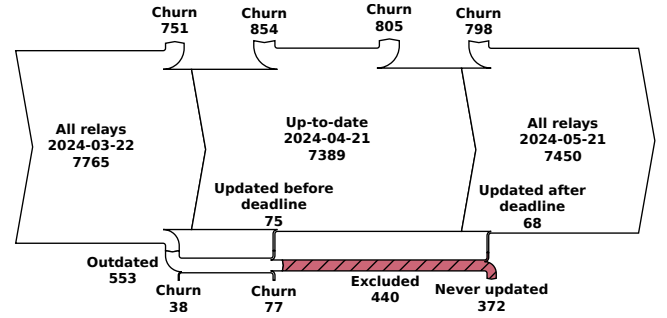


Figure 4: Flow of Tor relays before and after the 2024-04 exclusion (< 0.4.8.0-alpha-dev). *Churn* flows represent relays entering and leaving the network, unrelated to the EoL policy. *Outdated* relays run a version that is older than the one accepted on 2024-04-21; if they update, they join the *Up-to-date* flow, otherwise, they fall into the *Excluded* flow.

3.3.5 Key takeaways. There is no single way to operate relays in the Tor community. For various reasons, the *majority* of operators prefer manual updates, whereas others prefer automatic updates of the Tor daemon. Although the Tor Project recommends against using default package repositories for well-known OSes [73], we encountered operators who use them. The differing lifecycles of OS distributions may result in fresh relays from new contributors being excluded immediately. *Some* participants are aware of the EoL policy. Yet, even within our sample of dedicated operators, *most* were only vaguely aware. Others became aware of it only after their relay was actually excluded. The policy is widely supported, primarily for improving network security, enabling faster feature deployment by Tor developers, and reducing maintenance by limiting parallel versions. However, experienced operators highlight concerns about reduced diversity, unclear policy communication, and the impact of announcement timing.

In the next sections, we define Tor’s threat model and adversaries and analyze how the rejection of relays due to the EoL policy impacts its security (i.e. its anonymity). This direction is supported by the results of the quantitative analysis presented: participants generally support the EoL policy and cite the security of the network as a main benefit. We also analyze the impact of the hundreds of relays entering and leaving the network, irrespective of the EoL policy.

4 Threat model

Tor’s threat model aims to defend and provide a degree of anonymity to users, on a best-effort basis, against non-global adversaries. Prior literature has developed several deanonymization techniques employing local adversaries to identify Tor users [32, 36, 44, 57]. These techniques are generally feasible when an adversary can intercept communication on both sides of a Tor circuit. The adversary can then correlate traffic between the client and the guard with traffic between the exit and the final destination [44]. Such attacks are called traffic confirmation attacks: they reveal that a given client connected to a given destination address. In this threat model, adversaries include relays within the Tor network itself; an adversary

controlling a portion of end paths could position itself as both guard and exit relays in circuits to deanonymize users [57]. Other potential adversaries include autonomous systems (ASes) [19] that handle Internet traffic passing through their networks and are therefore capable of correlating traffic at each end of a circuit. However, these are not the only entities capable of observing Tor user traffic. Some studies [31, 34, 45] also suggest considering Internet exchange points (IXPs) as potential threats.

Similar to Barton and Wright [6], Rochet et al. [58], Sun et al. [61], we categorize these adversaries into two classes. The first class is the relay adversary, which controls a subset of Tor relays to deanonymize users. The second class is the network-level adversary, namely an AS or IXP, capable of intercepting communications on both sides of a circuit when traffic passes through them. We consider such entities suspect if they are able to observe both ends of Tor circuits.

5 Exclusion anonymity impact

To study the technical anonymity impact that relay rejections had on Tor users (RQ 1), we created an updated version of TorPS (Johnson et al., 2013 [31]) to match the path selection algorithm in effect during the exclusion periods. This tool evaluates, via a Monte Carlo method, the probability of compromise that a typical Tor client faces while using the Tor network. This probability is cumulative as time advances: the more a Tor client uses the network, the more likely it is to be compromised. We measure this cumulative probability over time for a typical Tor client. We evaluate the impact that relay rejections had on Tor clients during a timeframe spanning 14 days before to 14 days after an exclusion date to gauge the risk created by rejecting honest relays. We assume that potentially malicious relays are up to date or misreport their versions and therefore would not be rejected. This is a reasonable assumption: attacking the network already takes high effort, whereas announcing a fake version or updating a relay takes little effort.

The exclusions considered are the following: 2021-12-01, 2022-03-21, 2022-10-07, and 2023-05-17 [63]. These exclusion rounds are those for which the Tor Project shared the list of excluded relays. We also use the algorithm from Qiu and Gao [52] to infer the AS path between the client and the guard and between the exit and the destination. Qiu and Gao [52] use Border Gateway Protocol (BGP) data to learn about existing paths and neighboring ASes on the Internet. When a path is announced via BGP, Qiu and Gao [52] save it as what they call a *sure path*. During inference, paths might be extended using neighboring ASes, and the number of hops added to a path using neighboring ASes is called the *unsure length*. When inferring a path from a given AS to a destination prefix, we use the *Least Uncertainty First* strategy from Qiu and Gao [52]: the inference procedure returns paths with the smallest *unsure length*, ordered by the probability that the path is used on the Internet. This probability is derived from the number of times the path has been observed in BGP data. We always take the first result as the inferred path to ensure reproducible results across executions.

Our goal is to measure the impact of the exclusions on Tor clients' anonymity. To this end, we consider (i) a network adversary compromising some ASes (Section 5.1), (ii) a network adversary compromising some IXPs (Section 5.2), and (iii) a relay adversary adding

malicious relays to the Tor network (Section 5.3). Following Johnson et al. [31], for each exclusion and adversary type, we compute the probability of compromise over time. A client is considered compromised as soon as an adversary compromises one of its circuits.

Similar to previous literature [6, 15, 55, 58, 61, 76], we use data from multiple sources to reproduce the Tor network state during past exclusions. The data used are time-dependent whenever available; that is, the past is reconstructed as faithfully as the available data allow.

In total, four sets of simulations are run (one set per exclusion) with the following parameters:

- 10,000 Tor clients whose locations are computed based on Tor Metrics [70] and on the AS ranking provided by CAIDA [8];
- the 100 most popular destinations, taken from Tranco [40], which provides a ranking of popular domains by aggregating the rankings of other providers over a period of 30 days;
- consensus documents from the Tor network [64] to model the Tor topology;
- BGP and IXP data to model the topology of the Internet, from RouteViews [59] and the Center for Applied Internet Data Analysis based at the University of California's San Diego Supercomputer Center (CAIDA UCSD) [9], respectively.

For each set, two simulations are run: one with the historical data [64] and one with the historical data augmented with the relays that were scheduled for exclusion, effectively simulating the network as if the exclusion had not occurred. The difference between the two results represents the impact of the exclusion. The results we obtain from the simulations may differ if we decide to change the above parameters. However, this setup reflects the general trend of the evolution of user anonymity. Moreover, the differences between two experiments using the same setup are in any case worth comparing.

5.1 AS adversary

In the AS adversary setting, the two most frequent ASes (i.e., the two ASes that appear most frequently on both ends of the circuit) are considered suspect ASes but non-colluding. Similar to DeNASA's approach [6], any AS appearing at both ends of a Tor circuit is considered suspect; however, a few ASes appear significantly more frequently. These two ASes are treated as suspect ASes, and the evaluation is conducted against them. An evaluation against all of them could also be performed, which would shift the base probability closer to one. This choice is motivated by the marked difference from the first two ASes (Appendix F, Figure 11).

Figure 5 shows the cumulative distribution function of compromised clients over time for four recent exclusions, based on our simulations. On each graph, the solid red line represents the historical data: when the exclusion occurs, some clients are impacted and observe their guard go offline. Those clients must select a new guard. As the new guard is located elsewhere, migrating clients may now have a suspicious AS on the path between them and their guard. As they connect to many destinations over time, it is also likely that the same suspicious AS is on the path between the exit relay and the destination. For this reason, a number of clients generally become compromised a few hours after the exclusion.

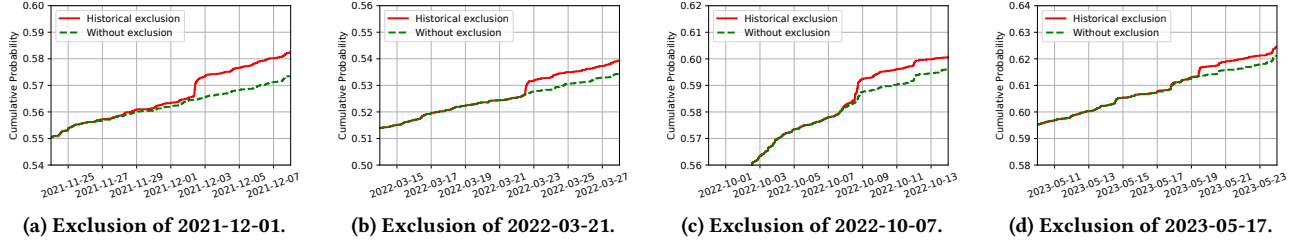


Figure 5: Probability of path compromise with top-2 suspicious ASes. The gap between the red and green lines shows the advantage the exclusion gave the suspicious ASes – roughly a few days of running the network without excluding relays.

The dotted green line on each graph represents a scenario in which the exclusion does not occur. In this case, clients do not observe their guard go offline, and they can continue using it without selecting a new guard and potentially using a compromised path.

A general slope of the cumulative distribution function is also observed, and analysis of the data reveals that the natural churn of relays in the network (observed earlier in Figure 4) is the main factor. Another factor is the guard-flag assignment logic for some low-bandwidth relays. The data reveal that some low-bandwidth relays gain and lose their guard flag due to variation in their bandwidth as measured by the authorities. This instability could be mitigated by slightly increasing the bandwidth bar to gain the guard flag and reducing the bar to lose it.

It is noteworthy that around 1% of additional clients were compromised during the 2021 exclusion, whereas only about 0.30% were compromised in 2023. This decrease over time can be attributed to two main factors. First, relay operators may have become accustomed to updating their relays regularly; otherwise, their relays are rejected from the network. Second, the Tor Project gained insights from previous exclusions and improved the workflow over the years to keep relay operators up to date with the policy. This decrease is also evident in mailing list communications from the Tor Project: in their email regarding the 2019 exclusion round, they stated that the excluded relays accounted for 13.25% of the consensus weight (computed approximately one month before the exclusion) [47]. For the 2023 exclusion round, the Tor Project stated that the excluded relays accounted for “roughly 7% of the advertised bandwidth of the network” (computed five months before the exclusion) [37].

While the number of relays listed for exclusion may seem high (ranging from 344 to 644 for the exclusions we consider) [63], it is noteworthy that the effect of the exclusion is not as alarming. Although an additional fraction of Tor clients (up to $\approx 1\%$) is suddenly compromised as a result of the exclusion round, a closer examination of the graph in Figure 5a shows that the proportion of compromised clients in the historical scenario immediately after the exclusion (on 2021-12-03) is the same as in the more optimistic scenario where the excluded relays were kept four days later (on 2021-12-07). This implies that, even without exclusion, the same proportion of compromised clients would have occurred, but four days later.

Note that the high value for the base compromise rate is due to how the Internet is currently built around high-speed tier-1 transit ASes such as AS6939 (Hurricane Electric) and AS3356 (Level 3 Parent). It is possible to reduce this base-rate compromise with

path selection logic that leverages the current surplus of bandwidth within guard and middle positions of circuits to steer paths away from these suspicious ASes while maintaining a balanced network [58]. Another unexplored approach could be to revisit the guard-flag assignment logic to account for such issues; the assignment does not even need to be the same for everyone. It is, however, important to understand that the *how* is not the most difficult problem; addressing this problem correctly requires tapping into the surplus of bandwidth within path positions, which is equally useful for solutions to other adversarial threats that the Tor network faces, such as website fingerprinting potentially addressed by traffic coverage [43, 51], which also exploits bandwidth surplus. Ultimately, it becomes more a question of *what* should be done rather than *how* to approach these problems individually. Possibly, a compromise could be found to tap into the same bandwidth surplus, but *what* compromise is needed remains an open question.

5.2 IXP adversary

Similar to the AS adversary, the top three organizations that administer multiple IXPs are considered suspicious. Figure 6 presents the experimental results. The same observations made for AS adversaries apply.

Inferring which IXP transfers data between two ASes is not straightforward. First, previous literature [5] indicates that detecting an IXP between two ASes is not trivial. Second, when ASes on the path are known to be connected by multiple IXPs, there is no simple way to determine which IXP is used, if any. In fact, even if two ASes are connected by a single IXP, those two ASes might also be paired directly. In that case, the traffic might completely avoid the intermediary IXP. To determine the IXP that observes the traffic flowing between a pair of ASes, multiple strategies are possible.

In Figure 6, two selection strategies are depicted. First, one IXP is selected at random when multiple IXPs are likely to connect two ASes. Second, all IXPs that could connect two ASes are selected and presented as the worst case on the graphs in Figure 6. That is, when an IXP must be selected to connect two ASes, all IXPs connecting the two ASes are selected. As expected, the proportion of compromised paths is slightly higher with this heuristic than when a single IXP is selected at random. However, the relative difference between the historical exclusion and the simulation without exclusion remains similar. This indicates that the inference algorithm used to select IXPs influences the proportion of compromised paths but does not invalidate our claim.

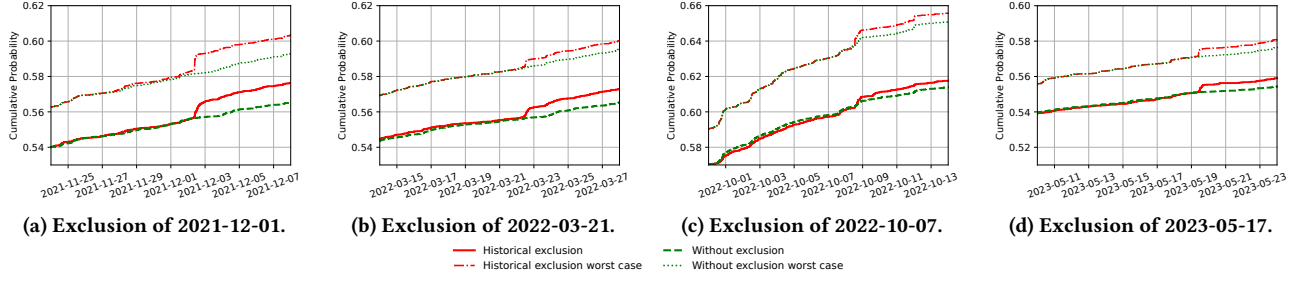


Figure 6: Probability of path compromise with top-3 suspicious IXP organizations. The gap between the red and green lines shows the advantage the exclusion gave the suspicious IXPs – roughly a few days of running the network without exclusion.

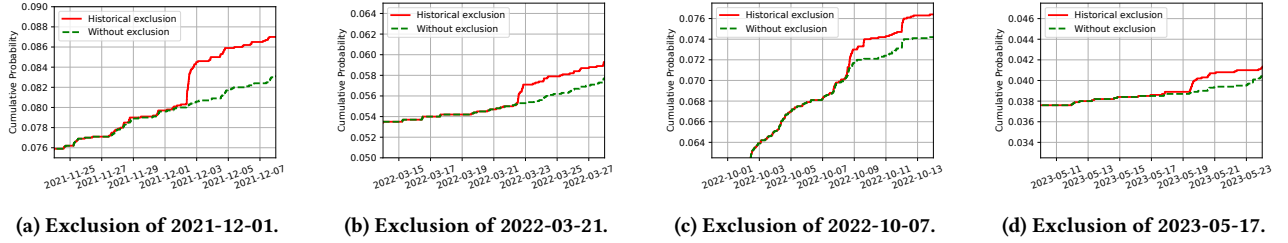


Figure 7: Probability of path compromise with adversarial relays (10 exits and 20 guards, each with a 150,000 consensus weight). The vertical gap between the red and green lines represent the advantage the exclusion gave to the adversarial relays: the advantage is equivalent to a few days of running the network without excluding relays.

5.3 Relay adversary

For each relay adversary experiment, the adversary controls 10 exit relays and 20 guard relays added to the network. Each adversary-controlled relay has a consensus weight of 150,000, similar to the top relays in the network. These figures are reasonable assumptions based on previously reported malicious actors [48, 49].

Figure 7 shows the cumulative distribution function of clients compromised by the relay adversary over time. As with the network adversary, the increase in the number of compromised clients diminishes over time. This can be explained in the same way as for the network adversary. However, it should be noted that the consensus weight for the adversary remains the same across all experiments, while the Tor network grew over time. Thus, the total fraction of the consensus weight controlled by the adversary decreases over time.

One key difference is worth noting: in Figure 7a, the probability of compromise after the exclusion is not reached as quickly as in Figure 5a. In Figure 5a, it took four days for the scenario in which the excluded relays are kept in the network to reach the same level of client compromise as in the historical scenario. Here, the graph in Figure 7a is not wide enough to show when this occurs, but the raw data indicate that it took nine days (instead of four). This could be explained by the overall diversity that currently exists in the network against each adversarial model. The higher the diversity, the lower the adversary’s strength. Our results show that the concentration of paths to a few top ASes is more significant than the concentration of paths to a few top relays. This might be addressed by adapting the path selection algorithm, and the literature already explores this idea [56].

6 Individual relay utility and security contribution

In the previous section, we showed that excluding relays gives adversaries (presented in Section 4) an advantage over a situation without exclusion. We now quantify which relay provides the most security to the network among the relays about to be excluded. To this end, we propose metrics that capture the utility and security contribution of a specific relay (RQ 2). This contrasts with security metrics in related work, which usually aim to evaluate the security of the whole network [7, 16, 56, 60, 62]. However, summing the contribution of each relay would yield an evaluation of the network’s security. Our metrics focus on quantifying relays’ contributions rather than evaluating client anonymity directly. Each metric is designed to capture the security contribution against a specific adversary we consider. This depends on multiple properties of the relay and the network, and on how those properties are used in the path selection algorithm.

In the context of exclusions, the goal of these metrics is to indicate which relays are most important to maintain in the network.

6.1 Notations

We extend the conventions in notation used by previous literature [25, 26] and the Tor specification [66]. For a set of relays in the network, we use uppercase calligraphic letters (e.g., the set $\mathcal{G}$ of all relays flagged as guards in the network). We use lowercase italicized letters for a particular relay (e.g., a guard g).

For the remainder of the section, we define the following sets:

- $\mathcal{R}$, the set of all relays in the network.
- $\mathcal{A}$, the set of all network adversaries.

- $\mathcal{G}$, the set of all relays flagged only as guard.
- $\mathcal{E}$, the set of all relays flagged only as exit.
- $\mathcal{D}$, the set of all relays flagged both as guard and exit.

As mentioned in Section 2.1, for a relay r , the bandwidth used in the path selection algorithm is computed differently depending on the position in the circuit in which we want it to be. We denote the bandwidth for a relay in the guard position as $\text{bw}_g(r)$. Likewise, for the exit position, we write $\text{bw}_e(r)$.

We also use the normalized bandwidth of a relay with respect to its position in a circuit. For instance, let r be a relay, and let $\text{bw}_g(r)$ be its bandwidth in the Tor network as a guard. Then the normalized bandwidth of that relay is computed with the following formula:

$$\text{norm}_{\mathcal{G} \cup \mathcal{D}}(\text{bw}_g(r)) = \frac{\text{bw}_g(r)}{\sum_{g \in \mathcal{G} \cup \mathcal{D}} \text{bw}_g(g)}. \quad (1)$$

Note that this normalization is performed over the bandwidth of all the guard flagged relays in the network, not all the relays.

We also define, for a given guard g used in a circuit, the set $\text{reach}_{\mathcal{E} \cup \mathcal{D}}(g)$ corresponding to all exit flagged relays that can be paired with g in the circuit. Likewise, for a given exit e in the circuit, $\text{reach}_{\mathcal{G} \cup \mathcal{D}}(e)$ is the set of guard flagged relays that can be paired with e as the guard of a circuit.

6.2 Relay utility metric

We define relay utility as the number of circuits that can be built using that relay. This depends on the number of other relays that can be combined with it to complete the circuit. Instead of considering the raw number of relays, we use the relays' bandwidth. This captures not only the number of circuits but also how many clients can use those circuits.

This metric does not make strong assumptions about a potential relay adversary. We assume that relays about to be rejected are not malicious and that the adversary already operates other relays in the network. The metric estimates relay utility for clients: useful relays have relatively high bandwidth and can be paired with a substantial share of exit capacity. This value also indicates the marginal increase in success for an already operating relay adversary: if a relay is used by a large number of clients and is excluded, all those clients will have to select new relays for their circuits and might end up selecting an adversary-operated relay.

When considering a relay g as a guard in a circuit, we are interested in the exit relays that can be paired with it. For a guard $g \in \mathcal{G}$, we evaluate its utility as follows:

$$\text{RM}(g) = \text{norm}_{\mathcal{G} \cup \mathcal{D}}(\text{bw}_g(g)) \times \sum_{e \in \text{reach}_{\mathcal{E} \cup \mathcal{D}}(g)} \text{norm}_{\mathcal{E} \cup \mathcal{D}}(\text{bw}_e(e)). \quad (2)$$

For an exit $e \in \mathcal{E}$, we obtain the following metric using the same reasoning:

$$\text{RM}(e) = \text{norm}_{\mathcal{E} \cup \mathcal{D}}(\text{bw}_e(e)) \times \sum_{g \in \text{reach}_{\mathcal{G} \cup \mathcal{D}}(e)} \text{norm}_{\mathcal{G} \cup \mathcal{D}}(\text{bw}_g(g)). \quad (3)$$

When a relay $d \in \mathcal{D}$ can serve as a guard or an exit relay in a circuit, its total utility is the sum of its utility as a guard and as an exit. Note that there is no double counting of resources: $\text{bw}_g(d)$ and $\text{bw}_e(d)$ are computed by the authorities to distribute the resources

of d between the guard and exit positions. This yields the following metric:

$$\begin{aligned} \text{RM}(d) = & \text{norm}_{\mathcal{G} \cup \mathcal{D}}(\text{bw}_g(d)) \times \sum_{e \in \text{reach}_{\mathcal{E} \cup \mathcal{D}}(d)} \text{norm}_{\mathcal{E} \cup \mathcal{D}}(\text{bw}_e(e)) \\ & + \text{norm}_{\mathcal{E} \cup \mathcal{D}}(\text{bw}_e(d)) \times \sum_{g \in \text{reach}_{\mathcal{G} \cup \mathcal{D}}(d)} \text{norm}_{\mathcal{G} \cup \mathcal{D}}(\text{bw}_g(g)). \end{aligned} \quad (4)$$

Multiple factors govern the value of the relay utility metric: (i) the share of bandwidth the relay provides in its position (captured by $\text{bw}_g(r)$ and $\text{bw}_e(r)$), (ii) the share of relays that can be used to build a circuit (captured by $\text{reach}_{\mathcal{G} \cup \mathcal{D}}(r)$ and $\text{reach}_{\mathcal{E} \cup \mathcal{D}}(r)$), and (iii) relay-selection constraints such as families and network addresses (captured by $\text{reach}_{\mathcal{G} \cup \mathcal{D}}(r)$ and $\text{reach}_{\mathcal{E} \cup \mathcal{D}}(r)$). For more details, the reader may refer to Appendix D.

6.3 Network adversary metric

The second metric we propose measures how unlikely it is that a relay is used to build a compromised circuit. This is applicable to AS and IXP adversaries. However, we consider only AS adversaries in this section. As mentioned in Section 5.2, inferring IXPs along an Internet path depends on the selected heuristic and can yield very different results.

The idea is to compute the probability that a network adversary a appears on both ends of a circuit given that a relay r is used to construct the circuit. We denote that probability as $\text{Pr}(a|r)$. This probability is the product of the probability $\text{Pr}(a|g)$ that a appears before the guard g and the probability $\text{Pr}(a|e)$ that a appears after the exit e . We also use the probability $\text{Pr}(r)$ that a relay r is in a circuit and the probability $\text{Pr}(r|r')$ that r is in the circuit given that r' is in the circuit.

For a relay r , we then compute the following metric:

$$\text{NM}(r) = \text{Pr}(r) \prod_{a \in \mathcal{A}} 1 - \text{Pr}(a|r), \quad (5)$$

capturing, for all adversaries a , the probability that r is in the circuit and does not put a in a position to compromise the client. In practice, we can replace the probability $\text{Pr}(r)$ with the bandwidth $\text{bw}(r)$, as this is how path selection weights the relays when choosing them randomly.

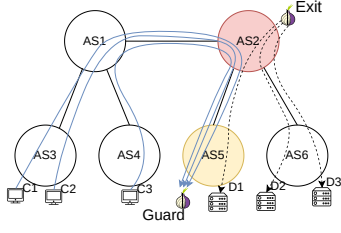
If we consider a guard $g \in \mathcal{G}$, the fully developed metric is as follows:

$$\text{NM}(g) = \text{bw}_g(g) \prod_{a \in \mathcal{A}} 1 - \left[\text{Pr}(a|g) \times \sum_{e \in \text{reach}_{\mathcal{E} \cup \mathcal{D}}(g)} \text{Pr}(a|e) \text{Pr}(e|g) \right]. \quad (6)$$

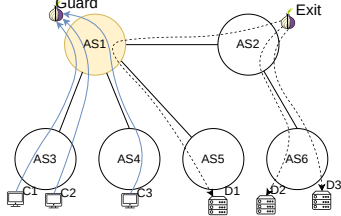
Similarly, if we consider an exit $e \in \mathcal{E}$, we obtain:

$$\text{NM}(e) = \text{bw}_e(e) \prod_{a \in \mathcal{A}} 1 - \left[\text{Pr}(a|e) \times \sum_{g \in \text{reach}_{\mathcal{G} \cup \mathcal{D}}(e)} \text{Pr}(a|g) \text{Pr}(g|e) \right]. \quad (7)$$

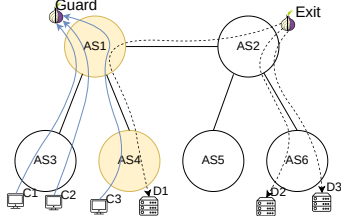
Finally, if we consider a relay $d \in \mathcal{D}$ that can be a guard or an exit relay, its total contribution is the sum of its contributions as guard and exit. As with $\text{RM}(d)$ (Equation (4)), there is no double counting of resources, as $\text{bw}_g(d)$ and $\text{bw}_e(d)$ are computed by the authorities to distribute the resources of relay d between the guard



(a) Different relay positions. AS2 can see all flows, and AS5 can see one flow.



(b) Changing the Internet topology. AS1 can correlate one flow.



(c) Changing the destination position. AS1 and AS4 can both correlate the same flow.

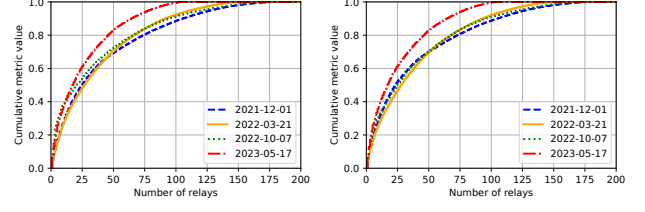
Figure 8: Examples of factors influencing the network adversary metric. Colored ASes can correlate the traffic.

and exit positions. This yields the following metric:

$$\begin{aligned}
 \text{NM}(d) = & \text{bw}_g(d) \prod_{a \in \mathcal{A}} 1 - \left[\Pr(a|d) \times \sum_{e \in \text{reach}_{\mathcal{G} \cup \mathcal{D}}(d)} \Pr(a|e) \Pr(e|d) \right] \\
 & + \text{bw}_e(d) \prod_{a \in \mathcal{A}} 1 - \left[\Pr(a|d) \times \sum_{g \in \text{reach}_{\mathcal{G} \cup \mathcal{D}}(d)} \Pr(a|g) \Pr(g|d) \right].
 \end{aligned} \tag{8}$$

The different probabilities can be computed at a given time based on public data from the consensus, knowledge of Tor’s path selection algorithm, and the AS path inference used in Section 5 once the distribution of paths is known.

Multiple network factors can influence the result of our network adversary metric. Figure 8 illustrates variants of a small example network involving a guard and an exit alongside some clients and destinations those clients intend to visit. In a perfect situation for the guard according to our metric, all possible client paths to reach the guard never traverse an AS that can also intercept traffic flowing from an exit to a potential destination. The solid blue arrows indicate all the paths used by client data to reach the guard. Similarly, the dashed arrows indicate all the paths used



(a) Relay metric.

(b) Network metric.

Figure 9: Cumulative value for the relay metric (Equations (2) to (4)) and network metric (Equations (6) to (8)). For each date, only the relays set for exclusion are considered.

by client data to reach their destination from the exit relay. All those paths traverse the topology of ASes used in this example. For each variation, we show how the guard’s metric changes due to the modification in the network. To make the example clearer, we chose not to represent more than one exit that can be paired with the guard. In practice, the metric would be computed by taking into account all exits in the network that can be paired with the guard to form a circuit.

Figure 8a illustrates how the guard’s position in the network can influence its metric. In that situation, the guard is placed in AS5. Now two ASes have a nonzero probability of being in a position to correlate traffic. AS5 can now see, with probability 1/3, the traffic on both ends. Even worse, AS2 has probability 1 of being in a position to correlate the traffic. This gives the guard a network adversary metric of $\text{NM}(g) = 0$.

Figure 8b illustrates how the Internet topology can also influence the guard’s network adversary metric. In this example, AS5 is connected to AS1 instead of AS2. This changes the path from the exit relay to destination D1. Thus, AS1 now has probability 1/3 of intercepting both ends of a circuit. For that reason, the guard’s metric is now $\text{NM}(g) = \text{bw}_g(g) \times 2/3$.

Finally, in Figure 8c, we illustrate the influence of the destination’s position in the network. Here, destination D1 has moved from AS5 to AS4. In that case, AS1 and AS4 are both able to perform traffic correlation. For AS1, the probability that both ends of a circuit traverse it equals 1/3, and for AS4 that probability equals $1/3 \times 1/3 = 1/9$. This yields a final metric of $\text{NM}(g) = \text{bw}_g(g) \times 16/27$ for the guard. Symmetrically, it is clear that the clients’ positions also contribute to that guard’s metric.

6.4 Evaluating excluded relays’ contribution

We compute both relay and network metrics for the relays set for the four exclusion rounds: 2021-12-01, 2022-03-21, 2022-10-07, and 2023-05-17 to illustrate how they can help identify important relays to keep in the network.

6.4.1 Relay adversary metric. Figure 9a shows the cumulative distribution functions of the relay adversary metric for the four exclusion rounds mentioned earlier. Each curve in the figure represents the proportion of all excluded relays’ contributions to resisting relay adversaries. For example, the curve for the exclusion round in May 2023 indicates that 25 relays contributed 60% of the total resilience to relay adversaries provided by all excluded relays.

6.4.2 Network adversary metric. Figure 9b shows the cumulative distribution for the metric over all the relays listed for exclusion. It is worth noting that, to cover 50% of the cumulative metric, it is generally sufficient to ensure that the 25 “largest” relays according to this metric are updated before the exclusion. Contacting all the relay operators is not as critical, as a few relays account for a large share of the cumulative value of the metric.

The first three exclusion rounds represented on the graph have similar curves. However, the curve for the exclusion of May 2023 is shifted to the left: a smaller set of relays accounts for the same share of the cumulative metric that will be excluded. The exclusion of May 2023 is slightly smaller than that of March 2022 (344 relays and 354 relays, respectively). As the metric has a value of zero for all middle relays, the leftward shift of the May 2023 curve is caused by a smaller number of non-middle-only relays being excluded (106 for May 2023 versus 182 for March 2022).

6.4.3 Ranking relays. We use the metrics to rank relays about to be excluded from the network. To provide insight into how the metrics can identify important relays, we list the 15 relays with the highest scores for each metric in Tables 1a and 1b. We depict only the exclusion of October 2022.

The tables in Appendix E show that relays with higher bandwidth generally appear at the top of the ranking: bandwidth is part of the computation for both metrics and is directly linked to the number of clients that the relay can handle. However, both metrics provide additional nuance: sorting by the metric does not simply sort by bandwidth. The position the relay can take in a circuit is also taken into account: most exit relays are pushed to the top of the ranking, especially in Table 1a.

It is interesting to note that the relay `xeracition` is in the top-3 in both tables with only 63% of the bandwidth of the first relay (`slauthtertodeport1`). For the utility metric (Equation (3)), this means its normalized bandwidth as an exit is much lower than `slauthtertodeport1`, but it can connect to more guard capacity, as the final metric value is close to the metric of `slauthtertodeport1`. Likewise, for the network metric (Equation (7)), `xeracition` is better at avoiding network-level adversaries than `slauthtertodeport1`: it has nearly the same metric value with significantly lower bandwidth. Another interesting relay is `Password`: this is the only relay with less than 30 MiB/s of bandwidth to be ranked in Table 1a. These results show that unusual locations of some relays have a significant positive impact on the network’s provided security against our modeled adversaries. Potentially, such a metric and ranking could be part of incentives to foster diversity.

Looking at the most popular ASes appearing in Tables 1a and 1b, one of them stands out. AS16276 (OVH, a popular French hosting provider) appears five times in Table 1a and six times in Table 1b. The popularity of OVH across relay operators does not help the relays achieve large metric values. The Tor path selection algorithm prevents any two relays hosted in the same /16 subnet (or /32 for IPv6) from being used in the same circuit. This restriction is captured in both metrics and lowers the score despite their high bandwidth. Having a large share of relays hosted in a single AS especially lowers the score for the network adversary metric, as this increases the probability of this AS being able to observe both ends of the circuit. In the particular case of OVH, the provider is

also popular for web hosting, so destinations to which Tor clients connect are also likely to be observable by this AS. For those reasons, it is reasonable that relays hosted at OVH are ranked lower in Tables 1a and 1b than other relays with similar bandwidth.

From the tables, we see that, in general, relays with a high metric value for one metric also have a high value for the other metric: 11 relays appear in both tables. However, four relays appear in only one table, indicating that the metrics effectively capture different aspects.

Finally, the human factor in relay operation is also visible: multiple relays from the same family were set for exclusion and appear in the top 15 both Tables 1a and 1b. Relays from the same family are operated by the same person or the same organization. Similarly, multiple relays were quite old (3+ years), indicating that the operator already has some experience in operating Tor relays but may suffer from the “up-to-date within the official repository but rejected from the Tor network” issue, as discussed in Section 3.

6.5 Discussion

While the metrics presented in Sections 6.2 and 6.3 are based on the consensus weight and other factors (such as network location and flags), we observed that the consensus weight is the most significant component of the metric on average. That is, a relay’s consensus weight is a good estimate for our suggested metrics in the average case, given the current Internet structure and relay distribution relative to the adversary’s position. Either may change in the future. This observation also depends on how we reason about adversaries: our analysis considers that any AS on the path might potentially be malicious. However, if the adversarial ASes or victims are known [26], the metrics may depend less on consensus weight and more on network topology and the location of those adversarial ASes and victims.

7 Limitations

To create the list of popular destinations (Section 5), we used the list produced by Tranco [40]. The list provides only domain names, whereas Tor circuits are built to reach a specific IP address. To this end, we resolved the domains found in the Tranco lists using our DNS server. Because of the prevalent use of content delivery networks (CDNs), the IP addresses obtained when resolving the domain names may depend on our location. This limitation is common to most papers using such a list [33, 46, 53] and to widely used systems for Tor simulation, such as Shadow [27, 28], which also resolve IPs from a small set of vantage points.

In addition to the CDN issue, all DNS resolutions were performed in 2024 (at the time of the experiments). Thus, domains that were popular in 2019 were resolved in 2024 and may no longer have the same IP addresses or may simply no longer exist. However, the methodology we introduce, if used by the Tor Project to analyze imminent relay rejection, would not face this limitation.

Considering this, the AS-paths constructed between the exit nodes and the destinations in our experiments may not be the same for all destinations when using another vantage point in space or time. The potential impact of this is not clear, as already discussed by other simulation tools for Tor [21].

Such limitations could be mitigated using a crowdsourced DNS measurement approach with data retention. Clients distributed across the world could periodically resolve popular domains and share the responses they obtained, as well as some network location data. Initiatives like OONI [20] or RIPE Atlas [54] could be leveraged to implement such crowdsourced data collection and to provide historical results similar to the Tor Project’s CollecTor archive [71] for Tor data.

While OONI and RIPE Atlas offer DNS resolution capabilities, their application to large-scale DNS measurement presents challenges. OONI’s domain list is relatively static, with no straightforward mechanism for frequent updates. RIPE Atlas, in contrast, requires users to earn credits (earned by running probes) which constrains query volume based on both probe availability and credit accumulation. Moreover, RIPE Atlas imposes rate limits on DNS resolution requests per probe within defined timeframes.

8 Ethical considerations

Interviews followed university guidelines compliant with GDPR, with participants giving informed consent. As stated in the consent form, audio recordings of the interviews have been deleted since the publication of this paper (September 2026), and transcriptions of the interviews will be retained for a maximum of two years after publication. Only the research team accessed the data, and participants could skip questions or withdraw, with their data destroyed if they did so. This was explained beforehand in the interview guide (Appendix C.2). Results were reported with aggregated numbers and anonymized comments.

All experiments were conducted in controlled simulators, interacting only with the live Internet for DNS resolution using institutional servers.

9 Conclusion

Since 2019, the Tor Project has rejected outdated relays after each new major release. In some cases, the number and bandwidth of rejected relays seemed concerning despite Network Health Team mitigation efforts. We study this policy’s impact (RQ 1) from relay operators’ perspective (C 1, Section 3) and via network simulations (C 2, Section 5).

Interviews with 26 Tor relay operators revealed that manual updates are common, even among large operators. Some used default package repositories for Debian and Ubuntu, despite Tor Project recommendations against this practice [73], leading to exclusion from the network even when their distributions were still supported. While operators acknowledged the security benefits of the EoL policy, some noted concerns about increased operator burden and reduced network diversity.

The operators we interviewed are probably among the most committed: the vast majority of them were never excluded from the network and would understand if one of their relays were rejected because it was too out of date. However, when analyzing the data related to the April 2024 exclusion, we note that 85% of excluded relays did not return with an updated version within a month.

Based on the simulations, we showed that past rejections had a limited immediate security impact on Tor clients. However, there is a potentially more concerning and persistent community impact

on operators, since we observe that most excluded relays did not return. We also observe that although the rejections strengthened the adversary’s success, they merely accelerated it by a few days. Our results provide evidence that the churn of existing relays is an important factor in reducing the anonymity of Tor clients.

Relay rejections may conflict with existing “up-to-date” Tor versions within OS distributions, which may cause new operators to install a Tor version that is immediately rejected. There is no simple technical solution to a problem inherently caused by policy misalignment among independent software projects. An ambitious direction may be to redesign the lifecycle of the Tor software to make it independent of the systems on which it runs. A proof-of-concept [55] supports controlling the software lifecycle by hot-swapping updates in memory via JITed bytecode, but introduces its own downsides and open questions.

We also propose an approach to measure the security contribution provided by individual Tor relays (RQ 2) using security metrics (C 3, Section 6). We design these metrics based on two commonly considered adversaries against Tor: a network adversary controlling part of the network infrastructure and a relay adversary adding malicious relay capacity to Tor. We show how the metrics can be used to rank relays about to be rejected from the network and provide insight into the most important relays in the network.

Using insights from qualitative analysis, simulations, and relay ranking, we present three key recommendations addressed to the Tor Project developers to facilitate EoL policy implementation.

- (1) **Consider not distributing Tor via official distribution package repositories known to fall out of date regularly** (e.g., Debian, Ubuntu, and possibly Alpine, based on Figure 3). This approach would prevent accidental installation of outdated Tor versions and require new operators to read Tor documentation, which advises against using the default package manager for Debian and Ubuntu [73].
- (2) **Keep the EoL policy clear for operators.** *When will my relay be excluded if I do not update? What happens next? How can I rejoin the network? How frequently am I supposed to update?* The policy should explicitly answer these questions for operators. Currently, the information is dispersed across multiple documents [63, 72] and does not address all questions from the operators’ perspective.
- (3) **Provide clear, early, and predictable notifications about upcoming rejections to affected operators.** Multiple channels (mailing lists, social networks, logs, monitoring tools) are already used, but there is documentation describing where operators can subscribe to only receive information about relay updates. The mailing lists announcing relay updates also announce other topics (e.g. Tor Browser updates) which are not relevant to relay operators. We suggest using a specific mailing list for those announcements, a dedicated hashtag on social networks, and an RSS feed.

Acknowledgments

The authors would like to thank Julien ALBERT for his input to the qualitative analysis method and the interview process.

Partially funded by the CyberExcellence project of the Public Service of Wallonia (SPW Recherche), convention No. 2110186.

References

- [1] Amagicom AB. 2020. Multihop with WireGuard. <https://mullvad.net/en/help/multihop-wireguard> Accessed: 2024-07-09.
- [2] Amagicom AB. 2025. Mullvad has partnered with Obscura VPN. <https://mullvad.net/en/blog/mullvad-partnered-with-obscura-vpn> Accessed: 2025-03-24.
- [3] Apple Inc. 2022. About iCloud Private Relay. <https://support.apple.com/en-us/102602> Accessed: July 2024.
- [4] AS 396507 (@emeraldionion@disobey.net). 2026. Looking to interview Tor relay operators. <https://disobey.net/@emeraldionion/115990763792243988>
- [5] Brice Augustin, Balachander Krishnamurthy, and Walter Willinger. 2009. IXPs: Mapped?. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*. ACM, 336–349. doi:10.1145/1644893.1644934
- [6] Armon Barton and Matthew Wright. 2016. DeNASA: Destination-Naive AS-Awareness in Anonymous Communications. *Proceedings on Privacy Enhancing Technologies* 2016, 4 (Jan. 2016). doi:10.1515/popets-2016-0044
- [7] Armon Barton, Matthew Wright, Jiang Ming, and Mohsen Imani. 2018. Towards Predicting Efficient and Anonymous Tor Circuits. In *27th USENIX Security Symposium*. USENIX Association, 429–444.
- [8] Center for Applied Internet Data Analysis based at the University of California's San Diego Supercomputer Center (CAIDA UCSD). 2024. AS Rank. doi:10.21986/CAIDA.DATA.AS-RANK Dates used: 2019-10-01, 2012-12-01, 2022-04-01, 2022-10-01 and 2023-06-01. Accessed: 2024-05-28.
- [9] Center for Applied Internet Data Analysis based at the University of California's San Diego Supercomputer Center (CAIDA UCSD). 2024. Internet eXchange Points Dataset. doi:10.21986/CAIDA.DATA.IXPS Dates used: 2019-10, 2022-01, 2022-04, 2022-10 and 2023-04. Accessed: 2024-05-28.
- [10] Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement* 20, 1 (April 1960), 37–46. doi:10.1177/001316446002000104
- [11] Jules Dejaeghere. 2026. [tor-relays] Looking to interview Tor relay operators (online or at FOSDEM). <https://lists.torproject.org/mailman3/hyperkitty/list/tor-relays@lists.torproject.org/message/ERYI5F2D4JULTLXWSOFAXJAJ4U7NM/>
- [12] Claudia Diaz, Harry Halpin, and Aggelos Kiayias. 2021. The Nym Network — The Next Generation of Privacy Infrastructure. <https://www.nym.com/nym-whitepaper.pdf>
- [13] Roger Dingledine, Nicholas Hopper, and George Kadianakis. 2014. One Fast Guard for Life (or 9 months). In *7th Workshop on Hot Topics in Privacy Enhancing Technologies*. <https://petsymposium.org/2014/papers/Dingledine.pdf>
- [14] Roger Dingledine, Nick Mathewson, and Paul Syverson. 2004. Tor: The Second-Generation Onion Router. doi:10.21236/ADA465464
- [15] Thien-Nam Dinh, Florentin Rochet, Olivier Pereira, and Dan S. Wallach. 2020. Scaling Up Anonymous Communication with Efficient Nanopayment Channels. *Proceedings on Privacy Enhancing Technologies* 2020, 3 (July 2020), 175–203. doi:10.2478/popets-2020-0048
- [16] Claudia Diaz, Stefaan Seys, Joris Claessens, and Bart Preneel. 2003. Towards Measuring Anonymity. In *Privacy Enhancing Technologies*, Vol. 2482. Springer, 54–68. doi:10.1007/3-540-36467-6_5
- [17] Pardis Emami-Naeini, Henry Dixon, Yuvraj Agarwal, and Lorrie Faith Cranor. 2019. Exploring How Privacy and Security Factor into IoT Device Purchase Behavior. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. doi:10.1145/3290605.3300764
- [18] Nathan S Evans, Roger Dingledine, and Christian Grothoff. 2009. A Practical Congestion Attack on Tor Using Long Paths. In *18th USENIX Security Symposium*. USENIX Association, 33–50.
- [19] Nick Feamster and Roger Dingledine. 2004. Location diversity in anonymity networks. In *Proceedings of the Workshop on Privacy in the Electronic Society*. ACM, 66–76. doi:10.1145/1029179.1029199
- [20] Arturo Filastò and Jacob Appelbaum. 2012. OONI: Open Observatory of Network Interference. In *2nd Workshop on Free and Open Communications on the Internet*. USENIX Association.
- [21] Florentin Rochet. 2018. parsealex.py and CDNs · Issue #63 · shadow/shadow-plugin-tor. <https://github.com/shadow/shadow-plugin-tor/issues/63> Accessed: 2026-09-09.
- [22] David Goulet. 2019. Removing End-Of-Life Relays from the Network | Tor Project. <https://blog.torproject.org/removing-end-life-relays-network/> Accessed: 2024-06-27.
- [23] Hana Habib, Sarah Pearman, Jiamin Wang, Yixin Zou, Alessandro Acquisti, Lorrie Faith Cranor, Norman Sadeh, and Florian Schaub. 2020. "It's a scavenger hunt": Usability of Websites' Opt-Out and Data Deletion Choices. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–12. doi:10.1145/3313831.3376511
- [24] IPInfo. 2026. ASN Rankings — Networks ranked by importance. <https://ipinfo.io/asn-rankings> Accessed: 2026-06-10.
- [25] Aaron D Jaggard, Aaron Johnson, Sarah Cortes, Paul Syverson, and Joan Feigenbaum. 2015. 20,000 In League Under the Sea: Anonymous Communication, Trust, MLATs, and Undersea Cables. *Proceedings on Privacy Enhancing Technologies* 1 (2015), 4–24. doi:10.1515/popets-2015-0002
- [26] Aaron D. Jaggard and Paul Syverson. 2017. Onions in the Crosshairs: When The Man really is out to get you. In *Proceedings of the 2017 on Workshop on Privacy in the Electronic Society (Dallas, Texas, USA) (WPES '17)*. Association for Computing Machinery, New York, NY, USA, 141–151. doi:10.1145/3139550.3139553
- [27] Rob Jansen and Nicholas Hopper. 2012. Shadow: Running Tor in a Box for Accurate and Efficient Experimentation. In *Proceedings of the 19th Network and Distributed System Security Symposium*. Internet Society.
- [28] Rob Jansen, Jim Newsome, and Ryan Wails. 2022. Co-opting Linux Processes for High-Performance Network Simulation. In *Annual Technical Conference*. USENIX Association, 327–350.
- [29] Rob Jansen, Florian Tschorsch, Aaron Johnson, and Björn Scheuermann. 2014. The Sniper Attack: Anonymously De-anonymizing and Disabling the Tor Network. In *Network and Distributed System Security Symposium*. Internet Society. doi:10.14722/ndss.2014.23288
- [30] Aaron Johnson, Aaron D. Jaggard, and Paul Syverson. 2023. Throwing Your Weight Around: Fixing Tor's Positional Weighting. *Proceedings on Privacy Enhancing Technologies* 2023, 4 (Oct. 2023), 593–612. doi:10.56553/popets-2023-0127
- [31] Aaron Johnson, Chris Wacek, Rob Jansen, Micah Sherr, and Paul Syverson. 2013. Users Get Routed: Traffic Correlation on Tor by Realistic Adversaries. In *Proceedings of the SIGSAC Conference on Computer and Communications Security*. ACM, 337–348. doi:10.1145/2508859.2516651
- [32] Marc Juarez, Sadia Afroz, Gunes Acar, Claudia Diaz, and Rachel Greenstadt. 2014. A Critical Evaluation of Website Fingerprinting Attacks. In *Proceedings of the SIGSAC Conference on Computer and Communications Security*. ACM, 263–274. doi:10.1145/2660267.2660368
- [33] Jordan Jueckstock, Shaown Sarker, Peter Snyder, Aidan Beggs, Panagiotis Papadopoulos, Matteo Varvello, Benjamin Livshits, and Alexandros Kapravelos. 2021. Towards Realistic and Reproducible Web Crawl Measurements. In *Proceedings of the Web Conference*. ACM, 80–91. doi:10.1145/3442381.3450050
- [34] Joshua Juen. 2012. *Protecting anonymity in the presence of autonomous system and Internet exchange level adversaries*. Ph.D. Dissertation. University of Illinois at Urbana-Champaign.
- [35] Sabrina Klivan, Sandra Höltervennhoff, Rebecca Panskus, Karola Marky, and Sascha Fahl. 2024. Everyone for Themselves? A Qualitative Study about Individual Security Setups of Open Source Software Contributors. In *2024 IEEE Symposium on Security and Privacy (SP)*. 1065–1082. doi:10.1109/SP54263.2024.00214 ISSN: 2375-1207.
- [36] Katharina Kohls and Christina Pöpper. 2018. DigsTor: Comparing Passive Traffic Analysis Attacks on Tor. In *Computer Security*, Vol. 11098. Springer, 512–530. doi:10.1007/978-3-319-99073-6_25
- [37] Georg Koppen. 2023. [tor-relays] PSA: Tor 0.4.5 reaches end of life (EOL) on 2023-02-15. <https://lists.torproject.org/pipermail/tor-relays/2023-January/020985.html> Accessed: 2024-06-13.
- [38] Alexander Krause, Harjot Kaur, Jan H. Klemmer, Oliver Wiese, and Sascha Fahl. 2025. "That's my perspective from 30 years of doing this": An Interview Study on Practices, Experiences, and Challenges of Updating Cryptographic Code. In *34th USENIX Security Symposium (USENIX Security 25)*. USENIX Association, Seattle, WA, 2907–2926. <https://www.usenix.org/conference/usenixsecurity25/presentation/krause>
- [39] Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser. 2017. *Research methods in human-computer interaction*. Morgan Kaufmann.
- [40] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczynski, and Wouter Joosen. 2019. Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. In *Network and Distributed System Security Symposium*. Internet Society. doi:10.14722/ndss.2019.23386
- [41] Akshaya Mani, T. Wilson-Brown, Rob Jansen, Aaron Johnson, and Micah Sherr. 2018. Understanding Tor Usage with Privacy-Preserving Measurement. In *Proceedings of the Internet Measurement Conference*. ACM, 175–187. doi:10.1145/3278532.3278549
- [42] MaxMind, Inc. 2022. GeoLite2 Database. <https://www.maxmind.com/en/geolite-free-ip-geolocation-data> Used data from 2022-10-28, downloaded from <https://archive.org/download/geolite2-2022>.
- [43] Mike Perry. 2016. Load Balancing with Overhead Parameters. <https://spec.torproject.org/proposals/265-load-balancing-with-overhead.txt> Accessed: Feb. 2026.
- [44] S.J. Murdoch and G. Danezis. 2005. Low-Cost Traffic Analysis of Tor. In *Symposium on Security and Privacy*. IEEE, 183–195. doi:10.1109/SP.2005.12
- [45] Steven J. Murdoch and Piotr Zieliński. 2007. Sampled Traffic Analysis by Internet-Exchange-Level Adversaries. In *Privacy Enhancing Technologies*, Vol. 4776. Springer, 167–183. doi:10.1007/978-3-540-75551-7_11
- [46] Arian Akhavan Niaki, Shinyoung Cho, Zachary Weinberg, Nguyen Phong Hoang, Abbas Razaghpanah, Nicolas Christin, and Phillipa Gill. 2020. ICLab: A Global, Longitudinal Internet Censorship Measurement Platform. In *Symposium on Security and Privacy*. IEEE, 135–151. doi:10.1109/SP40000.2020.00014

- [47] nusenu. 2019. [tor-relays] end-of-life relays will be removed from the network eventually. <https://lists.torproject.org/pipermail/tor-relays/2019-September/017711.html> Accessed: 2024-06-13.
- [48] nusenu. 2020. How Malicious Tor Relays are Exploiting Users in 2020 (Part I). <https://nusenu.medium.com/how-malicious-tor-relays-are-exploiting-users-in-2020-part-i-1097575c0cae> Accessed: 2025-03-25.
- [49] nusenu. 2021. Tracking One Year of Malicious Tor Exit Relay Activities (Part II). <https://nusenu.medium.com/tracking-one-year-of-malicious-tor-exit-relay-activities-part-ii-85c80875c5df> Accessed: 2025-03-25.
- [50] nusenu. 2026. [tor-relays] support policy vs. actual expectations beyond the end-of-life date. <https://lists.torproject.org/mailman3/hyperkitty/list/tor-relays@lists.torproject.org/message/HHO3RDX34AYXHK5Z6UEZYT6DO7PKRU7J/>
- [51] Tobias Pulls and Ethan Witwer. 2023. Maybenot: A Framework for Traffic Analysis Defenses. In *Proceedings of the 22nd Workshop on Privacy in the Electronic Society (Copenhagen, Denmark) (WPES '23)*. Association for Computing Machinery, New York, NY, USA, 75–89. doi:10.1145/3603216.3624953
- [52] Jian Qiu and Lixin Gao. 2006. AS Path Inference by Exploiting Known AS Paths. In *Globecom*. IEEE. doi:10.1109/GLOCOM.2006.27
- [53] Vera Rimmer, Theodor Schnitzler, Tom Van Goethem, Abel Rodríguez Romero, Wouter Joosen, and Katharina Kohls. 2022. Trace Oddity: Methodologies for Data-Driven Traffic Analysis on Tor. *Proceedings on Privacy Enhancing Technologies* 2022, 3 (July 2022), 314–335. doi:10.56553/popets-2022-0074
- [54] RIPE NCC. 2025. RIPE Atlas Documentation. <https://atlas.ripe.net/docs/> Accessed: 2025-08-19.
- [55] Florentin Rochet, Jules Dejaeghere, and Tariq Elahi. 2024. Towards Flexible Anonymous Networks. In *Proceedings of the 23rd Workshop on Privacy in the Electronic Society (Salt Lake City, UT, USA) (WPES '24)*. Association for Computing Machinery, New York, NY, USA, 1–16. doi:10.1145/3689943.3695038
- [56] Florentin Rochet and Olivier Pereira. 2017. Waterfilling: Balancing the Tor network with maximum diversity. *Proceedings on Privacy Enhancing Technologies* 2017, 2 (April 2017), 4–22. doi:10.1515/popets-2017-0013
- [57] Florentin Rochet and Olivier Pereira. 2018. Dropping on the Edge: Flexibility and Traffic Confirmation in Onion Routing Protocols. *Proceedings on Privacy Enhancing Technologies* 2018, 2 (April 2018), 27–46. doi:10.1515/popets-2018-0011
- [58] Florentin Rochet, Ryan Wails, Aaron Johnson, Prateek Mittal, and Olivier Pereira. 2020. CLAPS: Client-Location-Aware Path Selection in Tor. In *Proceedings of the SIGSAC Conference on Computer and Communications Security*. ACM, 17–34. doi:10.1145/3372297.3417279
- [59] RouteViews. 2024. RouteViews – University of Oregon RouteViews Project. <https://www.routeviews.org/routeviews/> Accessed: 2024-05-28.
- [60] Andrei Serjantov and George Danezis. 2003. Towards an Information Theoretic Metric for Anonymity. In *Privacy Enhancing Technologies*, Vol. 2482. Springer, 41–53. doi:10.1007/3-540-36467-6_4
- [61] Yixin Sun, Anne Edmundson, Nick Feamster, Mung Chiang, and Prateek Mittal. 2017. Counter-RAPTOR: Safeguarding Tor Against Active Routing Attacks. In *Symposium on Security and Privacy*. IEEE, 977–992. doi:10.1109/SP.2017.34
- [62] Paul Syverson. 2009. Why I'm not an entropist. In *International Workshop on Security Protocols*. Springer, 213–230. doi:10.1007/978-3-642-36213-2_25
- [63] The Tor Project. 2024. Relay EOL policy · Wiki · The Tor Project / Network Health / Team · GitLab. <https://gitlab.torproject.org/tpo/network-health/team/-/wikis/Relay-EOL-policy> Accessed: 2024-05-24.
- [64] The Tor Project. 2024. Sources – Tor Metrics. <https://metrics.torproject.org/sources.html> Accessed: 2024-05-28.
- [65] The Tor Project. 2024. Tor funding sources. <https://www.torproject.org/about/supporters/> Accessed: July 2024.
- [66] The Tor Project. 2024. Tor Protocol Specification. <https://spec.torproject.org/tor-spec/index.html> Accessed: 2024-06-07.
- [67] The Tor Project. 2024. tor-relays Info Page. <https://lists.torproject.org/cgi-bin/mailman/listinfo/tor-relays> Accessed: 2024-06-10.
- [68] The Tor Project. 2024. @torproject / X. <https://x.com/torproject> Accessed: 2024-06-10.
- [69] The Tor Project. 2024. @torproject@mastodon.social. <https://mastodon.social/@torproject> Accessed: 2024-06-10.
- [70] The Tor Project. 2024. Users – Tor Metrics: Top-10 countries by relay users. <https://metrics.torproject.org/userstats-relay-table.html> Accessed: 2024-05-27.
- [71] The Tor Project. 2025. CollecTor, your friendly data-collecting service in the Tor network. <https://metrics.torproject.org/collector.html> Accessed: August 2025.
- [72] The Tor Project. 2025. Expectations for relay operators. <https://community.torproject.org/policies/relays/expectations-for-relay-operators/> Accessed: 2026-07-10.
- [73] The Tor Project. 2026. How to install Tor. <https://support.torproject.org/little-tor/getting-started/installing/> Accessed: 2026-02-25.
- [74] The Tor Project. 2026. Middle/Guard relay – FreeBSD. <https://community.torproject.org/relay/setup/guard/freebsd/> Accessed: 2026-02-25.
- [75] Warda Usman, Jackie Hu, McKynlee Wilson, and Daniel Zappala. 2023. Distrust of big tech and a desire for privacy: Understanding the motivations of people who have voluntarily adopted secure email. In *Nineteenth Symposium on Usable Privacy and Security (SOUPS 2023)*. USENIX Association, Anaheim, CA, 473–490. <https://www.usenix.org/conference/soups2023/presentation/usman>
- [76] Gerry Wan, Aaron Johnson, Ryan Wails, Sameer Wagh, and Prateek Mittal. 2019. Guard Placement Attacks on Path Selection Algorithms for Tor. *Proceedings on Privacy Enhancing Technologies* 2019, 4 (Oct. 2019), 272–291. doi:10.2478/popets-2019-0069
- [77] Nick Weaver. 2025. [tor-relays] Accelerated depreciation of older versions? <https://lists.torproject.org/mailman3/hyperkitty/list/tor-relays@lists.torproject.org/thread/CS5UK7B35MQYGGMIGPKMOZK2JK7QGICV>
- [78] Shikun Zhang, Yuanyuan Feng, Yaxing Yao, Lorrie Faith Cranor, and Norman Sadeh. 2022. How Usable Are iOS App Privacy Labels? *Proceedings on Privacy Enhancing Technologies* (2022). doi:10.56553/popets-2022-0106

A Open science

The tools used to produce the results presented in this paper are available here online. Note that those tools were not part of the peer-review process.

<https://github.com/dont-kill-my-relay/relay-rejection>

B Generative AI usage

We used generative AI to bootstrap minor parts of the code, proof-read the text, and rephrase parts of it. Other than that, generative AI was **not** used to produce the research output or to write the paper.

C Interview process

C.1 Participant recruitment

We recruited participants via two channels: the Tor relay operators mailing list [11] and paper signs displayed at FOSDEM [4]. The consent form was attached to the email on the mailing list so participants could read it and decided for themselves to take part in interview. For participants contacting us during FOSDEM, we shared the consent form using the same channel the participant used to contact us. We answered participant questions before starting the interview. Questions raised on the mailing list were addressed by replying to the mailing list and can be found in the online archive of that list [11].

C.2 Interview guide

Below is the information provided and the questions posed to participants during the interviews.

Introduction.

- (1) Questions are open-ended and have no right or wrong answers.
- (2) You may skip any question. Simply reply “I don’t know” or “I would rather not answer”, and we will move on.
- (3) Take the time you need to reply and gather your thoughts.
- (4) You may stop the interview at any time. Simply tell me, and I will not save or use your answers.
- (5) The interview is structured in three parts:
 - (a) The first concerns your background as a Tor relay operator.
 - (b) The second concerns your installation and update process in relation to Tor.
 - (c) The last concerns the EoL policy applied by the Tor Project since 2019.

Tor relay operator background.

- (1) How many relays do you operate?

- (2) How much total bandwidth do they provide? You may specify the total advertised or the total actually used, whichever is easier for you to find or remember.
- (3) Since when have you operated these relays?
- (4) How are the relays hosted?

Installation and update process.

- (1) How did you install Tor on your relays?
- (2) How do you keep your relays up to date with the latest Tor version?
- (3) In your view, what happens between the moment the Tor Project releases a new version and the moment the update is applied to your relay? Specifically, who is involved? What are the steps?
- (4) What are the main benefits and disadvantages of this approach?

EoL policy and exclusion. At this point, and to provide all participants with the same context, the following text is read aloud verbatim: “Since 2019, Tor has been rejecting relays running outdated versions from the network. Currently, only non-EoL version series are accepted. When a version series becomes EoL, it is rejected from the network soon after. This has resulted in between one and three version series being accepted in the network since 2019.”

During the first interviews, the wording was: “Since 2019, Tor started to reject relays running outdated versions from the network. Currently, only the three most recent version series are accepted. When a new version series is released, the oldest accepted version series is rejected from the network.” A participant pointed out that this did not exactly reflect reality; hence the change.

- (1) Were you aware of this policy before starting the interview?
- (2) What is your opinion of this policy?
- (3) Has one of your relays ever been excluded from the network because it was too out dated?
 - (a) If so, how did you feel, and how did you react?
 - (b) If not, how would you feel and react if one of your relays were rejected from the Tor network?

D Factors influencing the relay metric

Multiple properties of the relays and the network can influence the value of our relay utility metric, presented in Section 6.2. Figure 10 highlights influencing factors with a small network setup. Unless otherwise specified, we assume that all relays have the same bandwidth b for this example. Also, the relay metric depends on normalized weights but is not normalized itself. As such, the sum of the metrics, even for the same set of relays, does not necessarily equal one.

In the setup depicted in Figure 10a, g_1 holds one third of the total guard bandwidth and can connect to all exits. Therefore, the following computation holds for its utility metric: $RM(g_1) = 1/3 \times 1$. The same applies to g_3 . For g_2 , it can connect to only two thirds of the exits. As expected, its utility is lower because the number of possible circuits is lower: $RM(g_2) = 1/3 \times 2/3 = 2/9$.

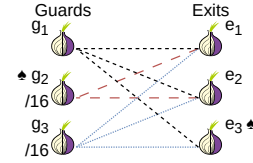
Intuitively, guards g_1 and g_3 are more useful to clients than g_2 : they provide access to one more exit relay. For the same reason, an adversary already operating in the network would prefer to see g_1 or g_3 excluded rather than g_2 , as more clients would have to re-sample relays to build new circuits. This is especially true for guards,

as they are long-lived: a client attempts to keep using the same guard for an extended period to avoid being compromised [13].

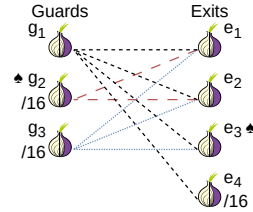
If g_2 has double the bandwidth (it now has $2b$) and other relays remain unchanged, we have: $RM(g_1) = 1/4 \times 1$, $RM(g_3) = 1/4 \times 1$, and $RM(g_2) = 1/2 \times 2/3 = 1/3$. Relay g_2 now has a higher utility metric because its bandwidth is twice that of all the other relays, although it can connect to only two thirds of the exit capacity.

Figure 10b (with all relays having the same bandwidth b) shows the importance of normalization in our metric. Adding one exit relay e_4 that can be reached only by g_1 will not affect the utility metric of g_1 : this guard will still be able to connect to all exits and holds one third of the guard bandwidth (hence its metric remains $RM(g_1) = 1/3 \times 1$, as in Figure 10a). However, the metric for both g_2 and g_3 is now lower: neither can connect to e_4 , as they are hosted in the same subnet. Their metrics are now, respectively, $RM(g_2) = 1/3 \times 1/2 = 1/6$ and $RM(g_3) = 1/3 \times 3/4 = 1/4$.

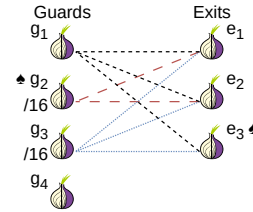
Finally, Figure 10c shows how adding a guard lowers the utility of all other guards compared to the setup in Figure 10a. Assuming that the new guard also has a bandwidth of b , g_1 and g_3 have a metric of $1/4 \times 1$, and g_2 has a metric of $1/4 \times 2/3 = 1/6$. As there are now more guards in the network, the utility of each guard is relatively lower. Similarly, the marginal success increase of an already operating adversary will be lower, as clients will be distributed over more guards.



(a) Initial network setup to evaluate the relay utility metric.



(b) Adding an exit relay e_4 decreases the utility of guards g_2 and g_3 .



(c) Adding a new guard relay g_4 decreases the utility of all guards.

- ♠ Relays share the same family
- /16 Relays share the same subnet

Figure 10: Examples of factors influencing the relay adversary metric.

(a) Top-15 relays ranked by relay utility metric (RM)

(b) Top-15 relays ranked by network adversary metric (NM)

Table 1: Top-15 relays ranked by each metric for the exclusion of October 2022. **Blue and bold** relay names appear in both Tables 1a and 1b. The first column indicates the rank of the relay in the consensus when sorted by the metric value. GeoIP data from MaxMind, Inc. [42].

Figure 11 shows which ASes are in a position to observe both ends of a circuit. Note that the top-2 (AS6939 and AS3359) and the top-3 (AS6939, AS3359, and AS1299) are stable across the four dates considered. The ASes identified by our simulations are consistent with the top-ranked infrastructure ASes identified by IPInfo [24].

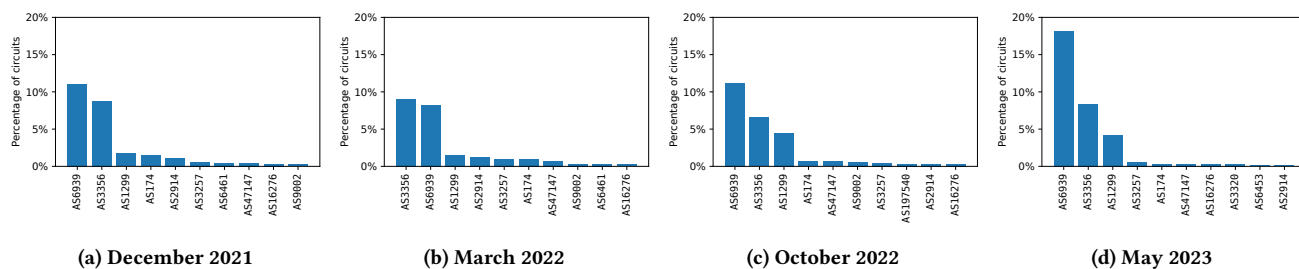


Figure 11: Top-10 ASes capable of observing circuits on both ends for the four exclusion dates considered. The percentage of circuits is the fraction of simulated circuits on which the AS is found on both ends.