POSTER: Improving Anonymity of Services Deployed Over Tor by Changing Guard Selection

Abhishek Singh
University of Oslo, Norway

ABSTRACT

Many P2P applications are emerging that use Tor to ensure anonymity of their users. Each user in such an application creates an onion service so that the user can receive requests from other users. Such large-scale use of onion services leak a lot of sensitive information to guards in Tor. The cause of these leaks is diversity in guards' resources and the guard selection algorithm in Tor that is designed to use guards' resources efficiently. We describe a preliminary approach for selecting guards which reduces the amount of sensitive information leaked to guards while using guards' resources with same efficiency. Experiments in the context of a P2P publish/subscribe application shows that the approach reduces information leaked to guards by 25%.

CCS CONCEPTS

• **Security and privacy → Pseudonymity, anonymity and untraceability**; *Privacy-preserving protocols*; Economics of security and privacy;

KEYWORDS

Tor; Onion Services; Guard Selection

1 INTRODUCTION

Tor hosts a large number of anonymous services which are referred to as *onion services*. There were nearly 60 thousands onion services in the beginning of August 2017 [6]. These services are used for various applications; for example, file sharing [1], chat [2], VoIP [3] and censorship avoidance [4]. There are emerging class of P2P applications (such as BitTorrent [7] and P2P publish/subscribe [8]) where each user creates an onion service.

A user sends a message using Tor to its destination by forwarding it through an *onion circuit* which consists of a chain of randomly selected relays. The message is encrypted in multiple layers at the source and a layer of encryption is removed at each hop of the onion circuit. The relay at the first hop of an onion circuit C created by a user U is referred to as the *entry guard* for U in C. An entry guard plays a crucial role in preserving user's anonymity as the entry guard knows user's IP address. A user selects an entry guard from a publicly available list of potential guards called *Tor consensus*. Each user does the selection independently of other users and does not disclose the entry guard identity to them.

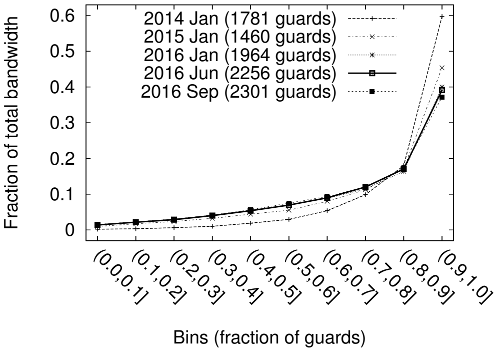

Figure 1: Fraction of total bandwidth contributed by different set of guards over a period of time. Guards are sorted by their bandwidth and then they are grouped together into bins.

There are guards with malicious intent of deanonymizing users of Tor. Such malicious guards have been used to identify an onion service provider [1] and to learn which onion services are of interest to a specific user [5]. The risks of exploits become especially severe when the same malicious guard is selected both by the user and by the onion service provider for the onion circuit between them. In this case, the guard can identify this situation and monitor communication patterns between the user and the service. This can be further correlated with additional information and exploited in the context of various applications [9].

In this work, we show experimentally that the currently used guard selection mechanism in Tor is deficient because it results in relatively many instances when a guard is selected by both ends of an onion circuit. The probability of selection in Tor is proportional to the amount of bandwidth contributed by the guard [2]. Biasing the selection towards guard's bandwidth helps Tor to scale and support a higher number of users. The fundamental problem, however, is that it increases the probability for an onion service and a service user to select the same bandwidth-rich guard. The problem is exacerbated by the fact that the bandwidth contributed by guards is highly skewed. Figure 1 shows this skewness: 10% of guards with most bandwidth provide 39% of all bandwidth in the Tor infrastructure in June 2016.

We propose an improved guard selection algorithm for Tor that reduces the probability of a guard being selected by both ends of a circuit. The algorithm retains the selection bias by the amount of bandwidth contributed by the guard, which is important for scalability.

[1] https://github.com/Tribler/tribler, https://wiki.vuze.com/w/Tor_HowTo
[2] https://github.com/prof7bit/TorChat
[3] https://trac.torproject.org/projects/tor/wiki/doc/TorifyHOWTO/Mumble
[4] https://www.facebook.com/facebookcorewwwi/

CCS '17, October 30-November 3, 2017, Dallas, TX, USA
© 2017 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-4946-8/17/10.
https://doi.org/10.1145/3133956.3138835

In Section 2, we describe the case when a guard is likely to be selected by both ends of an onion circuit, and we discuss trade-offs between scalability of the Tor infrastructure and the anonymity for selection of entry guards by onion services. We outline requirements for modifying the guard selection algorithm in Section 3. We present the intuition behind our approach in Section 4. The key idea is to change the guard selection algorithm for an onion service to take into account number of users accessing it. Experiments in case of a P2P publish/subscribe application show that the approach reduces the amount of information leaked to guards by 25%.

2 PROBLEMS AND CHALLENGES

Disclosure of Vulnerable Communication Links. Guards learn of a large number of users' access to onion services with P2P applications that use Tor. This is because: 1) a peer-to-peer connection from user A to user B is implemented as user A accessing user B's onion service, and 2) each user establishes peer-to-peer connections with many users. Consider an example of a P2P publish/subscribe application that disseminates micro-news from a popular user U to 2256 users. This scenario runs on a Tor network from June 2016 consisting of 2256 guards. Due to bias in selection of guards in Tor, guard G_l with the largest bandwidth in Tor contains 14 users and it learns of any circuit among these 14 users. In contrast, only 1 user selects G_l with the optimal strategy and guard G_l does not learn of both ends of any circuit created by the user.

We use *vulnerable communication link* to refer to disclosure of a circuit between two users using same entry guard. Specifically, two users have a vulnerable communication link between them if: 1) they use the same entry guard, and 2) there is a non-zero probability of them communicating. In the above example, $\binom{14}{2}$ vulnerable communication links are exposed to guard G_l. The actual use of a vulnerable communication link between two users depends on the application. In general, a vulnerable communication link between two users is used earlier (which reveals the link to their entry guard), in following situations: 1) when a user changes its communication partners frequently (for example, to fetch a missing chunk from a newly discovered peer in a BitTorrent session), 2) when users are not available all the time, and 3) when a user uses the application for a long duration (as in the case of P2P publish/subscribe).

Challenges in Reducing Vulnerable Communication Links. The challenge in providing the reduction is to ensure following goals at the same time: scalability of Tor infrastructure and high randomness for the selection of a user's entry guard. We discuss few naive approaches to demonstrate the difficulty in satisfying both goals.

Using the optimal strategy of selecting guards provides high randomness for selection of a user's entry guard and it minimizes vulnerable communication links. However, the strategy limits scalability of Tor as a guard with low bandwidth will exceed its resource limits with relatively small increase in number of users.

Another approach for the reduction is to: 1) partition users in to sets such that users in a set do not communicate with any user in the set, and 2) assign users in a set to the same entry guard. Achieving such a partitioning in a deterministic way is difficult as it requires users to disclose their communication partners. Thus, we focus on a probabilistic approach to partition users in a set that have a low probability of communicating among them.

Probability of two users to communicate is low if the users communicate with few other users. The heuristic is used to partition users such that number of users selecting a guard is proportional to the guard's bandwidth. This approach has following steps: 1) sort users in increasing order of number of their communication partners, 2) sort guards in decreasing order of bandwidth provided by them, and 3) first C_{g1} users select the first guard $g1$ where C_{g1} is proportional to guard $g1$'s bandwidth, and 4) next C_{g2} users select second guard $g2$, and so on. A drawback of the approach is that selection of a user's entry guard is not random. This makes it trivial to locate the entry guard if an estimate for number of communication partners for the user is known.

3 REQUIREMENTS

Our goal is to design a guard selection algorithm that reduces the total amount of vulnerable communication links while adhering to following constraints. First, a user selects his entry guard independently of other users and this is done without communicating with other users. Second, randomness associated with the selection of a user's entry guard should be large enough to prevent the adversary from guessing easily the selected entry guard. Third, in order to avoid load imbalances in Tor infrastructure the number of users that select a guard should be proportional to the guard's bandwidth.

4 PROPOSED APPROACH

We propose a new approach for guard selection where a user takes into account relative popularity of his onion service to select his entry guard. Popularity of a user's onion service provides an estimate of number of users that communicate with the user. This allows partitioning of users into different sets of users such that there is a low probability of users in a set to communicate among them. The approach uses following globally-defined constraints: the lower limit for randomness associated with the selection of entry guard for a user, and the upper limit for maximum probability of selection of a guard to prevent overloading it. The approach requires a user to estimate the fraction of onion services that are less popular than his onion service. One way to estimate this fraction is to adapt the monitoring system [3, 6] which measures number of onion services in Tor. A user uses this fraction to generate an appropriately skewed distribution containing selection probability of guards. A crucial aspect of the approach is that it satisfies the globally defined constraints even though users use different probability distributions for selecting their entry guards.

Intuition. Our idea is for users to generate different probability distributions for selecting their entry guards such that the bias towards guards with large bandwidth in a user's probability distribution decreases monotonically with increase in popularity of his onion service. This is in contrast to Tor where all users use the same probability distribution (refer to the curve labeled "Tor" in Figure 2). The approach ensures that average selection probability across all guards (refer to the curve labeled "Average" in Figure 2) is similar to the probability distribution with Tor; that is, the number of users selecting a guard in the approach is similar to that in Tor. Reduction in vulnerable communication links happens due to following reasons: First, there is a reduction in the number of vulnerable communication links for user U_l having a popular onion service as user

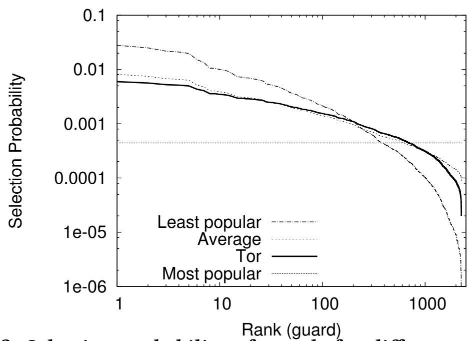

Figure 2: Selection probability of guards for different users with the proposed approach. Guards are ranked in decreasing order of their bandwidth.

U_l is likely to select a guard with small bandwidth. This is because there is a small (or no) bias towards guards with large bandwidth in probability distribution for user U_l (refer to the curve labeled "Most popular" in Figure 2) and there are more guards with small bandwidth in Tor (refer to Figure 1). Second, number of vulnerable communication links disclosed to guard G_l having large bandwidth is reduced as G_l is selected mostly by users with less popular onion services. This is because probability distribution for a user with less popular onion service is skewed towards guards with large bandwidth (refer to the curve labeled "Least popular" in Figure 2). Probability of communication among users that selected guard G_l is low as they are less likely to communicate as they communicate with few partners.

Results. We measured reduction in vulnerable communication links with the proposed approach in the context of a P2P publish/subscribe application [8] which uses Tor to hide communication between users. This publish/subscribe application distributes micro-news messages (such as Tweets in Twitter) from a publisher to all its subscribers. A user and his subscribers form a group. The P2P overlay for the group is constructed independently of other groups. An experiment consists of following steps: 1) users select their entry guards and they construct P2P overlays for distributing micro-news messages, and 2) given an assignment of users to entry guards, we measure the number of vulnerable communication links.

Workload for the experiment consists of 10,000 users which in turn means that there are 10000 groups. Each user has subscriptions (same as followees in Twitter) and a set of users (same as the user's followers in Twitter) that are interested in the user's micro-news messages. The distribution of subscriptions and the distribution of followers in the workload is similar to the one observed in Twitter [4]. Guards and distribution of their bandwidth is taken from the Tor consensus from June 2016 containing 2256 guards.

We measure vulnerable communication links over 10 runs. There were 16147 vulnerable communication links on average with Tor and these links leak information for 5200 groups. There was a reduction of 25% in vulnerable communication links with our approach and the number of groups which can be observed was 18% fewer than in case of Tor.

5 DISCUSSION

A malicious guard can increase the amount of bandwidth it provides so that it can observe more users. The malicious guard will observe more number of vulnerable communication links in Tor than in the proposed approach. This is because with with our approach increasing a guard's bandwidth increases the probability of selection of the guard by users that have less popular onion services. This makes the strategy sub-optimal as the cost of additional bandwidth does not yield proportional amount of disclosures.

In practice, an attacker controls multiple guards [1]. Tor assumes that an attacker controls guards only in a /16 subnet of the Internet as there is no reliable way in Tor to identify if two guards are being controlled by same attacker. This assumption is used in Tor to select relays in an onion circuit and to select multiple entry guards for a user. We outline a way to adapt our approach for handling attackers assumed in Tor's threat model. Guards in a /16 subnet are treated as a single logical entity referred to as *meta guards*. Selection of a user's entry guard is done in two steps: first a meta guard is selected and then a guard from the corresponding /16 subnet is selected. Selection of a meta guard is done in the same way as the selection of a guard in Section 4. Probability of selecting a guard from a meta guard is proportional to its bandwidth in the meta guard. The modification ensures that an attacker does not obtain disproportionate number of vulnerable communication links by increasing bandwidth of its guards in a /16 subnet or by adding more guards in the same /16 subnet.

REFERENCES

[1] Alex Biryukov, Ivan Pustogarov, and Ralf-Philipp Weinmann. 2013. Trawling for Tor Hidden Services: Detection, Measurement, Deanonymization. In *Proceedings of the 2013 IEEE Symposium on Security and Privacy.*

[2] Tariq Elahi, Kevin Bauer, Mashael AlSabah, Roger Dingledine, and Ian Goldberg. 2012. Changing of the Guards: A Framework for Understanding and Improving Entry Guard Selection in Tor. In *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2012).* ACM.

[3] George Kadianakis and Karsten Loesing. 2015. *Extrapolating network totals from hidden-service statistics.* Technical Report. Tor Project. https://research.torproject.org/techreports/extrapolating-hidserv-stats-2015-01-31.pdf.

[4] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. 2010. What is Twitter, a Social Network or a News Media?. In *Proceedings of the 19th International Conference on World Wide Web (WWW '10).* ACM, 10.

[5] Albert Kwon, Mashael AlSabah, David Lazar, Marc Dacier, and Srinivas Devadas. 2015. Circuit Fingerprinting Attacks: Passive Deanonymization of Tor Hidden Services. In *Proceedings of the 24th USENIX Conference on Security Symposium (SEC'15).* USENIX Association, 16.

[6] Karsten Loesing, Steven J. Murdoch, and Roger Dingledine. 2010. A Case Study on Measuring Statistical Data in the Tor Anonymity Network. In *Proceedings of the Workshop on Ethics in Computer Security Research (WECSR 2010) (LNCS).* Springer. https://metrics.torproject.org/hidserv-dir-onions-seen.html.

[7] Ruigrok R.J. 2015. *BitTorrent file sharing using Tor-like hidden services.* Master's thesis.

[8] Abhishek Singh, Guido Urdaneta, Maarten van Steen, and Roman Vitenberg. 2012. On Leveraging Social Relationships for Decentralized Privacy-preserving Group Communication. In *Proceedings of the Fifth Workshop on Social Network Systems (SNS '12).* ACM, New York, NY, USA, Article 5, 6 pages.

[9] Xinyuan Wang, Shiping Chen, and Sushil Jajodia. 2005. Tracking Anonymous Peer-to-peer VoIP Calls on the Internet. In *Proceedings of the 12th ACM Conference on Computer and Communications Security (CCS '05).* ACM.