Users Get Routed: Traffic Correlation on Tor by Realistic Adversaries

Aaron Johnson[1] Chris Wacek[2] Rob Jansen[1] Micah Sherr[2] Paul Syverson[1]

[1]U.S. Naval Research Laboratory, Washington DC
{aaron.m.johnson, rob.g.jansen, paul.syverson}@nrl.navy.mil

[2]Georgetown University, Washington DC
{cwacek, msherr}@cs.georgetown.edu

ABSTRACT

We present the first analysis of the popular Tor anonymity network that indicates the security of typical users against reasonably realistic adversaries in the Tor network or in the underlying Internet. Our results show that Tor users are far more susceptible to compromise than indicated by prior work. Specific contributions of the paper include (1) a model of various typical kinds of users, (2) an adversary model that includes Tor network relays, autonomous systems (ASes), Internet exchange points (IXPs), and groups of IXPs drawn from empirical study, (3) metrics that indicate how secure users are over a period of time, (4) the most accurate topological model to date of ASes and IXPs as they relate to Tor usage and network configuration, (5) a novel realistic Tor path simulator (*TorPS*), and (6) analyses of security making use of all the above. To show that our approach is useful to explore alternatives and not just Tor as currently deployed, we also analyze a published alternative path selection algorithm, Congestion-Aware Tor. We create an empirical model of Tor congestion, identify novel attack vectors, and show that it too is more vulnerable than previously indicated.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*

Keywords

Anonymity; metrics; onion routing

1. INTRODUCTION

Tor is a volunteer-operated anonymity network that is estimated to protect the privacy of hundreds of thousands of daily users [13, 22]. However, Tor is known to be insecure against an adversary that can observe a user's traffic entering and exiting the anonymity network. Quite simple and efficient techniques can correlate traffic at these separate locations by taking advantage of identifying traffic patterns [29]. As a result, the user and his destination may be identified, completely subverting the protocol's security goals.

CCS'13, November 4–8, 2013, Berlin, Germany.

ACM 978-1-4503-2477-9/13/11 ...$15.00.
http://dx.doi.org/10.1145/2508859.2516651.

The traffic correlation problem in Tor has seen much attention in the literature. Prior Tor security analyses often consider entropy or similar statistical measures as metrics of the security provided by the system at a *static point in time*. In addition, while prior metrics of security may provide useful information about *overall* usage, they typically do not tell users how secure a *type of behavior* is. Further, similar previous work has thus far only considered adversaries that control either a subset of the members of the Tor network, a single autonomous system (AS), or a single Internet exchange point (IXP). These analyses have missed important characteristics of the network, such as that a single organization often controls several geographically diverse ASes or IXPs. That organization may have malicious intent or undergo coercion, threatening users of all network components under its control.

Given the severity of the traffic correlation problem and its security implications, we develop an analysis framework for evaluating the security of various user behaviors on the live Tor network and show how to concretely apply this framework by performing a comprehensive evaluation of the security of the Tor network [41] against the threat of complete deanonymization. To enable such an analysis, we develop a detailed model of a network adversary that includes (*i*) the largest and most accurate system for AS path inference yet applied to Tor and (*ii*) a thorough analysis of the threat of Internet exchange points and IXP coalitions. We also develop realistic metrics that inform this analysis, considering the network topology as it *evolves over time*, for example, as new relays are introduced and others go offline.

Our analysis shows that 80% of all types of users may be deanonymized by a relatively moderate Tor-relay adversary within six months. Our results also show that against a single AS adversary roughly 100% of users in some common locations are deanonymized within three months (95% in three months for a single IXP). Further, we find that an adversary controlling two ASes instead of one reduces the median time to the first client de-anonymization by an order of magnitude: from over three months to only 1 day for a typical web user; and from over three months to roughly one month for a BitTorrent user. This clearly shows the dramatic effect an adversary that controls multiple ASes can have on security.

We observe that since the relays that comprise Tor's egress points may independently specify IP and port-based access control policies, the set of relays available for anonymous circuits is dependent on the user's application (web browsing, IRC, BitTorrent, etc.). We examine how this choice of application affects the security of the user's anonymous connections. Our analysis shows that BitTorrent users not only degrade performance of the Tor network for everybody else, but against a Tor-relay adversary they get significantly less anonymity protection than typical users. They are bested for

least anonymity among the uses we considered only by users of the collaborative-work real-time editor Gobby [1].

After describing background and related work, we next set out our adversary model and security metrics. We then describe our user models and the use of Monte Carlo simulation to sample how user traffic flows over the network, using our Tor Path Simulator (*TorPS*) to generate paths. We describe a newly-introduced Internet map that we use in the subsequent section to evaluate the security of circuits created via TorPS against a network adversary, after having analyzed security against a Tor-relay adversary. Finally we demonstrate the applicability of our approach beyond evaluation of the current Tor network by analyzing Congestion-Aware Tor [45], a system that attempts to improve Tor network performance by measuring relay congestion and avoiding the most congested parts of the network when sending application traffic.

2. BACKGROUND

The Tor network consists of roughly 3000 *relays* pushing over 2500 MiB/s in aggregate [40]. Tor clients select three of these relays to form a *circuit* through which they create TCP *streams* to communicate with external Internet destinations. Tor measures the real *bandwidth* (throughput over time) that each relay provides to the network, and assigns each relay a selection *weight* based on the bandwidth it provides. These weights are used to bias selection for circuits in order to distribute load toward relays with more available network resources. Relays may specify a bandwidth allowance over a desired time period: once the allowance is reached, the relay will hibernate until the end of the time period. Hibernating relays will neither participate in building new circuits nor transfer data for Tor.

Relays have status flags assigned to them by the *directory authority*, which clients consider when choosing relays for a circuit. The GUARD flag is assigned to relays whose uptime is at least the median for familiar relays, and if their bandwidth is at least the minimum of 250 KiB/s and the median relay bandwidth. Clients choose and maintain three active *guards* and use them as the entry relay for all of their circuits to reduce the chance of directly connecting to an adversary. Clients rotate each guard at a random time between 30 and 60 days. The EXIT flag is assigned to relays who allow direct connections with external Internet destinations. Exits set individual *exit policies* specifying the IP address ranges and port ranges to which they are willing to connect. Clients use these policies to determine which relay to choose for the final position in each circuit. Guards and exits are more-highly weighted for the entry and exit position in a circuit, respectively, as not all relays fulfill the requirements to obtain those flags. Additionally, a relay obtains the STABLE flag if its weighted mean time before failure is at least the median for known active relays. Clients building streams to a port in the *long-lived ports* list must choose stable relays in each position of the circuit. Finally, clients will never choose two relays from the same /16 subnet or *family* for the same circuit. A family is a set of relays that mutually indicate that they belong to a group together.

3. RELATED WORK

Anonymity systems have received significant study since Chaum's [11] seminal work on untraceable email in 1981. We highlight the most relevant methods for measuring anonymity and discuss many of the threats to anonymity systems.

Metrics and Methods for Evaluating Anonymity. Serjantov and Danezis [34] and Díaz *et al.* [12] independently propose evaluation frameworks that quantify anonymity using Shannon entropy computed over a set of potential senders (or receivers). Hamel *et al.* argue against entropy-based metrics and instead focus on how an adversary's actions can compromise anonymity [23]. They envision an adversary with a fixed bandwidth budget, and explore how the adversary can spend that budget to compromise anonymity. Syverson *et al.* also describe a bounded adversary and present a model in which the adversary can corrupt a fixed number of routers within a time period [38], using probabilistic analysis to quantify the resulting level of anonymity. Similar to this latter model, we assume the existence of a fixed adversary who either controls some relays ("Relay Adversary") or monitors a portion of the Internet such as an AS or IXP ("Network Adversary").

Elahi *et al.* [16] construct a simulation-based framework for measuring how well Tor's guard selection mechanism defends against profiling attacks [47]. Similar to our techniques, their *Changing of the Guards* simulator also uses data collected from the live Tor network [40] to repeatedly simulate the behavior a client. Their simulation study focuses on guard selection and adversarial relays. In contrast, this paper explores Tor's vulnerability to *traffic correlation attacks* (explained next) using various profiles of client behavior, adversary models, security metrics, and topological models of the Tor network.

Traffic Correlation Attacks. Onion routing is vulnerable to an adversary who can monitor a user's traffic as it enters and leaves the anonymity network; correlating that traffic using traffic analysis links the observed sender and receiver of the communication. Øverlier and Syverson first demonstrated the practicality of the attack in the context of discovering Tor Hidden Servers [32]. Later work by Murdoch and Danezis show that traffic correlation attacks can be done quite efficiently against Tor [29].

Given the potential severity of traffic correlation attacks, this paper explores in depth users' vulnerability to such attacks in the live Tor network. To quantify the anonymity offered by Tor, we examine path compromise rates and how *quickly* extended use of the anonymity network results in compromised paths.

Network Adversaries. Feamster and Dingledine first investigate the ability of AS-level adversaries to observe both sides of anonymous paths [19]. They argue that geographically diverse paths may adversely affect anonymity since paths that traverse many ASes are more likely than shorter paths to have the same AS on both sides of the path. Edman and Syverson also explore AS path diversity on Tor and introduce an AS-aware path selection algorithm that uses "snapshots" of Tor's AS graph to avoid AS-level traffic correlation attacks [15]. More recently, Akhoondi *et al.* propose a geographic-based relay selection method called LASTor [3] that ensures AS diversity in selected paths by relying on concise Internet atlases. A recent study by Wacek *et al.* indicates that the same AS may appear in both sides of as many as 18% of anonymous circuits [44].

Murdoch and Zieliński argue that ensuring AS diversity in anonymous circuits is insufficient to safeguard against traffic correlation attacks by network adversaries, since traffic is routed between ASes at IXPs (and hence a single IXP may observe traffic traversing multiple ASes) [30]. They apply a Bayesian approach to show that an adversary positioned at an IXP could sample traffic from multiple ASes and correlate flows. Juen proposes a refined relay selection algorithm that provides both AS and IXP diversity [28]. We remark that Tor does not currently implement any protection against adversaries who operate ASes or IXPs.

By considering how often *any* AS appears on both sides of circuits, these works implicitly assume that *all* ASes are malicious but are non-colluding. We also examine Tor's vulnerability to network

adversaries, but improve upon existing work by modeling a more realistic adversary who monitors a fixed set of ASes or IXPs.

In this paper, we do not consider circuit clogging, network latency, or application[1] or other attacks against Tor (cf. [2, 5, 8, 14, 18, 24, 29]). A comprehensive evaluation of *all* potential threats against Tor is beyond the goals of this paper. Instead, we study in depth a particular and well-understood threat against Tor—traffic correlation attacks by either malicious relay operators or networks that monitor traffic as it enters and exits Tor.

4. SECURITY MODEL AND METRICS

We present here realistic and useful adversary models and security metrics for the threat of traffic correlation in onion routing. In particular, we consider the types and amounts of adversary resources, as well as how he may use them. We argue that security metrics should be defined in terms of such adversaries and should present the probabilities of compromise over time. By applying these methods, we will be able to obtain novel and realistic quantitative estimates of Tor security against traffic correlation.

4.1 Adversary Model

In general we consider it realistic that an adversary can observe, delay, alter, drop, or add communication in a variety of ways. As will become clear from our analysis, however, an adversary that merely passively observes can be significant and illuminating. We limit our description and attentions herein to a passive end-to-end correlating adversary: one that learns source or destination of communication when in position to observe either or both of these and that always links observations of the same communication flow anywhere in its path. This linking occurs regardless of how the flow's appearance may have changed en route. We do consider an adversary that may actively add network resources or corrupt existing resources. But we do not consider any addition, alteration, or disruption of network traffic directed over those added adversary resources. Nor do we consider adversarial removal or degradation of network resources in this paper.

Adversary Resources. Our adversary is assumed to have one or more types of resources at his disposal. Tor relays themselves are an obvious resource, although it is useful to further specify if the adversary observes guard, middle, or exit relays. Besides relays themselves the most obvious possible adversary resource is the destination server. At a somewhat more abstract level, an adversary may control an amount of bandwidth. This could represent either a portion of the existing network or a resource that the adversary can add to the network by contributing additional relays. In this paper we do not consider adversarial bridges [36].

Tor and other low-latency anonymous communications networks are overlays above the transport layer. The primary organizational unit for managing Internet routing below the overlay is the autonomous system (AS). An adversary may control one or several ASes and is assumed in that case to observe any traffic entering or leaving the AS. Another potential adversarial resource is Internet Exchange Points (IXes or IXPs), which are increasingly common facilities that allow exchange of traffic between ASes, usually at a cost savings or performance improvement vs. sending via an upstream traffic provider. An IXP is in a position to see all traffic flowing between its peered ASes. It is typically in a single geographic location while an AS is often geographically distributed, and thus it would seem only more likely to be under adversary control than an AS.

Resource Endowment. We give the adversary certain endowments of the resource types. We view adversarial institutions – such as corporations, intelligence agencies, or countries – as the endowment of the resources they control. For example, the adversary might control all of the ASes in a given country. If the source or destination ISP is under adversary control, this puts the ISP AS in the set of adversary assets. If the ISP controls multiple ASes, these could all be considered adversary assets depending on details of how the ISP is under adversary control. Similarly companies that are not end-user ISPs may control multiple ASes [7]. A single company could also run multiple IXPs. For example, Equinix has 19 IXPs in 17 metropolitan areas worldwide [17].

The fraction or number of individual relays has always been a measure of adversary endowment for onion routing systems and is the basis of the c^2/n^2 risk of individual path compromise (where c is the number out of n relays that are compromised) [20, 38]. But the Tor path selection protocol weights relay choice by the bandwidth relays offer. Bandwidth is thus a more accurate measure of adversary endowment for Tor [6]. Whether using number of relays or relay bandwidth, type of relay is also a factor. Of the approximately 3000 current Tor relays, roughly a third are flagged to be stable and fast enough to serve as entry guards and roughly a third are considered exits, where these amounts each include relays that are both exits and guards. For an end-to-end correlating adversary that controls Tor relays, guards and exits are of primary importance. Combining these leads to guard and/or exit bandwidth as a still more accurate measure of adversary endowment. Adding other routing criteria to Tor could affect the impact of adversarial relay endowment in other ways. For example, in latency-aware routing [35], an adversary with more compromised exits near popular destinations or likely destinations of a given target source will be more effective than one with the same exit endowment distributed differently.

We will consider adversary goals presently, however, we can note now that allocation of adversary endowment is important to adversary success. We will discuss in Section 6.1 advantageous allocation of adversary relay bandwidth among guards and exits. This allocation could be by chance or it could be that the adversary has the capability and knowledge to strategically allocate resources. Analysis of dynamic and responsive strategic allocation of adversary resources against onion routing communication predates Tor itself. Such responsive allocation might be in order to compromise existing circuits [38] or to increase the likelihood that future communication will be compromised [6]. Against Congestion-Aware Tor [45], an adversary might generally mask congestion at a controlled relay by variable padding of processing time, which would increase the overall fraction of circuits using adversary relays. But it is also possible to do more targeted attacks, for example, prioritizing service for circuits of detected targeted clients to reduce their experience of congestion. We leave to future work, however, analysis in the presence of a dynamic, responsive, strategic adversary.

Adversary Goals. Much prior analysis of onion routing security has been against an adversary with the primary goal of deanonymizing (linking source and destination) as many circuits as possible. It is likely, however, that real adversaries will be more focused. For example, an adversary may wish to compromise as many circuits as possible for a given user or a given class of user. Or the adversary may wish to identify as many destinations as possible for a given user or class of user. (Note that these need not be coextensive goals. For example, the user might make a large majority of

[1]We do, however, show how the choice of application may influence the user's susceptibility to traffic correlation attacks.

connections to a few destinations, and the adversary wishes to also know those destinations the user visits rarely.) Or the adversary may wish to compromise circuits connecting to a given destination or set of destinations. An adversary may simply wish to know if specific sources and destinations ever connect at all or ever connect during a critical time period.

4.2 Security Metrics

Security metrics in general, and for traffic security in particular, should be defined with respect to a specific adversary, should be usable for assessing security over human timescales, and should allow estimation of probability of all reasonably-likely relevant events [37]. Also, most metrics give information either about the system itself or summarize usage of the system. Examples of these are compromised fraction of a network resource, entropy, min-entropy, or compromised fraction of all circuits using the network. These are important but not ideal for a user who would like to know, "If I use the system in the following way, how secure am I?" or "How much can I do the following while maintaining security at least to level foo?" With this in mind, we use the following metrics in our analysis:

1. The probability distribution on number of path compromises for a given user (in a given period).
2. The probability distribution on time until first path compromise.

While there are many other interesting and valuable metrics along these lines, we believe that these are particularly pertinent to the typical user of Tor. We evaluate these with respect to the adversaries described in Section 4.1.

5. METHODOLOGY

We evaluate the security of the Tor network with respect to the adversaries and metrics that we have proposed. This requires estimation of the probabilities of security events. To do so, we use the Monte Carlo method to sample how user traffic flows over the network during various types and amounts of user activity. For each sample, we use a model of the Tor network, simulate the user behavior, and simulate the resulting Tor client software actions. We evaluate the user anonymity of these samples against relay and network adversaries.

5.1 Path Simulator

To enable our evaluation, we built the TorPS path selection simulator [42], which uses historical network data to recreate the conditions under which clients operated in the past and then executes path selection algorithms over those conditions given user actions. TorPS includes a model of the Tor relays and their past states, a model of user behavior, and a model of the Tor client[2]. For each sample simulation, it takes streams produced by the user model and network states from the network model and uses them as input to the client model, which chooses circuits and assign streams to them.

5.1.1 Tor Network Model

TorPS uses data from Tor Metrics [40] to model the past states of the Tor network. Tor Metrics provides archives of network consensuses and server descriptors, which TorPS uses to determine relay status over time, including flags, exit policies, hibernation state, and other parameters. Relays that do not appear in a consensus or do not have a descriptor are taken to be inactive.

[2]TorPS is based on the code in Tor version 0.2.3.25.

Rank	Port #	Exit BW %	Long-Lived
1	8300	19.8	Yes
2	6523	20.1	Yes
3	26	25.3	No
65312	993	89.8	No
65313	80	90.1	No
65314	443	93.0	No

Table 1: Default-accept ports by exit capacity.

Model	Streams/week	IPs	Ports (#s)
Typical	2632	205	2 (80, 443)
IRC	135	1	1 (6697)
BitTorrent	6768	171	118
WorstPort	2632	205	1 (6523)
BestPorst	2632	205	1 (443)

Table 2: User model stream activity.

5.1.2 User Model

In order to understand the security of real users, we develop five models of Tor network use, which each consist of a sequence of Tor streams and the times at which they occur. Streams here include DNS resolution requests in addition to TCP connections to specific destinations. We construct three of our models by using client applications on the live Tor network and tracing the behavior of our local Tor client. Each trace consists of 20 minutes of a prescribed activity. The five user models we evaluate are as follows:

Typical. This model is designed to represent average Tor use. It uses four traces consisting of (*i*) Gmail / Google Chat, (*ii*) Google Calendar / Docs, (*iii*) Facebook, and (*iv*) web search activity. These traces are played every day during the desired period, with one session of (*i*) at 9 a.m., one session of (*ii*) at 12 p.m., one session of (*iii*) at 3 p.m., and two sequential sessions of (*iv*) starting at 6 p.m.

IRC. This model represents the use of Tor for the repeated but exclusive purpose of IRC chat. It uses the trace of a single IRC session and plays the trace sequentially from 8 a.m. to 5 p.m., Monday through Friday, a total of 27 times each day.

BitTorrent. This model represents using BitTorrent over Tor. It consists of activity during the download of a single file. The model replays the trace sequentially from 12 a.m. to 6 a.m. on Saturday and Sunday, totaling 18 replays each day.

WorstPort. This model modifies the Typical model by replacing the port numbers with 6523, which is a port used by the "Gobby" collaborative real-time editor [1]. As Table 1 shows, 6523 is supported by the second-least amount of exit capacity, excluding ports that are rejected in the default Tor exit policy. Port 6523 is interesting because it was recently added to the "long-lived ports" by request [39], indicating that it is in active use. Connecting to ports designated by Tor as long-lived requires using `Stable` exit relays, which must have a higher minimum uptime.

BestPort. This model modifies the Typical model by replacing the port numbers with the HTTPS port 443, which, as shown in Table 1, is supported by the largest amount of exit capacity.

Table 2 shows the number of streams, unique IP addresses, and unique ports that appear in each of the user models. The Typical model contacted a large number of IP addresses relative to the number of streams, but many of these were from subnets used by Facebook or Google. The IRC trace only contacted `irc.oftc.net`. The BitTorrent trace used a large number of ports, as the client chose peer ports randomly.

While these models are limited and somewhat artificial, we believe that they actually allow for good estimates of our metrics for many users. The most relevant properties of user activities are their number, duration, and destinations. Our user models explore each of these parameters over a reasonable range, in particular exploring very good and very bad ports. Moreover, our use of traces exposes how some popular applications behave according to these parameters, which provides insight into how whole classes of activity are likely to act.

5.1.3 Tor Client Model

TorPS faithfully mimics the behavior of Tor client software for creating exit circuits, taking into account features significant to path selection, such as: bandwidth weighting; relay hibernation; guard selection and rotation; exit policies; family and /16-subnet conflicts; and DNS resolution. A Tor Metrics consensus and its corresponding descriptors are used as if they were retrieved by the client when the consensus was published. In a slight deviation from Tor's current operation, we use full server descriptor to evaluate a relay's exit policy rather than use the microdescriptor format. In addition, we do not consider hidden services or bridges, although our methods could easily be used to evaluate the security of both systems.

By default, TorPS does not account for any side effects stemming from underlying network performance. That is, when evaluating basic Tor path selection it behaves as if each circuit construction succeeds, each circuit experiences the same performance, and circuits do not fail while being used. Section 7 discusses an extension to the basic simulator in which network congestion and delays are taken into account.

5.1.4 Statistical Inference

We use the empirical distribution function that results from TorPS simulations to infer the probabilities of security events. Let n be the number of TorPS samples, and let $D(x)$ be the absolute difference between the empirical CDF and the true CDF at x. The Dvoretzky–Kiefer–Wolfowitz Inequality [46] gives that $Pr[\sup_x D(x) > \epsilon] \leq 2e^{-2n\epsilon^2}$.

With all of our simulations we use either $n = 100000$ or $n = 50000$. Thus the probability that the CDF of any of our simulations has error of more than 0.01 at any point is at most $2e^{-10} \approx 9.1 \times 10^{-5}$. We infer fewer than 50 distributions, and therefore the probability that any one of them has an error at any point of more than 0.01 is less than 0.0046 by the union bound.

5.2 Internet Map

We construct a detailed Internet map in order to evaluate the security of circuits produced by TorPS against a network adversary that can observe or control pieces of the network infrastructure, such as the network links, routers, and facilities that host this equipment. This map, combined with path inference algorithms, allows us to identify the autonomous systems and internet exchange points traversed by our simulated Tor users.

We construct the network map at the AS level from two sources. First, we consider links contained within BGP paths gathered during March 2013 from eight geographically distributed *RouteViews* routers [43]. We then supplement those links with additional ones identified from traceroutes in the CAIDA *IPv4 Routed /24 AS Links Dataset* from December 2012 [9]. This combined dataset produces a graph consisting of 44605 ASes connected by 305381 links.

We use a layered approach to obtain a near-complete set of relationship information for the AS links contained in our graph. First, we apply the heuristic algorithm originally suggested by Gao [21] to our network graph. We then use relationships identified in the CAIDA AS Relationships Dataset for July 2012, overwriting any relationships previously identified through Gao's algorithm as necessary [10]. Finally, we use a set of sibling relationships heuristically identified from similarities in RIPE WHOIS records to correct misclassified sibling relationships. This approach results in relationship assignments for 88% of the links in our dataset; those links without relationships come primarily from the CAIDA IPv4 Routed /24 AS Links dataset, to which Gao's algorithm cannot be applied. We exclude links with missing relationships from our path inference algorithm, which we describe next.

5.2.1 Autonomous System Path Inference

When considering autonomous systems as adversaries, those with the capability to deanonymize Tor traffic are those which exist upon the AS path between the client and guard *as well as* between the exit and destination. For each simulated client stream, we compute the AS path from client to guard and exit to destination using the algorithm proposed by Qiu which extends known AS paths drawn from BGP tables to all ASes using a shortest path variant [33]. We use these computed paths in our network adversary analysis in Section 6.2.

5.2.2 Internet Exchange Point Map

In addition to autonomous systems, we are interested in the prevalence of Internet exchange points as another network administrative domain which could compromise Tor circuits.

The IXP Mapping Project [4] gathers data about IXPs across the Internet, and seeks to identify the ASes which peer at each IXP. We use the IXP peerings dataset they provide to identify locations along inferred AS paths where traffic transits through an IXP. The dataset contains 58524 AS peers which connect through 199 distinct IXPs. These peers represent 19.1% of the links in our network map. In cases where multiple IXPs exist as peering locations between ASes, we include both IXPs. While this may slightly overstate the ability of IXPs to compromise streams, selecting one at random for each path may understate their ability. Our analysis in Section 6.2 will focus on the potentially stronger adversary.

6. SECURITY ANALYSIS OF TOR

We evaluate the security of the Tor network against a range of plausible adversaries and with respect to several metrics, with the goal of yielding concrete numbers that are highly informative and relevant to end users.

We consider two general types of adversaries. The first is an adversary that has the resources to run relays in the Tor network. Specifically, we take bandwidth – both upload and download – to be the limiting resource and consider an adversary that allocates that bandwidth to Tor relays in order to deanonymize Tor users. The second adversary is a network operator able to observe some portion of the underlying network over which Tor traffic is transported.

6.1 Relay Adversary

Adversaries who run relays represent the most plausible and well understood threat to Tor users. Tor relays are run by volunteers and the Tor Project applies no restrictions on operators. Clients select relays for circuits roughly in proportion to relay bandwidth, and thus the amount of traffic that an adversary is in a position to deanonymize is essentially only limited by the adversary's bandwidth. Bandwidth comes at a cost, however, and the Tor network is large enough that overwhelming the Tor network could be expen-

Rank	Bandwidth (MiB/s)	Largest family member
1	260.5	herngaard
2	115.7	chaoscomputerclub19
3	107.8	ndnr1
4	95.3	GoldDragon
5	80.5	Paint

Table 3: Top Tor families, 3/31/03 23:00. Bandwidth is minimum of average and observed.

sive. Thus we seek to establish an adversary with significant but reasonable bandwidth at its disposal.

We suppose that the adversary is able to contribute 100MiBps to the Tor network. Table 3 shows the top five families listed in the last consensus in March 2013, ordered by the smaller of their average and observed bandwidths (self-reported in server descriptors) and represented by the relay with the largest consensus bandwidth. We can see that several organizations already contribute on the order of 100MiB/s to Tor. Bandwidth need not be provided by a single relay, so an adversary could supply that bandwidth by controlling a large botnet or by pooling the resources of a malicious collective. Furthermore, as consumer broadband speeds continue to increase [31], the cost for an adversary to pose a serious threat to the Tor network will continue to decrease.

Tor has a non-trivial process for assigning selection weights in the consensus that involves independently measuring node performance and then applying a proportional integral derivative (PID) feedback controller to minimize selection weight oscillation. Rather than simulate this process, we use the fact that the "observed" bandwidth numbers that relays report in their descriptors are correlated with their consensus weights. We use linear regressions on the relays in consensuses during the simulation period to convert the bandwidths of the adversary's relays to consensus weights. We use separate regressions for guard relays and exit relays, which result in correlations of determination of $r^2 = .71$ and $r^2 = .69$, respectively.

Adversary Resource Allocation. The adversary must determine how best to allocate his bandwidth to maximize the chance of compromising streams. Because the same relay cannot be chosen twice on a circuit, he must run at least two relays in order to perform a correlation attack. We therefore suppose that the adversary targets one as a guard and one as an exit. We assume the malicious guard relay provides enough uptime to obtain the GUARD flag. We assume that the malicious exit relay is not allowed to obtain the GUARD flag and is given an exit policy that allows exit to all addresses and ports. Both relays will have sufficient bandwidth to obtain the FAST flag.

To determine a good bandwidth allocation between the guard and exit relay, we ran five experiments using TorPS that varied the allocation of 100MiB/s of bandwidth between the guard and exit relays. We tested guard-to-exit bandwidth ratios of 1:1, 2:1, 5:1, 10:1, and 50:1. Clients used the Typical user model over a simulated six-month period from October 2012 through March 2013. The results, displayed in Figure 1, show that the expected rate of exit compromise decreases as more bandwidth is allocated to the guard. Thus an adversary must trade off between the likelihood of obtaining a guard position and the volume of exit traffic seen. A 5:1 guard-to-exit ratio maximized the probability of compromising both sides of at least one stream during the simulation period, so we adopt this ratio for the remainder of our experiments, as would a strategic adversary.

Allocating more bandwidth to guards makes sense for the adversary because, in the consensuses we use, exit-only relays are given

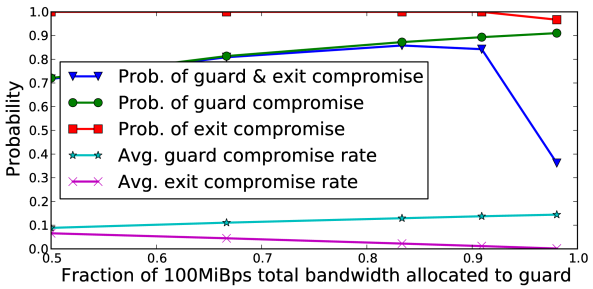


Figure 1: Probability to compromise at least one stream and rate of compromise, varying bandwidth allocation between guard and exit, 10/2012 – 3/2013.

a higher weight for use as an exit than guard-only relays are given for use as a guard. In addition, obtaining a guard is far more important for compromising the stream of a given user, as clients choose new guards much less frequently than new exits.

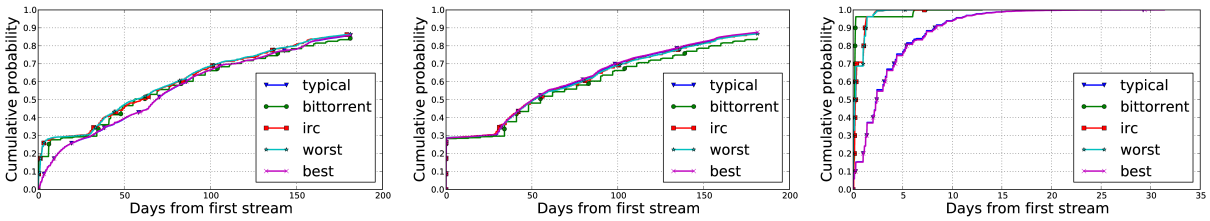
6.1.1 *Analysis*

We consider how different user behavior can have different security implications. For instance, sending many streams over Tor induces higher rates of circuit creation, increasing the number of chances the adversary has to compromise one. Alternatively, the specific destination addresses and ports that users connect to affect the probability a malicious exit is chosen because allowed exit policies differ from relay to relay.

We use TorPS to conduct simulations using each of the user models (described in Section 5.1.2) over a period from October 2012 to March 2013. We use several metrics to evaluate the security of those users against an adversary who runs one guard relay and one exit relay with 83.3 MiB/s and 16.7 MiB/s of bandwidth respectively. The results are shown in Figure 2.

Overall, we can see (Figure 2a) that in all user models there is more than an 80% chance of deanonymization within 6 months by a malicious guard and exit. The median time to full compromise is always less than 70 days. We also see that risk rises steadily over time. By looking separately at the times at which a guard or exit is compromised (Figures 2b and 2c), we see that the time it takes to choose a malicious guard, with a median of 50–60 days, dominates the time to choose a malicious exit, with a median of fewer than 2.5 days. This supports the suggestion of Elahi *et al.* [16] that the main impediment to full deanonymization by the adversary is being chosen as a guard by a given user. This implies that an adversary that observes the user's connection to the guard, such as an ISP, deanonymizes the destination much quicker than an adversary observing exit traffic, such as a malicious destination, deanonymizes the source.

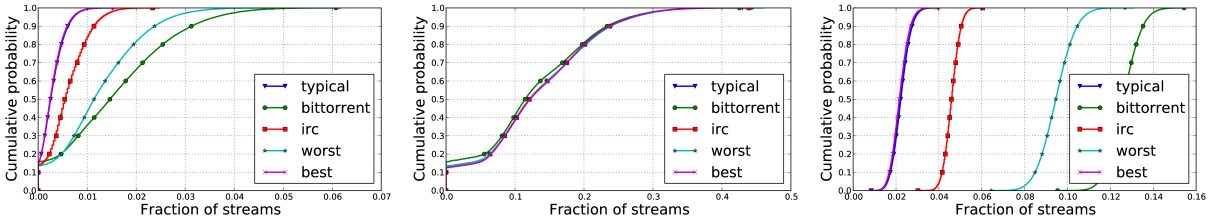
That an adversary compromises some streams is significant, but how many he compromises is just as important. Figure 2d shows median rates of full compromise between 0.25% and 1.5%, depending on user behavior. Rates of full compromise are roughly the product of the rates of exit and guard compromise, and thus are an order of magnitude lower. Guard compromise rates are generally the same for all users, as we would expect given that the destination address and port are not considered when selecting guards. The guard compromise rates show some bimodality, which corresponds to the event that the malicious guard is chosen again after expiring. Exits are chosen independently for each new circuit, and thus the exit compromise-rate distribution is roughly normal with a mean of the fraction of exit bandwidth provided by the adversary.



(a) Time to first compromised guard and exit.

(b) Time to first compromised guard.

(c) Time to first compromised exit.



(d) Fraction of streams with compromised guard and exit.

(e) Fraction of streams with compromised guard.

(f) Fraction of streams with compromised exit.

Figure 2: Empirical distribution of security metrics, 10/2012 – 3/2013, 83.3 MiB/s malicious guard and 16.7 MiB/s malicious exit.

This rate is roughly the compromise rate achieved by the adversary when chosen as the user's guards.

The differences in security between user models is due primarily to two factors: (*i*) the amount of user activity and (*ii*) the destination addresses and ports. Creating many streams increases the number of opportunities to choose malicious relays, and thus the speed at which that occurs, while connecting to destinations that are disallowed by many exits increases the chance that a selected exit relay will be malicious.

As described in Section 5, the BitTorrent model creates over 2.5 times as many streams as the Typical model and over 50 times as many as the IRC model. In addition, among the 171 different ports used are included several ports (6881, 6924, 6910, and 6966) that are rejected in the default Tor exit policy precisely because they are used by BitTorrent. Relatively few exits allow these ports, enabling the malicious exit to provide a larger fraction of that bandwidth. Thus we can see in Figure 2f that the BitTorrent model experiences exit compromise at a median time of less than 6 hours and median rate of over 12%, which is much quicker and more frequently than the Typical model. This translates to reduced security against full compromise as well. The IRC and WorstPort models see similarly bad security relative to the Typical user, as they both connect to ports that comparatively few exits support. Finally, we observe that the BestPort model has nearly identical compromise rates to the Typical model, which is not surprising as the Typical model only connects to port 80 in addition to 443, and nearly all exits that support 443 also support 80.

Finally, we consider the effect of changing how much bandwidth the adversary has and when he starts using it. Figure 3 shows the distribution of the time to full compromise of a Typical user as the adversary's bandwidth varies between 10MiB/s and 200MiB/s. Doubling the adversary's bandwidth roughly halves the time to first compromise, with the result that at 200MiB/s the adversary fully compromises a user within 30 days with probability 50%. On the other hand, an adversary that is limited to 10MiB/s (still more than a typical consumer-grade connection) has a less than 10% chance to compromise a user at all.

In addition, we show the time to full compromise when the adversary doesn't have a guard or exit relay until 12/1/2012, two

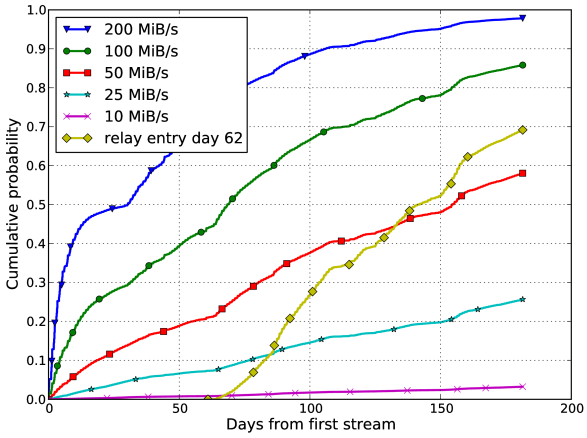


Figure 3: Time to first circuit with guard and exit compromised, varying total adversary bandwidth and date of malicious relay entry, 10/2012 – 3/2013

months into the simulation. At this point the user has already chosen guards after rotating them at least once. We can see that within the four remaining months of the simulation the adversary fully compromises the user with a probability of nearly 70%, which is nearly the probability of compromise after four months of running a relay from the outset.

6.2 Network Adversary

Unlike the relay adversary, a network adversary does not run relays in the hope that a client will choose one of those malicious relays at the guard and exit positions in its path. Instead, a network adversary leverages their position as a carrier of network traffic to correlate Tor traffic streams that cross their network *at some point* between the client and guard and exit and destination pairs.

We begin our discussion of how Tor clients are exposed to network adversaries by considering the placement of clients within the network and their behavior. We then consider the threat posed to those clients from three varieties of network adversaries: ASes, IXPs, and organizations which administer multiple IXPs.

6.2.1 Client Behavior and Location

We consider three types of clients in our analysis of a network-level adversary: Typical, BitTorrent, and IRC.

We do not consider the *WorstPort* and *BestPort* behavior patterns, as these are highly dependent upon exit policy diversity and do not directly affect a network adversary. Note, however, that an equivalently dangerous behavior pattern exists in the case of a network adversary: a client whose communication originates and terminates within the same autonomous system can be deanonymized by that autonomous system. All traffic will pass through the adversary on the path from client to guard and again on the path from exit to destination. We consider this an uninteresting case; for the remainder of this section, we *omit* ASes which contain clients, or destinations for a given client and activity, from the set of adversaries.

For each behavior, we use TorPS to conduct 50000 Monte Carlo simulations of three months of client activity spanning the period from January 2013 to March 2013. We use the output of these simulations to model multiple clients.

Client Location. TorPS simulated output paths are client agnostic; Tor currently makes no changes to path selection behavior based on client attributes (doing so could unintentionally decrease the safety of its users). However, since the path between client and guard is required to analyze exposure to network-level adversaries, we must place the clients somewhere within the network. We assign clients to the five most popular[3] client ASes (AS3320, AS3209, AS3269, AS13184, and AS6805) as identified by Edman and Syverson in 2009 [15], noting that similar techniques have been used recently by papers investigating Tor network security [44]. The five ASes include four from Germany and one from Italy. We then analyze the client-to-guard path five times for each sample stream from our Monte Carlo simulations, once for each of the client origins.

6.2.2 Network Adversaries

We consider three types of network adversaries: autonomous systems, Internet exchange points, and Internet exchange point organizations. A network connection often transits multiple ASes as it moves from source to destination; a network operator interested in deanonymizing Tor traffic need only have the traffic transit through its domain of control once on each side of the path.

IXPs represent points where ASes interconnect; traffic between multiple ASes may flow through a single IXP. In this position, IXPs may have significant ability to deanonymize Tor users.

As an extension of our IXP analysis, we also consider the situation in which a single organization may control multiple IXPs. Manually comparing IXP descriptions from the IXP Mapping Project and company websites for IXPs, we identify 19 IXP organizations which collectively administer 90 distinct IXPs. To identify the organizations which are able to compromise client streams we perform the same procedure as for individual IXPs, replacing IXP identifiers with organization identifiers where possible. We include IXPs for which we have no identified organization as standalone organizations.

6.2.3 Analysis

We begin our analysis by identifying a set of specific *adversarial entities* for each combination of client behavior and client origin. Previous work has often considered the ability of network adversaries to compromise Tor circuits independently, reporting that a large percentage of circuits can be deanonymized by *some* AS.

[3]We exclude Chinese ASes, since Tor has subsequently been blocked in China.

Adv. Type	ID	Description	Comp. %
AS	3356	Level 3 Communications	0.5%
AS	1299	TeliaNet Global	0.5%
AS	6939	Hurricane Electric	0.4%
IXP	286	DE-CIX Frankfurt	0.1%
IXP Org.	DE-CIX	DE-CIX	0.1%

Table 4: Identified Adversarial Entities for clients originating in AS3320 using BitTorrent. Comp. % gives the probability that that entity will compromise any given stream.

While this is a useful metric for system operators who are concerned with the security of Tor in the aggregate, it is not credible to consider the set of all independent ASes as potential adversaries from the user perspective. We identify distinct *adversarial entities* specific to each simulated user origin and behavior. By focusing our analysis on the ability of these entities to compromise user streams, we are able to produce security metrics which are more relevant to an end user of Tor.

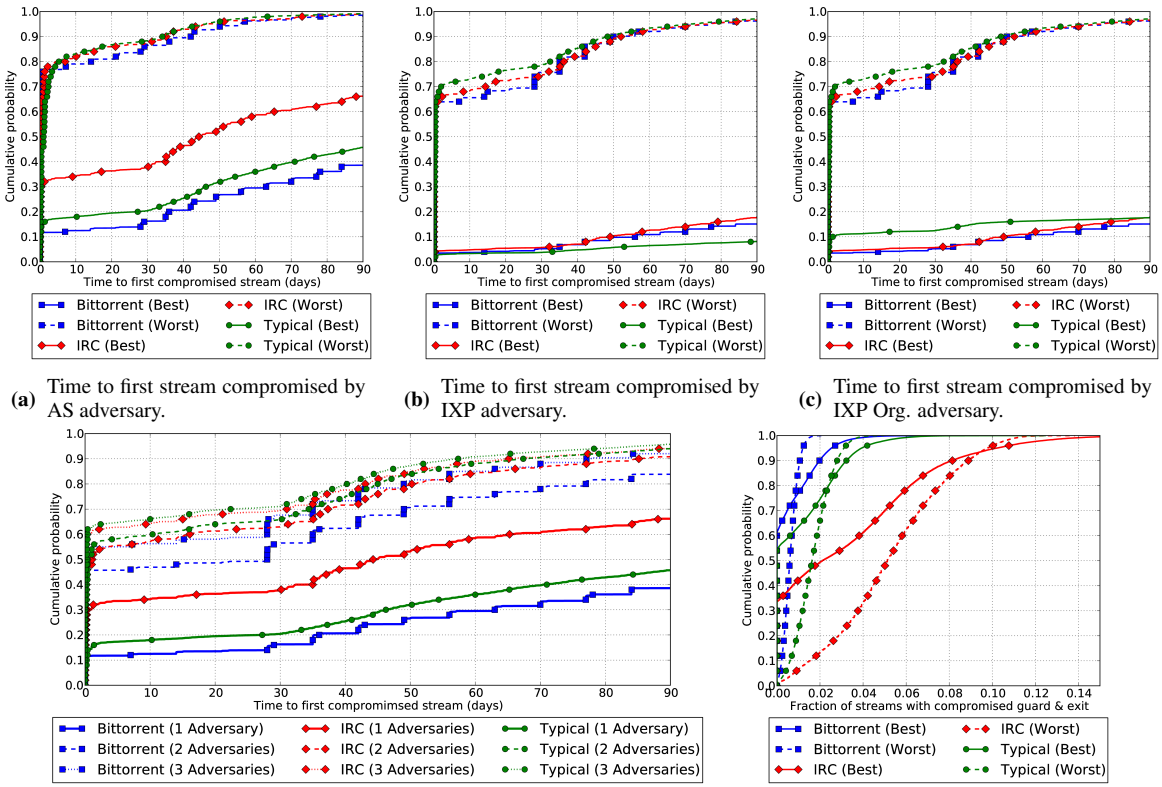
To identify candidate entities, we aggregate all streams over all client samples originating from a given client location. We then compute the client-side and destination-side paths, and count the number of streams in which a given adversarial entity (AS, IXP, or IXP organization) exists on both sides. We then select the entity which compromises the largest number of streams to understand the extent to which a strong adversary affects user security. Table 4 shows a sampling of the identified adversarial entities for BitTorrent users originating from AS3320.

Our simulation results show that there is significant variation in the ability of network adversaries to compromise Tor users depending on where the user is located, but that on the whole network adversaries present a significant potential threat. While we run experiments for all selected client origins (as described in Section 6.2.1), we display only the best and worst cases in our results for readability. We measure *best* and *worst* as the client origin with the smallest and largest area under the curve, respectively, in their CDF of time to compromise.

Against an AS-level adversary (Figure 4a), our results show compromise is highly likely in the worst case scenario regardless of user behavior. 45.9%, 64.9%, and 76.4% of Typical, IRC, and BitTorrent samples use a compromised stream within one day. At least one stream is compromised within the three month period for over 98% of samples. The best case client origins fare significantly better, but retain significant exposure to AS adversaries: IRC users are exposed within 44 days at the median. Although more than 50% of BitTorrent and Typical users evade compromise for the entire 90 day period, a significant proportion of them, 38% and 44% respectively, still use compromised streams.

IXPs (Figure 4b) and IXP organizations (Figure 4c) appear similar to the AS adversary in the worst case, but significantly less of a threat in the best case. Fewer than 20% of clients use a stream that could be compromised within three months. This difference is not terribly surprising: while IXPs represent high-degree connection points through the network, 80% of the network links do not traverse IXPs. Thus, the worst case likely indicates a situation in which the client's outbound path to a guard transits through an IXP while the best case traverses non-IXP links.

While IXPs and IXP organizations are generally similar, it is clear from the Typical user model that those concerned about the ability of IXPs to compromise Tor streams should consider organizations rather than individual IXP locations: in the best case standalone IXPs are able to compromise just 3.7% of samples within 30 days, while organizations compromise 12.4% in the same period.



(a) Time to first stream compromised by AS adversary.

(b) Time to first stream compromised by IXP adversary.

(c) Time to first stream compromised by IXP Org. adversary.

(d) Varying time to first compromise as an AS adversary controls more entities.

(e) Fraction of streams compromised by AS adversary.

Figure 4: Network adversary analysis. "best" and "worst" indicate the client origin from the top five ASes from [15] with the smallest and largest area under the curve, respectively. "*N* Adversaries" indicates an adversary that controls the top *N* AS entities.

We additionally consider how adversary strength affects the likelihood of stream compromise. We consider adversaries of varying strength by adjusting the number of *entities* they control from one to three. Thus, our weakest adversary controls the top *adversarial entity* of that type and the strongest controls the top three *adversarial entities*. Figure 4d shows how the time to first compromised stream drops as an adversary controls more of the top adversarial AS entities for each behavior model. Here we consider *only the best case* since just one AS entity is already able to compromise a significant fraction of samples in the worst case. The addition of even one more AS entity causes the number of samples compromised within 30 days to jump 156%, 65.8%, and 122% for BitTorrent, IRC, and Typical users. The amount of "ground" that two ASes can cover is significantly higher than the amount that one can cover.

In addition to the speed of compromise, we are equally interested in the probability that the adversary compromises any given stream. Figure 4e shows the fraction of streams that an AS adversary controlling the top entity compromises given a particular user activity. The probability of compromising each stream is quite low even in the worst case: 0.6%, 5.1% and 1.6% at the median for BitTorrent, IRC and Typical users respectively. As with the relay adversary, however, the compromises happen in higher-rate bursts during the period in which traffic to a guard is observable.

Discussion. At a high level, the network adversary analysis shows that – in contrast to the relay model – client behavior which results in low diversity of client destinations is most likely to result in a compromise. For example, a single AS adversary can compromise 50% of clients using IRC within 44 days, even under the most optimistic client placement. By contrast, fewer than half of Typical clients are compromised within the entire period, and BitTorrent users have even lower compromise rates. This effect is attributable to simple probability: as the set of destinations which the adversary must cover narrows it becomes more likely that the user picked at the same time a destination and guard that the adversary is near. An implication is that to the extent that Tor clients seek to evade a network adversary that can optimally position himself, they should go to a diverse set of destinations and should use as many different paths as possible.

It is also notable that a large number of clients encounter compromised streams very quickly followed by a steep decline in the rate. Figure 4e shows that the overall per-stream compromise rate is relatively low, so this phenomenon is somewhat surprising. Just as in the case of a relay adversary, this result is partly attributable to how guard selection interacts with relay selection. Given the initial set of guards, the path between the client and entry guard is relatively fixed. If the adversarial entity exists on the guard side of the path, then it need only wait until it appears on the exit side. However, if the entity does not exist on any of the paths from a client to it chosen guards, it will not compromise any streams until new guards are selected. This results in a relative plateau after initial compromises (disturbed only by minor guard as churn nodes leave the network or hibernate) until 30 days have passed and new guards begin to be selected.

Finally, while IXPs have a distinctly lower likelihood of compromising client traffic, it should be noted that the complexity of performing traffic correlation at an IXP is likely to be significantly lower than at an AS. ASes may span large regions and traffic may not pass through the same routers on the forward and return path, while IXPs by their very nature are geographically concentrated.

This may make it easier for a single rogue agent at an IXP to perform traffic analysis than it is to organize a concerted AS-wide effort. Tor users evaluating the ability of network adversaries to compromise their communications should consider this factor; IXPs may represent a lower overall threat profile, but have fewer obstacles to effecting a coordinated traffic analysis attack.

7. ALTERNATIVE PATH SELECTION

This paper has thus far focused on Tor's path selection protocol and has outlined severe security implications. Unsurprisingly, path selection algorithms also have a large impact on Tor's performance because they directly affect how client load is balanced among the available relay resources and therefore how congested relays become. Researchers have investigated and proposed several improvements to Tor's current path selection algorithm. This section explores the security implications of the most effective of these proposals to both inform the adoption of these changes by The Tor Project and to show how future work can apply our methods to provide accurate security assessments of new proposals.

7.1 Congestion Awareness

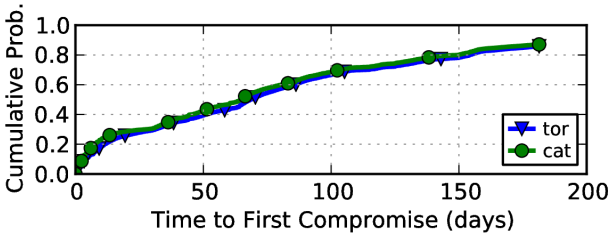
Based on the results of Wacek *et al.* [44], the most effective proposed improvements to path selection is the "instant response" mode of Congestion-Aware Tor (CAT) by Wang *et al.* [45]. The main idea in CAT is that clients create a local view of circuit congestion through opportunistic and active measurements of circuit round trip times (RTTs). Circuits that are or become too congested are ignored or dropped, respectively. To accomplish this, CAT collects five RTT measurements at circuit construction time before the circuit is used. When a pre-built circuit is needed, the client chooses the circuit with the lowest average circuit congestion, where circuit congestion for each of the RTT measurements is computed by subtracting the minimum RTT ever measured on that circuit. While using the circuit, the client continues to opportunistically measure circuit RTTs using existing Tor protocol cells. If the mean of the last 5 congestion measurements is greater than 0.5 seconds, the client stops using that circuit for new streams.[4]

7.2 Methodology

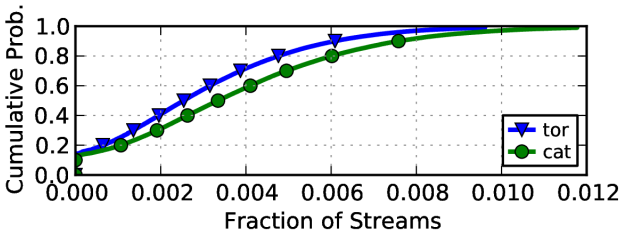
As CAT only slightly modifies Tor's original path selection algorithm, our methodology for evaluating CAT's security is largely the same as described in Section 5. However, we made several changes to TorPS to incorporate the new selection algorithm. In addition to implementing the instant response mode of CAT, we also needed a source for relay congestion over our analysis period. Although TorPS uses historical data collected from The Tor Project in order to build paths, data about relay congestion has not been collected historically.

Without available data about relay congestion in the live Tor network, we created relay-specific congestion models as follows. We created a virtual Tor network using Shadow [25] following a standard Tor network modeling approach [26]. As Shadow runs the real Tor software, we maximize fidelity to Tor protocols. We then implemented the CAT extensions in Tor[5] and instrumented Shadow to collect fine-grained timing information for packets as they travel between the virtual nodes and through the circuit. This timing information allows us to precisely isolate congestion due to network, kernel, or application latency, giving us a unique view of congestion at each relay over time.

[4]Although unspecified by Wang et al. [45], we assume the circuit is marked dirty and destroyed only when exiting streams are finished.
[5]Our CAT implementation is based on Tor stable version 0.2.3.25.



(a) Time to first compromised guard and exit.



(b) Fraction of streams with compromised guard and exit.

Figure 5: Empirical distribution of security metrics, 10/2012 – 3/2013, 83.3 MiB/s malicious guard and 16.7 MiB/s malicious exit, Tor vs Congestion-aware Tor (CAT) path selection, "typical" user model.

Running thousands of Tor nodes simultaneously is time consuming, and therefore it is not feasible to gather congestion data for the 6-month analysis period that we require. Instead, we ran several short identical pairs of experiments with different seeds, and use the Kolmogorov-Smirnov (K-S) test statistic as a distance metric between each resulting pair of relay-specific congestion traces. We lengthen and repeat the experiment to increase our sample size until the median K-S distance of all relay trace pairs is below five percent, increasing our confidence in the consistency of congestion produced in our virtual network.

We create congestion *profiles* by smoothing each relay congestion trace by binning the values. We assign each simulated relay the profile with the closest consensus bandwidth weight to its own. When running the CAT path simulation, each relay's profile is queried whenever a congestion value would have been measured for that relay.

7.3 Analysis

Figure 5 shows the results of our CAT simulations with the relay adversary compared to Tor, under the typical user model. Figure 5a indicates that CAT reduces the time to first compromise, but that the difference is quite minor. This is expected: while congestion awareness affects which circuits get used, it does not affect selection of entry guards (recall that guards provide the largest influence on time to first compromise). However, congestion awareness does affect which circuits get used over time, which in turn directly affects the total fraction of streams that get compromised. Figure 5b indeed shows a more drastic increase in the total amount of streams that are compromised. This is likely caused by less congested adversarial relays biasing the client's circuit choices to those the adversary compromised. This behavior presents a new vector for attack: an adversary can bias the circuits that get used by the client to those it has compromised by increasing circuit response time for those circuits of which it is a member but has not compromised. This is an active attack similar to selective denial of service (but much harder to detect), and therefore it falls outside the scope of our adversary model. Note that we came to similar conclusions with the results from the BitTorrent and IRC user models.

8. CONCLUSION

We present in this paper a realistic and comprehensive analysis of the security of Tor against traffic correlation. Our approach carefully defines adversaries and uses them to define security metrics that capture user security over time. We propose adversaries that control one or more fixed ASes or IXPs. We present new, practical security metrics that show for the first time how long a user can stay anonymous and how often an adversary can deanonymize. We developed several tools and techniques to allow us to evaluate our security metrics on the live Tor network. These include models of user activity online, an up-to-date and comprehensive Internet map with BGP routes, and a model of relay congestion based on full-network simulations with Shadow.

The results show that Tor faces even greater risks from traffic correlation than previous studies suggested. An adversary that provides no more bandwidth than some volunteers do today can deanonymize any given user within three months of regular Tor use with over 50% probability and within six months with over 80% probability. We observe that use of BitTorrent is particularly unsafe, and we show that long-lived ports bear a large security cost for their performance needs. We also observe that the Congestion-Aware Tor proposal exacerbates these vulnerabilities.

Some of our results against an adversary controlling ASs or IXPs are similarly alarming. Some users experience over 95% chance of compromise within three months against a single AS or IXP. We see that users' security varies significantly with their location. However, an adversary with additional ASes or IXPs has much higher compromise speed, notably against even those users in "safer" locations. Such an adversary is highly relevant in today's setting in which many large organizations control multiple ASes or IXPs. Surprisingly, we observe that high diversity in destinations may actually result in improved security against a network adversary.

These results are somewhat gloomy for the current security of the Tor network. However, they do suggest several ways in which security could be significantly improved. The results against the relay adversary show that choosing multiple guards amplifies the probability that the adversary's guard is chosen. Reducing the default number of guards by some factor would immediately cut compromise rates by the same factor. Those same results show that guard expiration, which starts after 30 days, noticeably speeds up the time to first compromise. Increasing the time to expiration would significantly increase the time to compromise (in fact, the minimum guard expiration time was increased to 60 days in Tor version 0.2.4.12-alpha for exactly this reason). Elahi et al. [16] report some results on how making such changes in guard selection improves security. It seems more difficult to improve security against the network adversary, but several proposals have been given [15, 19, 28]. We suggest evaluating these defenses using the methodology we have presented as well as designing new solutions with our adversary models and security metrics in mind.

Our results do suggest that current users of Tor should carefully consider if it meets their security needs. In particular, users facing persistant adversaries who might run relays or monitor network traffic should be aware of the threat of traffic correlation. While improved defenses are still being developed, such users may be able to take defensive measures on their own. For example, they can choose to limit which relays their client will select using manual configuration options (EntryNodes, ExitNodes, ExcludeNodes, etc.). While this does break the uniformity of path selection among clients, that may be a worthwhile risk tradeoff for these users. Johnson et al. [27] suggest an approach along these lines that balances choosing relays using per-client trust with blending in with other clients.

A goal of our analysis is that it inform safer use of Tor and inspire more secure designs. Despite our pessimistic results, Tor has provided real and valuable privacy to thousands of users. We are optimistic that it can continue and improve this service.

Acknowledgments

We thank Leiah Stevermer for assistance with the user models. Work by Jansen, Johnson, and Syverson supported by ONR and DARPA. For work by Sherr and Wacek: This material is based upon work supported by the Defense Advanced Research Project Agency (DARPA) and Space and Naval Warfare Systems Center Pacific under Contract No. N66001-11-C-4020. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research Project Agency and Space and Naval Warfare Systems Center Pacific. This work is partially supported by NSF CAREER CNS-1149832 and NSF grants CNS-1064986, CNS-1204347, and CNS-1223825.

References

[1] 0x539 Dev Group. Gobby: A Collaborative Text Editor. http://gobby.0x539.de, 2013.

[2] T. G. Abbott, K. J. Lai, M. R. Lieberman, and E. C. Price. Browser-Based Attacks on Tor. In *Privacy Enhancing Technologies Symposium (PETS)*, 2007.

[3] M. Akhoondi, C. Yu, and H. V. Madhyastha. LASTor: A Low-Latency AS-Aware Tor Client. In *IEEE Symposium on Security and Privacy (Oakland)*, 2012.

[4] B. Augustin, B. Krishnamurthy, and W. Willinger. IXPs: Mapped? In *ACM SIGCOMM Conference on Internet Measurement (IMC)*, November 2009.

[5] S. L. Blond, P. Manils, A. Chaabane, M. A. Kaafar, A. Legout, C. Castellucia, and W. Dabbous. De-anonymizing BitTorrent Users on Tor (poster). In *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2010.

[6] N. Borisov, G. Danezis, P. Mittal, and P. Tabriz. Denial of Service or Denial of Security? How Attacks on Reliability can Compromise Anonymity. In *ACM Conference on Computer and Communications Security (CCS)*, 2007.

[7] X. Cai, J. Heidemann, B. Krishnamurthy, and W. Willinger. Towards an AS-to-organization Map. In *Internet Measurement Conference*, 2010.

[8] X. Cai, X. C. Zhang, B. Joshi, and R. Johnson. Touching from a Distance: Website Fingerprinting Attacks and Defenses. In *ACM Conference on Computer and Communications Security (CCS)*, 2012.

[9] CAIDA. IPv4 Routed /24 Topology Dataset. http://www.caida.org/data/active/ipv4_routed_24_topology_dataset.xml, December 2012.

[10] CAIDA. The CAIDA AS Relationships Dataset. http://www.caida.org/data/active/as-relationships/, June 2012.

[11] D. L. Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[12] C. Díaz, S. Seys, J. Claessens, and B. Preneel. Towards Measuring Anonymity. In *Privacy Enhancing Technologies (PET)*, 2003.

[13] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The Second-Generation Onion Router. In *USENIX Security Symposium (USENIX)*, 2004.

[14] P. Eckersley. How Unique is Your Browser? In *Privacy Enhancing Technologies Symposium (PETS)*, 2010.

[15] M. Edman and P. Syverson. AS-Awareness in Tor Path Selection. In *ACM Conference on Computer and Communications Security (CCS)*, 2009.

[16] T. Elahi, K. Bauer, M. AlSabah, R. Dingledine, and I. Goldberg. Changing of the Guards: A Framework for Understanding and Improving Entry Guard Selection in Tor. In *ACM Workshop on Privacy in the Electronic Society (WPES)*, 2012.

[17] Equinix. Equinix Internet Exchange Enables Efficient Interconnection between Hundreds of Networks. http://www.equinix.com/solutions/by-services/interconnection/exchanges/equinix-internet-exchange/.

[18] N. S. Evans, R. Dingledine, and C. Grothoff. A Practical Congestion Attack on Tor using Long Paths. In *USENIX Security Symposium (USENIX)*, 2009.

[19] N. Feamster and R. Dingledine. Location Diversity in Anonymity Networks. In *ACM Workshop on Privacy in the Electronic Society (WPES)*, 2004.

[20] J. Feigenbaum, A. Johnson, and P. Syverson. Probabilistic Analysis of Onion Routing in a Black-box Model. *ACM Transactions on Information and System Security (TISSEC)*, 15(3):14:1–14:28, 2012.

[21] L. Gao. On Inferring Autonomous System Relationships in the Internet. In *IEEE/ACM Transactions on Networking*, volume 9, pages 733–745, December 2001.

[22] S. Hahn and K. Loesing. Privacy-preserving Ways to Estimate the Number of Tor Users, November 2010. Available at https://metrics.torproject.org/papers/countingusers-2010-11-30.pdf.

[23] A. Hamel, J.-C. Grégoire, and I. Goldberg. The Misentropists: New Approaches to Measures in Tor. Technical Report 2011-18, Cheriton School of Computer Science, University of Waterloo, 2011.

[24] N. Hopper, E. Y. Vasserman, and E. Chan-Tin. How Much Anonymity Does Network Latency Leak? *ACM Transactions on Information and System Security (TISSEC)*, 13(2): 13, 2010.

[25] R. Jansen and N. Hopper. Shadow: Running Tor in a Box for Accurate and Efficient Experimentation. In *Network and Distributed System Security Symposium (NDSS)*, 2012.

[26] R. Jansen, K. Bauer, N. Hopper, and R. Dingledine. Methodically modeling the tor network. In *USENIX Workshop on Cyber Security Experimentation and Test (CSET)*, August 2012.

[27] A. Johnson, P. Syverson, R. Dingledine, and N. Mathewson. Trust-based anonymous communication: Adversary models and routing algorithms. In *Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS 2011)*, pages 175–186. ACM, 2011.

[28] J. P. J. Juen. Protecting Anonymity in the Presence of Autonomous System and Internet Exchange Level Adversaries. Master's thesis, University of Illinois, 2012.

[29] S. J. Murdoch and G. Danezis. Low-Cost Traffic Analysis of Tor. In *IEEE Symposium on Security and Privacy (Oakland)*, 2005.

[30] S. J. Murdoch and P. Zieliński. Sampled Traffic Analysis by Internet-Exchange-Level Adversaries. In *Privacy Enhancing Technologies (PET)*, 2007.

[31] Office of Engineering and Technology and Consumer and Governmental Affairs Bureau. A Report on Consumer Wireline Broadband Performance in the U.S. Technical report, Federal Communications Commission, February 2013.

[32] L. Øverlier and P. Syverson. Locating Hidden Servers. In *IEEE Symposium on Security and Privacy (Oakland)*, 2006.

[33] J. Qiu and L. Gao. AS Path Inference by Exploiting Known AS Paths. In *Global Telecommunications Conference*, 2006.

[34] A. Serjantov and G. Danezis. Towards an Information Theoretic Metric for Anonymity. In *Privacy Enhancing Technologies (PET)*, 2003.

[35] M. Sherr, M. Blaze, and B. T. Loo. Scalable Link-Based Relay Selection for Anonymous Routing. In *Privacy Enhancing Technologies Symposium (PETS)*, August 2009.

[36] R. Smits, D. Jain, S. Pidcock, I. Goldberg, and U. Hengartner. BridgeSPA: Improving Tor Bridges with Single Packet Authorization. In *ACM Workshop on Privacy in the Electronic Society (WPES)*, 2011.

[37] P. Syverson. Why I'm not an Entropist. In *International Workshop on Security Protocols*, 2009.

[38] P. Syverson, G. Tsudik, M. Reed, and C. Landwehr. Towards an Analysis of Onion Routing Security. In *Designing Privacy Enhancing Technologies*, 2000.

[39] The Tor Project. Changelog Tor 0.2.4.12-alpha. https://gitweb.torproject.org/tor.git?a=blob_plain;hb=HEAD;f=ChangeLog.

[40] Tor Project, Inc. Tor Metrics Portal. https://metrics.torproject.org/, 2013.

[41] Tor Project, Inc. The Tor Project. https://www.torproject.org/, 2013.

[42] TorPS. TorPS: The Tor Path Simulator. http://torps.github.io, 2013.

[43] University of Oregon. RouteViews Project. http://www.routeviews.org/, 2013.

[44] C. Wacek, H. Tan, K. Bauer, and M. Sherr. An Empirical Evaluation of Relay Selection in Tor. In *Network and Distributed System Security Symposium (NDSS)*, 2013.

[45] T. Wang, K. Bauer, C. Forero, and I. Goldberg. Congestion-aware Path Selection for Tor. In *Financial Cryptography and Data Security (FC)*, 2012.

[46] L. Wasserman. *All of Nonparametric Statistics (Springer Texts in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[47] M. Wright, M. Adler, B. N. Levine, and C. Shields. The Predecessor Attack: An Analysis of a Threat to Anonymous Communications Systems. *ACM Transactions on Information and System Security (TISSEC)*, 4(7):489–522, November 2004.