# Onions in the Crosshairs

When The Man really *is* out to get you

Aaron D. Jaggard
U.S. Naval Research Laboratory
aaron.jaggard@nrl.navy.mil

Paul Syverson
U.S. Naval Research Laboratory
paul.syverson@nrl.navy.mil

## ABSTRACT

We introduce and investigate *targeting adversaries* who selectively attack users of Tor or other secure-communication networks. We argue that attacks by such adversaries are more realistic and more significant threats to those most relying on Tor's protection than are attacks in prior analyses of Tor security. Previous research and Tor design decisions have focused on protecting against adversaries who are equally interested in any user of the network. Our adversaries selectively target users—e.g., those who visit a particular website or chat on a particular private channel—and essentially disregard Tor users other than these. We investigate three example cases where particular users might be targeted: a cabal conducting meetings using MTor, a published Tor multicast protocol; a cabal meeting on a private IRC channel; and users visiting a particular .onion website. In general for our adversaries, compromise is much faster and provides more feedback and possibilities for adaptation than do attacks examined in prior work. We also discuss selection of websites for targeting of their users based on the distribution across users of site activity. We describe adversaries attempting to learn the size of either a cabal meeting online or a set of sufficiently active visitors to a targeted site, and we describe adversaries attempting to identify guards of each targeted user. We compare the threat of targeting adversaries versus previously considered adversaries, and we briefly sketch possible countermeasures for resisting targeting adversaries.

## KEYWORDS

adversary models; Tor; targeted attacks; website fingerprinting

## 1 INTRODUCTION

Tor is a network for traffic security of Internet communications [6] with millions of users [32]. Most Tor users are unlikely to be of specific interest to an adversary; they are primarily protected by Tor against opportunistic local eavesdroppers and local censors or against hostile destinations. Deanonymizing adversaries are generally modeled as attempting to attack as many users as possible rather than targeting particular users or groups.

For many Tor users this may be appropriate, but Tor is explicitly intended to protect human rights workers, law enforcement officers, military servicemembers, journalists, and others [31] who may face large, well-financed, and determined adversaries. More to the point, some of these adversaries will hoover up whatever they can, but they may also be more interested in specific individuals or groups of Tor users, possibly based on offline or out-of-band reasons, or on these combined with results of hoovering if they hoover. An adversary whose interest is directed primarily or more fervently at particular users may employ different strategies. And if Tor's design decisions are motivated by analyses of what hoovering adversaries can do, those most in need of Tor's protections may be the least well served.

We regard the adversaries in this paper as the next step in an ongoing evolution of most appropriate and important onion-routing adversaries, away from abstracting reality till it matches models and towards better matching models to reality. Our focus in this work is on *targeting adversaries*. These need not differ at all from previously studied adversaries in terms of their capabilities or resource endowment, though they might. They differ primarily in their goals and strategies. We will set out various types of targeting adversaries presently; however, we mention an example here to give the basic idea. A targeting adversary, Tom, who has compromised a particular user of interest, Alice, and observed her connecting to Bob, an interesting or unusual .onion website (essentially, a website reachable only over Tor) may wish to target other users of that site. Tom might be particularly interested to learn which are the most active site users or how popular the site is in general.

Most research on security for widely used systems follows the paradigm of assuming hoovering adversaries. Nonetheless, targeting has been shown to sometimes be much more effective than hoovering. For example, password guessing that is targeted based on knowledge about the intended victim has been shown to be more effective than hoovering in analyses of real data based on leaked datasets of passwords, and typically much more than twice as effective for passwords of security-savvy users [35]. And, NIST authentication guidelines, which had been created in consideration of hoovering strategies, were quickly modified in light of these analyses.

**Background:** We sketch here a basic background on Tor to provide context for this work. For further descriptions, see the Tor design paper [6], or related documentation at the Tor website [34]. Tor clients randomly select sequences of three out of roughly 10,000 relays [33] forming the current Tor network, and create a cryptographic circuit through these to connect to Internet services. Since only the first relay in the circuit sees the IP address of the client and only the last (exit) relay sees the IP address of the destination, this technique separates identification from routing. In response to a vulnerability analysis [25], Tor began having clients each choose a small set of *entry guards* from which to persistently choose the first relay in any circuit. The appropriate size and rotation criteria for the set of guards is the focus of ongoing work. For purposes of this paper, we assume the default for official distributions of Tor


*WPES'17, October 30, 2017, Dallas, TX, USA*
2017. ACM ISBN 978-1-4503-5175-1/17/10…$15.00
https://doi.org/10.1145/3139550.3139553

software: a single long-persistent guard per client. Except where explicitly addressed, we also ignore any guard dynamics. An extended version of this paper considers the default Tor used for the decade following the introduction of guards, in which the basic client guard set had three members [13]. Tor currently has roughly two million users connecting in this way [32]. For simplicity of this first analysis of targeting adversaries, we ignore the 150,000 users currently connecting to Tor via bridges, a mechanism to access Tor in environments that censor users connecting to the Tor network at all. Nonetheless, much of our analysis will apply there as well.

Tor also facilitates onion services, which are services at Internet sites ("onionsites") on the .onion top-level domain that is reserved by IETF standard [2]. Onionsites create circuits into the Tor network to *Introduction Points* at which they are then reachable. Thus, though not part of the Tor network itself, onionsites are only reachable via Tor. Connections to them, including for address lookup, do not exit the Tor network described above.

**Structure and Result Highlights:** After noting some relevant prior work next, we present in Sec. 2 a brief description of the targeting adversaries we use in our worked examples, along with the general strategy they all follow. A more general and abstract description of various types of targeting adversaries, their goals, and their properties is given in the extended version of this paper.

Our main results are in the following contexts:

- A study of MTor, a published protocol for multicast over Tor [21]. Its security goals are to "prevent an adversary from (1) discerning the sender of the message and (2) enumerating group members." We show in Sec. 3 that a targeting adversary with capabilities within those assumed by MTor's authors can enumerate group members and can identify the guard relay of a message sender.
- A study of Internet Relay Chat (IRC). In Sec. 4, we describe how a targeting adversary will, within a few weeks of attack initiation, have a high probability of locating the leader of a cabal that meets several times per day on a private IRC channel. In contrast, to achieve the same expectation of cabal-leader location by an adversary with roughly the same resources using previous strategies would require several months [18]. Some of our analysis from Sec. 3 applies to this scenario as well.
- A study of onionsites in Tor. We show in Sec. 5 that a moderately resourced adversary can assess not just the level of site activity but the distribution of client interaction with a targeted onionsite and will identify the guards of more active clients, potentially for additional targeted attacks. This uses attacks similar to those that we describe against MTor and IRC cabals. The importance of this type of attack is illustrated by the fact that Tor has recently taken steps to make it difficult for adversaries to predict the onionsites for which relays they own would function as directory [8] or recognize which onionsite is being requested when receiving a directory request [22]. This was in part to counter published attacks allowing an adversary to monitor interest in onionsites by monitoring the rate of directory requests [3].
- An analysis of adversary adaptation and defenses against targeting adversaries. In Sec. 6, we show that targeting adversaries receive feedback on intermediate goals such as cabal size and activity, allowing them to decide, adapt, or refocus subsequent attacks at a similarly faster rate than previous adversaries. We also briefly describe possible counters to the kinds of targeted attacks we introduce in this paper.

**Related Work:** We will primarily discuss related work at points in the text where that work is relevant. We here note a few highlights of prior general work on Tor (or, more generally, onion routing) adversary models and security analysis.

Analysis of onion routing security has generally focused on end-to-end correlation. To be practical, onion-routing networks are generally low latency. Thus, an adversary able to observe both ends of a connection can correlate patterns of communication and correlate connection source and destination with little error, regardless of what happens between the ends. Given the fraction $f$ of Tor relays that are compromised or observed, this provides roughly $f^2$ probability of any one onion-routing circuit being compromised [30]. Various end-to-end correlating adversaries and this metric of security against them are the basis for the bulk of Tor security research and design. Hintz, however, was the first to observe that if an adversary can recognize a destination from the usual pattern of traffic that results from connecting to it, then it is sufficient to observe the client end to recognize its destination in a *fingerprinting* attack [10]. Such recognition is a central feature of attacks for all three of our examples.

Feamster and Dingledine were the first to examine an adversary occupying the network links between relays rather than at the relays themselves [7]. Vulnerability to link adversaries is significant enough that any useful Tor security design must take them into account. Nonetheless, we will show that a targeting relay adversary is sufficient to carry out effective attacks.

Prior to the last half decade, research has primarily looked at the risk of correlation at a network snapshot. Johnson et al. considered security of using the Tor network over time, examining such questions as the time until a user with a given type of behavior is likely to experience its first correlated connection, and given such behavior, the fraction of connections that will be so compromised over a period of use [18]. Since they consider IRC use as one of their classes of behavior, we will compare the attacks we devise on an IRC cabal to those they examined.

Not all work prior to that of Johnson et al. ignored observation over time. Predecessor and intersection attacks examine repeated connections or message transmissions to see who is the only one or the most common one who could have been sending when a given destination is receiving. Crowds was a published system design for anonymous web browsing that was created with these attacks in mind [28]. Wright et al. analyzed these attacks for many traffic security systems including pre-Tor onion routing [38]. Most intersection attacks and analyses looked for *any* association of a sender and receiver. As such they were not targeted. However, the first such attacks conducted on a deployed, publicly-used system were used to show that an adversary could repeatedly connect to and thereby find the IP address of a specific hidden onion service. These were a basis for introducing guard relays to Tor [25]. Nonetheless, end-to-end correlation still remains the primary concern of most Tor analyses and defenses.

## 2 TARGETING ADVERSARIES

We expect selective targeting Tor adversary description and analysis to be amenable to rigorous formal reasoning. We also anticipate future analyses of other targeting adversaries than in our examples, e.g., an adversary that attempts for a given targeted user to build a profile of that user's selected destinations and activity at them over a given period. To this end, the extended version [13] of this work sets out an abstract model of both system elements and actions, as well as different categories of targeting adversaries and their goals. Here we simply sketch the basic adversary properties and strategy that should apply to all of our worked examples. In the first two of these examples, the adversary is interested in a cabal of users communicating through Tor with either a multicast system or IRC; in our third main example, he is interested not in a cabal per se, but in the set of users who frequently visit a targeted onionsite.

The general approach that all of our examples follow is to have an adversary that initially deploys all of its relay capacity at middle relays in the network. We assume that communication within a targeted cabal or with a targeted onionsite is recognizable by its traffic patterns at middle relays. The basis of that assumption varies with example and is stated in each section. The initial strategy of the adversary is then to attempt to find a guard for each of the targeted clients by observing which guards transmit or receive recognizable target traffic. This may be a final strategy if the adversary's only goal is to learn the size of a cabal and/or monitor the distribution of its activity, e.g., if the adversary is only tasked with tracking network indicators of cabal significance. But, it may be just a stepping stone, e.g., to inform the decision whether to attempt to "bridge" guards—i.e., to transition beyond knowing that a guard is being used by one or more clients of interest to identifying client IP addresses. The adversary may be selective in this decision as well; rather than targeting all cabal members it might, e.g., attempt to bridge only for those that send the most or prompt the most responses when they send. Note that "bridging" typically implies getting across a network obstacle, rather than compromising it directly [4]. In our setting this could be done via requests to or extortion of the guard ISP, compromised ASes between a guard and client, etc. For convenience, we will subsume under "bridging" both bridging in that sense and compromise of a guard itself by physical access to its hardware, threatening its operator, exploiting configuration errors, using zero-day exploits on its software, etc.

Another adversary goal is to assign confidence to its assessment of cabal size. This can involve evaluation of confidence in the correctness of the fraction of cabal users who have had a guard identified. The adversary will also need to evaluate the expected degree of guard collision amongst cabal members.

## 3 EXAMPLE: MULTICAST CABALS

MTor is a design for multicast group communication over Tor recently introduced by Lin et al. [21]. Unlike both of our other examples, MTor is not currently available on the deployed primary Tor network. Nearly all of our analysis in this section, however, applies directly to the IRC cabal example of the next section. Each client that joins a given MTor multicast group creates a circuit to a Tor relay that serves as the group's current multicast root (MR). We do not describe MTor's selection or rotation of MR and skip many other details as well. Communication for the group travels up this circuit and then propagates down from any node that has untraversed subtrees to the group members at the leaves.

MTor also modifies normal Tor behavior for the potential performance gain from message deduplication that is typical of multicast. Tor normally creates cryptographic circuits by tunneling a Diffie–Hellman protocol to establish a session key known only to the client and to the next onion router in the circuit being built. MTor uses group keys: if a client attempts to build a circuit through a relay that is already part of the same multicast tree, the relay will recognize the session group identifier (GID) sent by the client and join the new circuit to the existing group circuit rather than continue building as per the client request. To further manage tree size and improve the advantages of multicast, MTor allows the restriction of middle relay selection for MTor communication to a designated subset of relays and/or to relays having a minimum bandwidth.

### 3.1 MTor adversary

A targeted and compromised Alice belonging to a cabal that meets only via MTor reveals to the adversary all cabal communications, as well as long-term group keys and identifiers. A targeted but uncompromised Alice with a compromised guard connecting to an MTor cabal could make the cabal a target by association. (And if the targeted group is open, the adversary can simply join it too.)

*3.1.1 Adversary goals.* Lin et al. consider adversary goals of looking at all pairs of users and trying to link each pair as part of a multicast group (by their guards seeing the same GID) and of identifying a user as participating in a multicast group (by a guard seeing the GID—experiments consider only a single multicast group) [21]. While these may be useful for some purposes, our targeting adversary has goals of identifying all members of a multicast group of interest, estimating the cabal's size, or identifying MTor groups to which a targeted user might belong. A targeted user communicating over MTor is a natural subject of all the adversary goals identified in App. B.3 to the extended version of this paper [13].

*3.1.2 Adversary endowment and capabilities.* For simplicity, and like Lin et al., we will consider only a relay adversary. On the other hand, it will be useful for our adversary to compromise relays other than guards. An adversary that owns middle relays can both estimate the cabal size and identify guards to target for compromise or bridging so as to identify the clients behind them. Even the MR can estimate cabal size.

The guard of an MTor group member can see all session GIDs for the group, and may then wish to identify, e.g., others in that group. We will assume, however, that traffic patterns for any cabal member in a multicast session will be adequately linkable so that the GID will not be needed for this adversary to associate other clients with the cabal. (This assumption is also made by MTor's authors.) In general, we consider an adversary capable of active attacks, including disrupting group communications or generating its own group traffic if a member. For simplicity, however, our initial analysis assumes a passive adversary. Since MTor sessions are always identifiable by a participating adversary relay and our analysis will parametrize over the number of sessions, this is not as significant a limitation on the adversary as is usually the case for Tor communications.

## 3.2 MTor cabal analysis

Note that while the GID is not that significant to a targeting adversary who can observe traffic patterns, the multicast tree structure is. To illustrate with an unrealistic example, if the middle-relay set were restricted to a singleton, then in sessions where the adversary has compromised this relay he has thereby identified all the guards used by any cabal members in that session. Lin et al. do not give criteria for middle-relay-set restriction. If gameable, an adversary might be able to improve its expected inclusion in this set disproportionate to its actual relay and bandwidth resources. On the other hand, a restriction of middle-relay-set size can obscure cabal size estimates by an adversarial MR.

*3.2.1 Learning a guard of every cabal member.* For our initial analysis, we consider an adversary who controls a fraction $B$ of the middle-relay bandwidth and seeks to identify a guard of each member of a cabal. (This might be used to adaptively target guards for future attack or as part of an estimation of cabal size. If the adversary has joined or compromised a cabal member, he can associate a guard with every cabal member who ever sends or receives through an adversary-owned middle relay.) We also consider the effects of the number $c$ of cabal members and the number $m$ of cabal multicast sessions observed. If the instance of MTor restricts the set of usable middle relays to obtain the associated deduplication benefit, we take $B$ to be the fraction of MTor-available middle-relay bandwidth that is controlled by the adversary. Here, we also consider a probability $T$ that an adversary might allow for his failure.

We assume for simplicity a static network that does not change for the period of our analysis. In each multicast session, a new random MR is chosen and the circuits constituting the multicast tree are also reformed. We also assume that all members of the cabal participate in all multicast sessions (meetings) and that cabal composition does not change. If a cabal member constructs a circuit that uses a middle relay controlled by the adversary, then the adversary learns the client's guard for that circuit. We take this as the only way that the adversary learns guards.

The probability that a given cabal member never uses a compromised middle relay in any of $m$ sessions is $(1-B)^m$. Given the simplifying assumption that such compromise is independent for all clients, the probability that a guard of every one of the $c$ cabal members is identified at least once over the course of $m$ meetings is thus $(1-(1-B)^m)^c$. We can then gauge adversary success by bounding the probability that the adversary fails to carry out this compromise, giving us

$$1-(1-(1-B)^m)^c < T. \quad (1)$$

We now explore the parts of the $(c, m, B, T)$ space that satisfy this inequality. If the number of meetings

$$m > \log_{1-B}\left[1-(1-T)^{1/c}\right], \quad (2)$$

where $B \in (0,1)$, then, with probability at least $1-T$, the adversary learns at least one guard of each of the $c$ cabal members. The left column of Fig. 1 plots the right-hand side of (2) as a function of $B$ for $c = 5$, 20, and 50 (top to bottom subplots) and $T = 0.1$, 0.5, and 0.9 as indicated on the curves within each subplot. This gives the expected (minimum) number of meetings $m$ required for the adversary to learn a guard of each member of a cabal of size $c$ with allowed failure probability $T$. Thus, for cabal sizes up to 50 an adversary holding 10% of middle-relay bandwidth has a 90% chance of having identified a guard for each cabal member after no more than 60 meetings. After 10 meetings, however, even for $c = 5$ the adversary must hold a more ambitious middle-relay bandwidth of nearly 40%. Note that the more relays the adversary introduces the more gradually he must introduce them and the more the relays should be in multiple locations if he is to avoid suspicion.

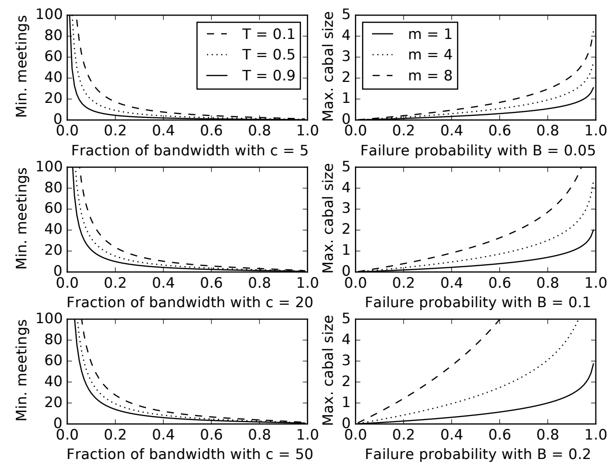


**Figure 1: Left column: The expected (minimum) number of meetings $m$, as a function of middle-relay bandwidth $B$ controlled by the adversary, required for the adversary to learn a guard of each cabal member for cabal sizes $c = 5, 20,$ and $50$ (top to bottom). Different curves in each subplot correspond to different failure probabilities for the adversary as indicated. Right column: The expected (maximum) cabal size $c$, as a function of the adversary's failure threshold $T$, allowed for the adversary to learn a guard of each cabal member when the adversary controls $B = 0.05, 0.1,$ and $0.2$ of the middle-relay bandwidth (top to bottom). Different curves in each subplot correspond to different numbers of cabal meetings as indicated.**

Again using (1), if the number of cabal members

$$c < \log_{1-(1-B)^m}\left[1-T\right], \quad (3)$$

where $B \in (0,1)$ and $m$ is a positive integer, then the adversary will learn at least one guard of each of the $c$ cabal members. The right column of Fig. 1 plots the right-hand-side of (3) as a function of $T$ for $B = 0.05$, 0.1, and 0.2 (top to bottom subplots) and $m = 1$, 4, and 8 as indicated on the curves within each subplot. This gives the expected (maximum) cabal size $c$ allowed for the adversary to learn a guard of each cabal member.

From the cabal's perspective, it should keep its number of members minimal with respect to the qualities needed to accomplish its

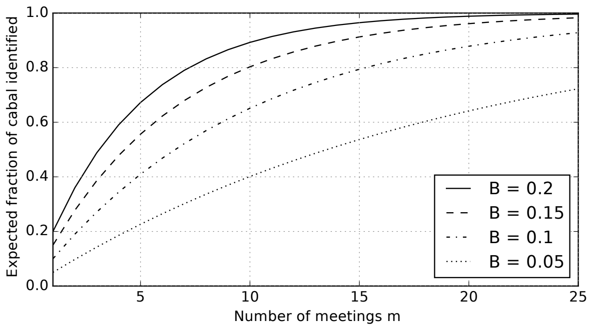


**Figure 2: The expected fraction of the cabal members for which the adversary identifies a guard as a function of the number of meetings $m$. Different curves show different fractions $B$ of middle-relay bandwidth controlled by the adversary.**

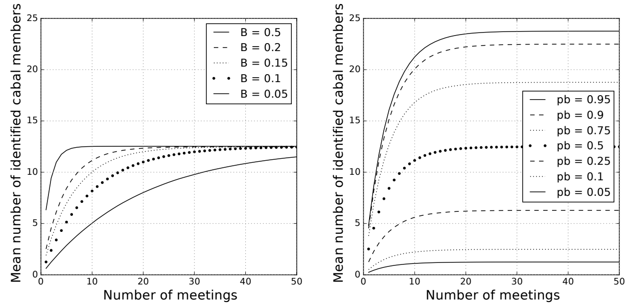


**Figure 3: Mean (over 10,000 trials) number of cabal members (out of 25) identified as a function of the number of cabal meetings. Different curves show $B$ = 0.05, 0.1, 0.15, 0.2, and 0.5 for $p_b = 0.5$ (left) and $p_b$ = 0.05, 0.1, 0.25, 0.5, 0.75, 0.9, and 0.95 for $B = 0.2$ (right).**

goals. However, the cabal might reasonably ask how many meetings it should hold and what the effects of additional meetings are on its security. Figure 2 shows the expected fraction of cabal members with identified guards after $m$ meetings for different fractions $B$ of middle-relay bandwidth controlled by the adversary. Perhaps unsurprisingly, failure probability appears to be highly responsive to either the number of meetings observed or the fraction doing the observing.

*3.2.2 Estimating cabal size.* In the extended version of this work, we show that being selected as MR is all the adversary needs beyond the guard of a targeted cabal member for a fairly accurate estimate of cabal size. Relatedly, for such cabals on the current Tor network, a star topology or a tree that is almost a star is reasonable to expect. Further, a $B = 0.2$ adversary will have a nearly 90% chance of being chosen after 10 multicast sessions [13].

Estimates based on relays serving as middles (second hops) in MTor sessions are also possible. Compared to results from compromising a multicast root, these will be much less likely to have information about the circuits of all cabal members but much more likely to provide some information about cabal size every session. In addition, middle relays will identify guards of cabal members with every observed cabal connection. Since these points will apply equally well to IRC cabals, we focus on middle-relay-based estimation and direct readers to the extended version for MR-based analysis of cabal size.

**Setting and assumptions:** Here, we look at some very basic numerical simulations of the information learned by the adversary from middle relays in the MTor usage scenario. These make assumptions that parallel those made in our analysis above, but they allow us to study the effects of MTor deduplication more easily. In particular, we assume a static network and that cabal members choose guards and middle relays uniformly at random from sets of 2500 and 5000 such nodes, respectively.

**Approach:** In each of 10,000 trials, we identify some middle relays as compromised; each is compromised independently with probability $B$, and the compromise status remains unchanged throughout $m$ cabal meetings. We then choose a set of guards (the size of which depends on the simulation) for each cabal member; these sets are unchanged throughout the $m$ meetings.

For each meeting, each guard selects a client from its set. For each guard selected, we then select a middle relay to use to connect to the MR. (This simulates the choice of a middle relay made by the first client using that guard in that meeting. Any other clients who use that guard for that meeting will use the same middle relay.) If that middle relay is compromised and the adversary has not attempted to bridge the guard before (i.e., that guard has not been used in conjunction with a compromised middle relay in this trial), then we bridge the guard with probability $p_b$. Across all meetings in the trial, we keep lists of the guards that have been bridged and that the adversary has tried but failed to bridge. Once the newly-compromised guards in a meeting are determined, we determine the clients that have been identified; this set is the union of the clients that were compromised before the current meeting and all of those that, in the current meeting, used a guard that has been successfully bridged during any meeting up to this point (including the current meeting).

As noted in Sec. 2, a guard might be bridged in many ways, and we expect an adversary to try them in parallel and/or successively. He will likely start with the least costly, least likely to raise suspicion, and most likely to succeed, perhaps with varied emphasis depending on setting. We use $p_b$ to capture the cumulative probability of success of these different approaches.

**Results:** Figure 3 shows the mean number of identified cabal members, out of 25 and averaged over 10,000 random trials, as a function of the number of meetings. Different curves correspond to different values of $B$ for $p_b = 0.5$ (left) and different values of $p_b$ for $B = 0.2$ (right).

Both $B$ and $p_b$ can have a significant impact on the adversary's success. Figure 3 illustrates the benefit to the adversary of having additional guards that he may attempt to bridge (for a fixed $p_b$). As we also discuss below, the adversary might be able to increase $p_b$ against long-lived guards by continuing to devote resources to bridging them if initial attempts fail.

# 4 EXAMPLE: IRC CABALS

We now consider the following scenario: A cabal of interest to the adversary communicates via a private IRC channel. All of the cabal members access the IRC server only via Tor, and each creates a new Tor circuit for each cabal meeting. The adversary compromises the middle relays independently with probability $B$, corresponding to the fraction of middle-relay bandwidth that he controls. For simplicity of initial analysis, we assume that the network is static.

## 4.1 Identifying guards and cabal members

For learning guards of cabal members, the analysis of Sec. 3.2.1 applies in this setting as well. The assumption made there that probabilities for clients never using a compromised middle relay are independent is even more realistic here.

We also consider the adversary's success in identifying and locating particular cabal members. If the adversary already has a cabal membership, by the properties of IRC he has a list of channel pseudonyms for all cabal members, even those attending meetings silently, and has a complete pseudonmymous record of all communications. We again assume the adversary owns some fraction $B$ of the middle-relay bandwidth. If the cabal leader Alice uses a middle relay controlled by the adversary, he observes the traffic pattern through that relay and the matching messages that appears in the channel and thus learns the guard used by Alice for that circuit. Once the adversary knows this guard, he is able to bridge the guard and identify Alice's IP address with probability $p_b$. Other than through this combination of events, we assume the adversary is not able to identify Alice's IP address.

Even if the adversary only owns the ISP of some targeted, uncompromised cabal member, he still passively observes everything there, including the traffic pattern for a cabal's IRC channel; as noted above, we assume that this or other information allows the adversary to identify cabal traffic at middle relays. And if traffic patterns are indicative of cabal leaders, or if content is indicative and the adversary can compromise some cabal member that has been identified, then a cabal leader can still be targeted for guard identification and bridging.

We make the simplifying assumption that bridging is actually with respect to client–guard pairs rather than individual guards. Thus, if clients $c_1$ and $c_2$ use the same guard $g$ for circuits that go through compromised middle nodes (which may or may not be the same for the two clients), then the adversary bridges $g$ and learns $c_1$ with probability $p_b$ and, *independently*, bridges $g$ and learns $c_2$ with probability $p_b$. Bridging that arises from compromising the guard itself would not be independent for these two clients, while bridging that arises from compromising client ISPs or some part of the network between the clients and $g$ might be. We thus think this assumption is reasonable, although others could be made.

Our computation proceeds as follows. With probability $B$ the cabal leader will choose a compromised middle relay during the first meeting, allowing the attacker to learn the leader's guard. With probability $p_b$, the attacker will bridge the guard. Alternatively, the leader does not use a compromised middle relay for the first meeting (which happens with probability $1 - B$, or with probability $(1-B)^i$ for the first $i$ meetings) but then uses a compromised middle relay (with probability $B$) for the second (or $(i+1)^{\text{st}}$) meeting. Once the compromised middle relay is used, then the attacker bridges the leader's guard with probability $p_b$. We note that it only matters when the leader first uses a compromised middle relay—the attacker only has one chance to bridge the leader's guard; if she fails the first time, then we assume that she is not able to successfully bridge that guard on a later occasion that the leader uses a compromised middle relay. Thus, we have that the probability of the adversary successfully bridging the cabal leader's guard is

$$Bp_b + (1-B)Bp_b + \cdots + (1-B)^{m-1}Bp_b = \left[1 - (1-B)^m\right] p_b. \quad (4)$$

Figure 4 plots the probabilities, for various cases, that the adversary is able to identify the cabal leader. Subplots (a) and (b) show this probability as a function of $B$ for $m$ meetings (different curves for $m = 1$, 5, 10, and 20). Subplots (c) and (d) show this as a function of $m$ with different curves for $B = 0.05$, 0.1, 0.15, and 0.2. The bridging probability $p_b$ is either 0.5 ((a) and (c)) or 0.95 ((b) and (d)).

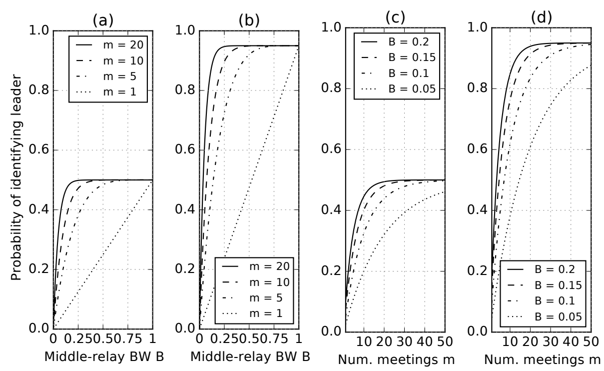


**Figure 4: Probability that the attacker is able to identify the cabal leader when the attacker controls a fraction $B$ of the middle-relay capacity over the course of $m$ meetings. This is shown as a function of $B$ in (a) and (b) (with different curves for different values of $m$) and as a function $m$ in (c) and (d) (with different curves for different values of $B$). The bridging probability $p_b$ is either 0.5 ((a) and (c)) or 0.95 ((b) and (d)).**

Considering Fig. 4, the attacker would likely be able to identify the cabal leader if he made a substantial but not unrealistic investment in middle-relay bandwidth ($B = 0.2$), even with a modest number of cabal meetings (10 or more). Subplot (a) shows the adversary's chances bounded by $p_b = 0.5$ because there is only one guard to bridge. However, we expect that, if there is a single long-lived guard, the adversary's chance of bridging the guard would go up over time, and the success probabilities would become closer to those shown in subplot (b).

## 4.2 Estimating cabal size

We turn now to estimating the total size of the cabal (when the adversary does not already own a member). In particular, if an IRC cabal of size $c$ has $m$ meetings, what estimate of $c$ will be made by an adversary who controls a fraction $B$ of middle-relay bandwidth? How is this estimate likely to be distributed, and how much of an error is he likely to make?

To answer these questions, we numerically compute the maximum-likelihood estimator (MLE) of the cabal size based on an $m$-tuple $\vec{x} \in \{0, 1, \ldots, c\}^m$ of observations that the adversary might make of the number of cabal members using compromised middle relays during each of $m$ meetings. We then compute the distribution on possible observations to determine, for each estimated value, the probability that the adversary will make observations that give rise to that estimate.

In doing this, we assume that the adversary considers only the number of circuits that he sees during each meeting window and not the guards used for these circuits. Given our assumptions of a static network with the default single guard per client, if the adversary sees one circuit in each of three meetings and these circuits all use different guards, then he knows that he has observed three different clients and not just one client multiple times. As a result, he should probably increase his estimate of the cabal size compared to his estimate when simply using a count of the circuits he observes. The approach we take already requires nontrivial computational resources, and accounting for guard identities makes it even more complex to the point that we expect it would be infeasible (while also not adding significantly to the adversary's accuracy). We do note that this issue may have an effect. For clients choosing one guard each out of 2,500 total guards, the probability that at least two clients share a guard for 3, 5, 10, 20, and 25 clients is 0.1%, 0.4%, 1.8%, 7.3%, and 11.3%, respectively.

Figure 5 presents a matrix of plots of distributions of the MLE value. Each subplot shows, for each value $c$ on the horizontal axis, the distribution of MLE values, with larger probabilities corresponding to darker shading. In the matrix of plots, $B$ increases from left to right (0.05, 0.1, 0.15, 0.2, 0.5, 0.75, and 0.9); $m$ equals 2 in the top row and 5 in the bottom row.

While the distributions may be spread out and discontinuous, by the time the cabal has $m = 5$ meetings and the adversary controls $B = 0.2$ of the middle-relay bandwidth, the distribution starts to converge and the probability of a substantial error is fairly small. (See additional analysis of expected errors in [13].)

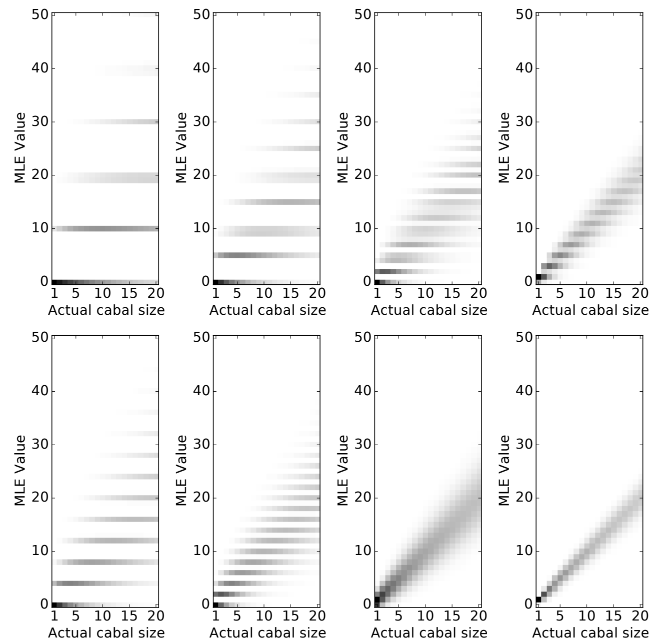


**Figure 5: The effect of increasing the number of meetings $m$ and changing the adversary-controlled middle-relay bandwidth $B$. Top: $m = 2$; bottom: $m = 5$. From left to right, $B = 0.05$, $0.1$, $0.2$, and $0.5$. Within each subplot, the actual cabal size $c$ increases from 1 to 20 along the horizontal axis, and possible MLE values increase from 0 to 50 up the vertical axis. For each $c$ and MLE value, the plot indicates the probability (darker values are larger) of the adversary observing a tuple for which it would compute the indicated value as its MLE for $c$.**

# 5 EXAMPLE: PICKING RIPE ONIONS

The set of users of a particular site may be similar to a cabal communicating via multicast or IRC. While they might not be holding simultaneous meetings or even see themselves as a group, an adversary may target them because they are users of that site, which might be of interest for a variety of reasons.

Our analysis here essentially applies to Tor users visiting many ordinary Internet sites, but we focus on onionsites, particularly hidden web services. These were designed to hide many features typically visible for ordinary websites. They have also had recent design changes specifically intended to make it harder for an adversary to discover a site's .onion address, popularity [22] or network location [20]. Beyond this inherent interest, such sites are plausible candidates for targeting of their users.

As noted, learning about site popularity may be an adversary goal for a targeted site or may be a criterion for deciding to target a site. Previous work [3, 26] has measured popularity of onionsites by requests to the onion address directory. Besides the directory system design changes that make this approach much less feasible, it can also be a misleading and inaccurate measure of onionsite popularity in several ways [23]. We can, however, use variants of the techniques described in previous sections to measure onionsite popularity (and the popularity of other Tor-accessed sites).

Directory requests and even site connections can be unreliable indicators of human interest in onionsites because of crawlers and automation such as link prefetching. Knowing the number of connections or distinct clients connecting to a site is not nearly as useful to a targeting adversary as would be having information about the distribution of connections among users, both for understanding a site's popularity and for selecting its users for targeting.

Unlike the cabal-meeting case, because visits are not synchronized the targeting-adversary technique of starting with middle relays might seem problematic for individuating onionsite visitors. But, as with our analyses above, distinct guards are a fairly accurate indicator of distinct clients up to a moderate-size set of clients.

Even if site users do share guards, as long as guard overlap is infrequent enough, this will still give a targeting adversary a much better idea about interest in the site than could be obtained via previously published techniques. "Infrequent enough" implies that the rough picture of popularity painted by targeted-site connections per guard per unit time presents a ballpark estimate of site-user activity distribution.

To use our middle-relay techniques, we must assume that destinations of potential interest can be recognized by an adversary in the middle of circuits. We do so based on traffic patterns plus possibly other factors like latency, which has been known since at least 2007 [11] to leak some identifying information for Tor communication. Destination fingerprinting of route-protected communications predates Tor [10] and continues to be a topic of study.

When monitoring for a single website, as when targeting a particular site for attack, Wang and Goldberg showed an open-world true-positive rate of 97% and false-positive rate of 0.2%, with sites [36]. And, onionsite fingerprinting is likely to be much more effective than fingerprinting destinations from within circuits that exit the Tor network. There are only about 60,000 unique onion addresses in the directory system, and of these typically only a few thousand at any time are reachable and return a connection to a server. (Our techniques are unaffected by whether a website is listed in the onionsite directory system or requires authentication to be reachable.) Further, current onionsite protocols are sufficiently different from applications connected over vanilla Tor circuit protocols that separating these at middle relays is very easy and reliable for relays carrying their streams, and Juarez et al. recently found 99.98% accuracy for such separation [19]. Targeted onionsites may also be more fingerprintable than typical popular sites. For example, against malicious applications monitoring hardware performance events, whistleblower sites were found to be much more susceptible to fingerprinting than top Alexa sites [9]. Also, an adversary looking for the significant users of a targeted onionsite is likely to tolerate a high initial false-positive rate, especially if fingerprinting is not just of a landing page, but of selectable subpages linked from it, and of other fingerprintable aspects of site usage that may occur during an active connection. Accuracy of fingerprinting targeted onionsite destinations from middle relays is now being investigated, but the above points indicate it is likely to be sufficiently accurate for our purposes.

The most direct technique for a middle-relay targeting adversary is then to simply count the number of connections going to a particular onionsite from each guard. This will already give a rough picture of the distribution of client activity as well as which guards are most worth targeting for further adversary interest.

### 5.1 Recounting Onions From Our Past

Given our assumptions about numbers and fingerprintability of oninonsites, and rough numbers and distribution of their users and implications for individuating clients by guards, there are additional estimation techniques at our disposal.

We can estimate the number of clients visiting a site $n$ or more times using capture-recapture techniques. These were originally used for species population estimates in biology where it would be impossible to observe the entire population, e.g., estimating the number of fish in a given lake. They are now used in many settings, including computer networks [1].

We use the Chapman version of the Lincoln–Petersen estimator to compute the number of clients making $n$ or more connections to a target site per unit time. We assume (1) that all clients (targeted or otherwise) visit a target site with the same frequency during different sampling intervals of the same length, and (2) that connections (visits) and observation intervals are such that no connection is counted in more than one interval. We show the results of numerical experiments with this estimator in Fig. 6. In these experiments, we assume 2,500 guards and 5,000 middle relays; each of the latter is compromised with probability $B$. We run each experiment 10,000 times. For each client, we pick a guard. We assume there are two types of clients: "regular" clients who visit the targeted site twice during each examination window and "interesting" clients who visit the targeted site 10 times during each examination window. Each experimental run specifies the number of each type of client. For each client, we have it repeatedly pick (twice or 10 times, depending on its type) a middle relay to use with its guard. We track the number of times that each guard is used with any compromised relay; this could be multiple compromised relays, and the same guard could be used by multiple clients. We specify a threshold value; guards that are seen by compromised middle relays at least this many times are considered marked and are remembered by the adversary. This process is repeated again to model the second examination window.

Note that, if multiple clients used the same guard simultaneously to connect to an onionsite through one or more compromised middles, the adversary might reasonably conclude that the guard is in fact serving multiple visitors to the onionsite. This does not follow with certainty, and we do not model such reasoning here.

The subplots of Fig. 6 show violin plots that focus on varying different parameters; except for the parameter being varied in a particular subplot, these use $B = 0.25$, a threshold of 3, a client mix of 25 targeted clients (who each visit the destination 10 times per period) and 225 "regular" clients (who each visit the destination 2 times per period), and 1 guard per client. The subplots examine the effects of varying $B$ (left; results for $B = 0.05$, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, and 0.55) and the threshold (right; results for thresholds of 1, 2, 3, 5, 7, and 10).

Considering Fig. 6, we see that the adversary's estimate increases in accuracy with $B$, as expected, but much of the gain comes in ensuring that the adversary controls one fifth to one third of the middle-relay bandwidth. We also see that the best threshold seems to be 3 for this combination of parameters; this also appears to be the case for other parameter combinations that we explored (with the regular clients visiting the destination twice and the targeted clients visiting 10 times).

## 6 DISCUSSION

In this paper, we have considered adversaries with different goals and strategies, but often with the same endowment and capabilities as adversaries in previous work. Another important difference only touched on above is how long an adversary may have (or be willing to use) some of his resources. This can affect both attack success and decisions about which attacks to attempt.

We now briefly describe some of the temporally dependent features of an adversary's endowment and strategy, although we leave detailed analysis of this for future work. We then describe possible countermeasures to a targeting adversary, particularly one with temporal limitations on his endowment.

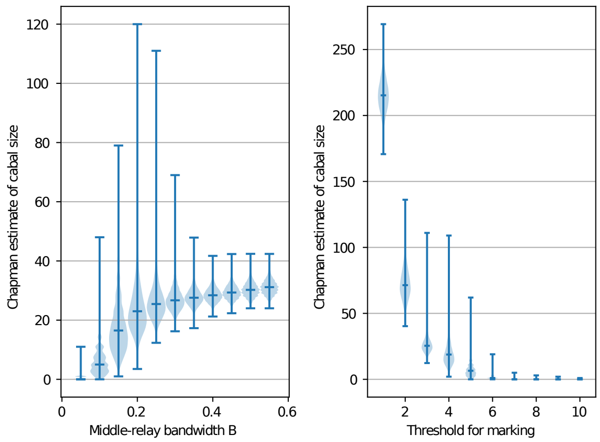


**Figure 6: Violin plots showing distributions of Chapman estimates of cabal size for 10,000 trials each of different values of $B$ with a threshold of 3 (left) and different threshold values with $B = 0.25$ (right); both have 25 targeted clients who visit 10 times per period and 225 regular clients who visit 2 times per period.**

## 6.1 Time Is on My Side

Temporal aspects of adversary endowment have been considered for a long time [24] and have also been applied to onion routing before Tor [30]. We have noted that intersection attacks are inherently across time. Nonetheless, the adversaries in those attacks have a uniform bound on resources; what varies is either the distribution of these [24] or of network resources and status [18]. Generally, these adversaries will still be equally happy to attack any user, or possibly any user with the same behavior on the network. We now discuss a Tor adversary that has, or is willing to deploy, different amounts and types of resources at different times, typically based on some particular target. This might be for various reasons [13].

Johnson et al. [18] set out as a behavioral user class an IRC user, who 27 times a day creates the same single IRC sessions to the same server. We now compare their analysis of IRC users' security against an end-to-end correlating relay adversary to security against the adversary in Sec. 4, targeting a cabal that meets on a private IRC channel. For this user type, they considered variously endowed adversaries and looked in detail at a relay adversary allocated 100 MiB/s, approximately 4% of the total network bandwidth at the time of their analysis, and on the order of the largest identified families of relays under a single operator or organization. The time until an IRC user experienced the first end-to-end correlation failure (both guard and exit of a circuit compromised) was analyzed under an optimal distribution of adversary bandwidth to guards and exits (roughly five to one). In our scenario, a relay is assumed to be able to always identify a cabal connection passing through it. So we should assume that the Johnson et al. adversary is able to devote all relays to the guard position. We thus very roughly estimate a 20% reduction in median time to compromise compared to what Johnson et al. reported.

For a cabal size of 10 or 20, and a roughly comparable fraction of bandwidth allocated to middle relays, our targeting adversary will have a good idea of cabal size and will identify the guards of nearly all cabal members in under 4 days (100 meetings). According to the analysis by Johnson et al., the just-described contributed-guard adversary will require about 10 times as long to get a much rougher idea of cabal size, having identified guards for roughly half the cabal. To get approximately the same likelihood as the targeting adversary of having identified guards for almost all of the cabal will take 40-50 times as long (under a week vs. 150-200 days). This is not just about size: in about a week of IRC usage the targeting adversary will have a good sense of cabal size, cabal guards, and client send-receive activity per cabal guard (which may indicate cabal leaders and *will* indicate which members send the most).

The contributed-guard adversary is also no more likely at any time to identify a cabal leader than any other member. This also holds for the targeting adversary with respect to identifying a leader's guard. But it will still typically take less than a week at the stated rate of meetings. And, for a leader recognized or recognizabile by message patterns, an adversary can tell whether or not it has identified the leader's guard yet or not and can dynamically decide whether to introduce additional resources or attacks to improve or speed up its chances of success. Similarly, once the guard is identified, a determined targeting adversary can bring all resources and attacks to bear to significantly increase his chance of bridging a leader's guard. Some attacks may take weeks to know if they have succeeded, but others will take only hours or even minutes. Recent research has also shown that 90% of Tor relays are vulnerable to relatively easy-to-mount BGP prefix hijack [29], which would quickly bridge any susceptible guard. This alone supports high expectation that a targeted guard will be bridged, even before supplementing with other bridging techniques, such as mentioned in Sec. 2. And in the rare case that a leader's guard might be resistant to all attempts at compromise or bypass, it could be subject to persistent DoS or other resource depletion attacks [14], giving the adversary another chance with a new leader guard.

Johnson et al. also assume a network at steady state, after adversary relays have obtained the guard flag. Middle relays can see some usage in less than a day after announcing themselves and reach steady state in about a week. Relays generally take over a week to obtain the guard flag and about ten weeks to reach steady state for guard usage [5]. For an adversary mounting an attack from scratch, the above time comparison thus overstates significantly in favor of the contributed-guard adversary.

## 6.2 Possible countermeasures

The primary purpose of this paper is to describe a class of adversary that has been overlooked but we believe is as pertinent or more pertinent for Tor than the one generally receiving the most attention. Before concluding, however, we wanted to at least sketch some possible ways to improve resilience against such selective adversaries on the Tor network. In the interest of space (and time) we have limited the scope of our analysis to adversaries at Tor relays, only minimally considering an adversary on the network links between them and/or between clients or destinations and the Tor network. Any countermeasure we describe here will likely need to be significantly redesigned to be effective once those resources are added to the adversary arsenal. So there is little point to providing more than a sketch here.

**Layered Guards:** The same paper that introduced Tor guards also introduced the idea of layered guards [25]. The notion of layered guards has been revisited periodically, most recently in a Tor Proposal on slowing guard discovery for onion services [20]. Using only one or a small set of relays for each client-guard's middle could make it hard to identify guards just as guards make it hard to identify clients. But, if single persistent middles are chosen, then randomly selected exits could possibly enumerate and monitor the behavior of associated users for many destinations and, worse, will always know for which destination without fingerprinting. In the case of fingerprinted onion services, cabal/user enumeration will be possible using (easily spun up) middles chosen as rendezvous points. In general, the number and rotation of guards and their second-layer guards can complicate determination of cabal size as well as guard discovery.

**Randomized selection of guard set size and duration:** As our analysis in [13] shows, single, persistent guards generally provide much more enumeration information about size of a cabal or set of targeted-site users and more information about targeted-site user behavior than does a set of guards with more overlap between clients' sets. And long persistence means that bridging an identifed guard is not needed for monitoring a portion of targeted client's behavior, and that, if a bridging is attempted, it will pay off for a long period of monitoring targeted-client behavior and IP address(es). And that bridging need not be quickly successful to be useful. In addition to enlarging and randomizing size of a client's guard set, selecting guards for less persistent or predictable periods would also counter targeting attacks, pseudonymous profiling, and confidence in the expected value of bridging a guard. Other related strategies may be worth investigating, such as a client limiting and controlling the use of the same guard for visits to the same sensitive site. Obviously there is a tension between the increased risk of correlation attack from using more guards for shorter periods and targeted attacks on cabals or clients of targeted sites.

**Trust:** One way to simultaneously reduce vulnerability from both targeted middle-relay attacks and untargeted correlation attacks is to incorporate trust into route selection. The paper that introduced guards for Tor observed that guards could be "chosen at random or chosen based on trust" [25]. Subsequent work noted that trust could be based on many of the criteria mentioned above as useful for determining which bridging strategies might be effective against which guards and introduced a mathematical characterization of trust as the complement to the probability of compromise [16]. The downhill algorithm [17] explored combining layered guards with trust: A first relay in a circuit is chosen from a small set of those most highly trusted. Later relays in a circuit are selected from ever larger sets that include relays further "down the hill" of assigned trust. This would add delay to enumeration attacks as well as reducing their effectiveness. It could also be combined with varying periods of guard rotation in various ways. An unpublished version of the downhill algorithm had a slowly rotating first-hop relay set and ever faster rotating relay sets for each subsequent hop [27]. For trust-based protections to be effective in practice, trust of all elements in the network path (ASes, IXPs, submarine cables, etc.) must be considered [12].

**Standardized onion service traffic templates:** Our targeting attacks on onion services are dependent on the effectiveness of fingerprint-based individuation of them. Making traffic fingerprints of many onion services similar to each other could reduce the effectivness of those attacks. Providing simple bundles or templates for users wanting to set up onionsites would be useful for many reasons, incorporating data management and communication protocols to create default standardized traffic fingerprints among them. To reduce the likelihood that a single draconian standard will discourage site operators from using these defaults, a small number of templates might be available depending on basic site configuration choices. Sites using standardized templates can also communicate this to clients either in their directory information or upon first contact. Traffic fingerpint normalization can then be enhanced by cooperation between clients and sites. To further facilitate adaptation to individual sites, onion protocols could use componentized chunks of fingerprint-normalized communication, possibly split over multiple circuits. Again the trade-offs against correlation vulnerability would need to be considered, but we hope we have shown that fingerprinting by interior elements of the network is as realistic or more realistic and serious a threat to Tor's most sensitive users.

## 7 CONCLUSIONS

We have introduced targeting adversaries, who focus on particular system users or groups of users, and shown that the difference in goals and strategy of these overlooked adversaries can lead to attacks on Tor users at least as devastating and relevant as any Tor attacks set out in previous research. While we have shown the capabilities of a targeting adversary in realistic scenarios involving a published multicast system, IRC, and onionsites, we cannot hope to quantify in this introductory treatment the possible effects for all Tor users, for all possible client locations and network destinations. We anticipate extensive future research into targeting adversaries, including: abstract characterization and formal treatment of targeting adversaries, expansion of contexts in applied targeting adversary models (such as network link adversaries), and analysis of security against combinations of targeted and hoovering adversaries, particularly when large and well-resourced. And we expect all of this to have an impact on future guard and path selection algorithms.

Onion services are also an active area of discussion and redevelopment [22, 37]. We expect that an interesting and useful direction for future research will be the analysis of the effects of different redesign proposals on security in the context of targeting adversaries. This will require a substantial extension to TorPS [15], which does not currently support modeling of onion services.

## ACKNOWLEDGMENTS

We thank Giulia Fanti, Nick Hopper, Rob Jansen, George Kadianakis, Matt Traudt, Ryan Wails, and the anonymous referees for their comments on earlier drafts of this paper.

## REFERENCES


[1] Nicola Accettura, Giovanni Neglia, and Luigi Alfredo Grieco. The capture-recapture approach for population estimation in computer networks. *Computer Networks*, 89:107–122, 2015.

[2] J. Appelbaum and A. Muffett. The ".onion" special-use domain name. https://tools.ietf.org/html/rfc7686, 2015.

[3] Alex Biryukov, Ivan Pustogarov, and Ralf-Philipp Weinmann. Trawling for Tor hidden services: Detection, measurement, deanonymization. *Proc. IEEE SP*, 2013.

[4] George Danezis and Paul Syverson. Bridging and fingerprinting: Epistemic attacks on route selection. *Proc. PETS*, pages 151–166, 2008.

[5] Roger Dingledine. The lifecycle of a new relay. https://blog.torproject.org/blog/lifecycle-of-a-new-relay, September 2013.

[6] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The Second-Generation Onion Router. *Proc. USENIX Security Symposium*, 2004.

[7] Nick Feamster and Roger Dingledine. Location diversity in anonymity networks. *Proc. ACM WPES*, pages 66–76, 2004.

[8] David Goulet and George Kadianakis. Random number generation during Tor voting (Tor proposal #250). https://gitweb.torproject.org/torspec.git/tree/proposals/250-commit-reveal-consensus.txt, August 2015.

[9] Berk Gulmezoglu, Andreas Zank, Thomas Eisenbarth, and Berk Sunar. PerfWeb: How to Violate Web Privacy with Hardware Performance Events. *Proc. ESORICS, Part II*, pages 80–97, 2017.

[10] Andrew Hintz. Fingerprinting websites using traffic analysis. *Privacy Enhancing Technologies: Second International Workshop (PET)*, pages 171–178, 2002.

[11] Nicholas Hopper, Eugene Y. Vasserman, and Eric Chan-Tin. How much anonymity does network latency leak? *Proc. ACM CCS*, pages 82–91, 2007.

[12] Aaron D. Jaggard, Aaron Johnson, Sarah Cortes, Paul Syverson, and Joan Feigenbaum. 20,000 in league under the sea: Anonymous communication, trust, MLATs, and undersea cables. *Proc. Privacy Enhancing Technologies*, 2015(1):4–24, April 2015.

[13] Aaron D. Jaggard and Paul Syverson. Onions in the crosshairs: When The Man really *is* out to get you, 2017. https://arxiv.org/abs/1706.10292.

[14] Rob Jansen, Florian Tschorsch, Aaron Johnson, and Björn Scheuermann. The sniper attack: Anonymously deanonymizing and disabling the Tor network. *Proc. NDSS*, 2014.

[15] Aaron Johnson. Tor Path Simulator code. https://github.com/torps/torps.

[16] Aaron Johnson and Paul Syverson. More anonymous onion routing through trust. *Proc. IEEE CSF*, pages 3–12, 2009.

[17] Aaron Johnson, Paul Syverson, Roger Dingledine, and Nick Mathewson. Trust-based anonymous communication: Adversary models and routing algorithms. *Proc. ACM CCS*, pages 175–186, 2011.

[18] Aaron Johnson, Chris Wacek, Rob Jansen, Micah Sherr, and Paul Syverson. Users get routed: Traffic correlation on Tor by realistic adversaries. *Proc. ACM CCS*, pages 337–348, 2013.

[19] Marc Juarez, Rob Jansen, Rafael Galvez, Tariq Elahi, Claudia Diaz, and Matthew Wright. Poster: Fingerprinting hidden service circuits from a Tor middle relay. *Proc. IEEE SP*, 2017.

[20] George Kadianakis and Mike Perry. Defending against guard discovery attacks using vanguards (Tor proposal #247). https://gitweb.torproject.org/torspec.git/tree/proposals/247-hs-guard-discovery.txt, July 2015.

[21] Dong Lin, Micah Sherr, and Boon Thau Loo. Scalable and anonymous group communication with MTor. *Proc. Privacy Enhancing Technologies*, 2016(2):22–39, June 2016.

[22] Nick Mathewson. Next-generation hidden services in Tor, (Tor proposal #224). https://gitweb.torproject.org/torspec.git/tree/proposals/224-rend-spec-ng.txt, November 2013.

[23] Nick Mathewson. Some thoughts on hidden services. https://blog.torproject.org/blog/some-thoughts-hidden-services, December 2014.

[24] Rafail Ostrovsky and Moti Yung. How to withstand mobile virus attacks. *Proc. ACM PODC*, pages 51–59, 1991.

[25] Lasse Øverlier and Paul Syverson. Locating hidden servers. *Proc. IEEE SP*, pages 100–114, 2006.

[26] Gareth Owen and Nick Savage. The Tor dark net. Paper Series: No. 20 20, Global Commission on Internet Governance, September 2015.

[27] Personal communication.

[28] Michael Reiter and Aviel Rubin. Crowds: Anonymity for web transactions. *ACM Trans. Information and System Security (TISSEC)*, 1(1):66–92, 1998.

[29] Yixin Sun, Anne Edmundson, Laurent Vanbever, Oscar Li, Jennifer Rexford, Mung Chiang, and Prateek Mittal. RAPTOR: Routing attacks on privacy in Tor. *Proc. USENIX Security*, 2015.

[30] Paul Syverson, Gene Tsudik, Michael Reed, and Carl Landwehr. Towards an analysis of onion routing security. *Proc. International Workshop on Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability*, pages 96–114, 2001.

[31] Who uses Tor. https://www.torproject.org/about/torusers.html. Accessed February 2017.

[32] Tor metrics portal. https://metrics.torproject.org/.

[33] Tor network size. https://metrics.torproject.org/networksize.html.

[34] The Tor Project. https://www.torproject.org/.

[35] Ding Wang, Zijian Zhang, Ping Wang, Jeff Yan, and Xinyi Huang. Targeted online password guessing: An underestimated threat. *Proc. ACM CCS*, pages 1242–1254, 2016.

[36] Tao Wang and Ian Goldberg. On realistically attacking Tor with website fingerprinting. *Proc. Privacy Enhancing Technologies*, 2016(4):21–36, October 2016.

[37] Tim Wilson-Brown, John Brooks, Aaron Johnson, Rob Jansen, George Kadianakis, Paul Syverson, and Roger Dingledine. Rendezvous single onion services, (Tor proposal 260. https://gitweb.torproject.org/torspec.git/tree/proposals/260-rend-single-onion.txt, 2015.

[38] Matthew Wright, Micah Adler, Brian Neil Levine, and Clay Shields. Defending anonymous communication against passive logging attacks. *Proc. IEEE SP*, pages 28–43, 2003.