www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 2 Issue 5 May, 2013 Page No. 1675-1679

THE TUNABLE PATH SELECTION BY USING ONION ROUTING NETWORK

[1]M.Lakshmi, [2]S.Amutha,[3]S.Saranya, [4]T.Tamilmani.

[1, 3 & 4]Dept.of.Computer science, Meenakshi Chandrasekaran College of Arts & Science Pattukkottai-614 626.

[2]School of Computer science, Engineering and applications, Bharathidasan University, Trich23.

vithish88@gmail.com , kumaran.bio82@yhoo.com

***Abstract*—The Tor anonymous communication network uses self-reported bandwidth values to select routers for building tunnels. Since tunnels are allocated in proportion to this bandwidth, this allows a malicious router operator to attract tunnels for compromise. Although Tor limits the self-reported bandwidth, it uses a high maximum value, effectively choosing performance over high anonymity for all users. We propose a router selection algorithm that allows users to control the tradeoff between performance and anonymity. We also propose an opportunistic bandwidth measurement algorithm to replace self-reported values that is more sensitive to load and more responsive to changing network conditions. Our mechanism effectively blends the traffic from users of different preferences, making partitioning attacks difficult. We implemented the opportunistic measurement and tunable performance extensions and examined their performance both through simulation and in the real Tor network. Our results show that users can get dramatic increases in either performance or anonymity with little to no sacrifice in the other metric, or a more modest improvement in both.**

***Keywords:* self-reported, bandwidth, Tor, anonymity**

I. INTRODUCTION

Tor is used by an increasing variety of parties: reporters communicating with sources, dissidents and embassies hiding their activities from local governments, people trying to get around geographic restrictions, and more. However, for the average user, the performance penalty introduced by Tor is still prohibitively high for everyday use. At the same time, the popularity of Tor has lead to development of a number of practical attacks on the system.Efforts to improve the performance of the Tor network can often decrease the anonymity, and vice versa. To address this problem, we propose a user-tunable mechanism for selecting routers based on their bandwidth capabilities. Rather than trying to find a compromise that satisfies both those users who desire strong anonymity protection and those for whom performance is more of a priority, as is done in the current Tor design, we suggest letting users express a preference in the tradeoff between anonymity and performance and make router selections accordingly. We design a mechanism that effectively blends the traffic of users with different preferences, making partitioning attacks difficult. At the heart of our work is the Tor load balancing algorithm. Currently, Tor routers self-report their band- width capabilities, and clients choose them in proportion to their fraction of the overall Tor capacity. This enables a low-resource attack, where routers misreport their bandwidth to be the artificially high and thereby capture a large fraction of tunnels. Additionally, due to constantly changing conditions, self-reported bandwidth is frequently an overestimate of the actual node capacity, leading to unreliable performance delivered to Tor users.

We propose to replace the Tor mechanism with an opportunistic bandwidth measurement mechanism. Due to the complete graph topology of the Tor network, each router will have a chance to interact with most other routers and thus observe their performance empirically. We show through experiments that this mechanism is a suitable replacement for self-reported bandwidth in that it accurately predicts the performance of the routers and is significantly less susceptible to low-resource attacks. Also, since over-utilized routers will show decreased performance, it also helps reduce the long tail of the transfer time distribution, making the worst case significantly better. Our experiments with Tunable Tor show that users can achieve great improvements in performance without sacrificing much anonymity, or significantly increase anonymity protection without any loss in performance. They also allow for moderate improvements in both. This improved flexibility should make Tor palatable to a wider range of users, and thus increase anonymity for everyone due to a larger community.

IMPLEMENTATION OF Tor

A. Tor Design

The Tor network is based on an onion-routing design, where

traffic is forwarded through several routers and multiply encrypted, with each router removing one layer of the encryption. The path through the network a tunnel is constructed in a telescoping fashion, so that each router knows only the previous and the next router in the path. In particular, the first (entry) router knows the source of the tunnel, but not its destination, and the last (exit) router knows the destination but not the source. However, if both routers cooperate, they can use traffic analysis to link communication over the same tunnel; hence there is little benefit to using long paths and in practice Tor path length is set to 3.

Tor routers are registered with a directory service. Each router reports its IP address, public key, policies about what traffic it will accept, and a bandwidth value that is determined by monitoring the peak bandwidth achieved by the router over a period of time. The directory service also maintains statistics about the uptime of each router. The Tor path construction algorithm, executed by the Tor client, will first select all routers that have an acceptable forwarding policy (e.g., many routers are unwilling to serve as exit routers) and then choose a random router out of the list, with the selection weighted by the reported bandwidth. This way, traffic is roughly balanced across Tor nodes in proportion to the bandwidth they have available. To prevent a router from reporting an unreasonably high bandwidth, an upper bound is enforced. To defend against the predecessor attack, recent versions have introduced guard nodes, first described by Wright et al. Each client picks a set of three nodes that will be used as entry routers for all of its tunnels.

Guard nodes are chosen among stable nodes, i.e., nodes with a high uptime that have a bandwidth higher than the median bandwidth reported by all Tor nodes. Fundamentally, Tor forms an overlay network for forwarding traffic, and thus needs to address the performance issues associated with this task. It also has an extra requirement of preserving anonymity, making this task that much more difficult. We next examine two shortcomings of the Tor load-balancing scheme that motivate our work.

B. Advertised Bandwidth

The bandwidth values used in the load balancing algorithm are self-reported by each node and are not verified in any way. This leaves the door open to attacks where malicious nodes can report a higher-than-actual bandwidth so that a larger fraction of tunnels are routed through them. Despite the enforced upper bound, the attack can be quite successful: Bauer et al. report that a small fraction of attacker nodes can attain the first and last node positions (thus violating anonymity) on nearly half the tunnels, despite using the older (and more secure) cap of 1.5 MB/s.

Even when nodes are honest, the reported values can be a poor predictor of the available bandwidth at a node due to changing network conditions and other factors. This makes Tor performance highly variable. Recent studies of Tor show that, although the Tor network provides reasonable bandwidth on most connections, the performance curve has a long tail. In particular, while the median bandwidth is 19 KB/s, the 90^{th} percentile bandwidth is less than a third of that, at 6 KB/s, and there is a significant fraction of tunnels that perform worse still. This presents a poor user experience, especially to users who are browsing the web (the majority of connections in Tor), with connections frequently slowing down.

C. User Heterogeneity

The Tor load balancing algorithm provides a single, static compromise between performance and anonymity. Users who are highly anonymity sensitive (e.g., whistle blowers) might wish to distribute all of the tunnels uniformly across all routers, to prevent (purportedly) high-bandwidth routers from having a higher chance of compromising their traffic. Users who are less privacy- sensitive and are using the network for casual web browsing (e.g., users who want to hide their browsing activities from their neighbors) might value performance more and would be more willing to use high-bandwidth routers more often. By using the same path selection algorithm for both of these, the Tor router selection algorithm sacrifices the needs of both classes.

II. PROPOSED IMPROVEMENTS

To address these issues, the fundamental questions of an overlay network must be readdressed: first, how the performance of a router is measured; and second, given a list of measured routers, how is the route selected. In this work, our performance metric is the bandwidth available to a Tor tunnel, rather than other performance characteristics such as latency or jitter. Our reason for focusing on bandwidth is threefold. First, bandwidth is already a key factor in Tor design. Second, bandwidth is typically a property of a node rather than a link between two nodes, since the bottleneck is likely to be close to the node rather than in the intermediate network. This makes measurements and optimizations much more feasible than for link properties, since for N nodes there are O (N2) links. Additionally, a scheme that optimizes latency is bound to leak at least some information about the starting point of a path, whereas it is possible to optimize bandwidth without such information leaks. Finally, the overwhelming majority of Tor traffic, by both data volume and number of connections, is from web and peer-to-peer traffic. Applications that are relatively insensitive to jitter, and where latency can be treated simply as a part of the total transfer time; when low bandwidth makes this transfer time large, latency effects are negligible. Finally, most latency in Tor comes from poor congestion control handling; observed end-to-end latencies significantly exceed the total network latency.

A. Router Measurement

A simple way to measure the available bandwidth at a router is to perform a probe. Though crude, this mechanism is likely to present a reasonably accurate picture of the performance of a node; probing to determine node availability and therefore reliability is done for high-latency anonymous-communication networks by Echolot. The correlation between probed router bandwidth and subsequently achieved tunnel bandwidth in the real Tor network when the probed router is the bottleneck router for that tunnel. The probe results are a good predictor of tunnel performance; however, probes themselves use up valuable bandwidth, which is a scarce resource in the Tor network. In particular, probes need to appear identical to real traffic, lest a node give priority to probe traffic to enhance its performance, and thus need to generate significant data transfers. Furthermore, since it is not realistic for all nodes to probe all other nodes, this task must be delegated to a small

collection of probing agents, which can act as a point of failure or compromise. For these reasons, we consider bandwidth estimation via active probing to be impractical. We propose instead that opportunistic monitoring be used to measure bandwidth capacity; that is, each router in the Tor network keeps track the bandwidth it has recently seen for each of its peers. In practice, Tor routers communicate with a large set of routers over a short period of time. We observed up to 800 routers contacted within a single day and thus can accumulate a large set of statistics. These statistics can then be aggregated by each router to a single observation per peer and then uploaded to the directory server (as the self-reported bandwidth is currently). The directory server can in turn aggregate these N2 observations into N router evaluations. The naive approach is for each node to use the maximum first, each router using its own view of the network creates the possibility of partition attacks, where an attacker focuses all of its bandwidth on nodes of interest. Thus these nodes, and only these nodes, are more likely to select the attacking nodes when creating tunnels. Additionally, aggregating observations via their maximum allows .spotlight attacks, where an attacker focuses all of its bandwidth of one node at a time for a single measurement interval, ignoring all other nodes. Assuming the maximum age of measurements is long enough; the attacker can thus convince the entire network that its bandwidth is many times the actual value.

B. Variable Router Selection Algorithm

In this section, we propose several modifications to the router selection algorithm used by Tor in order to decrease its vulnerability to subversion as well as provide a better experience for all classes of users; we call this algorithm Tunable Tor, due to its user-configurability. As described in Section 2.3, there is a trade-off between selecting routers for optimal performance and providing maximum anonymity protection. Even if the bandwidth measurements are accurate, using high bandwidth nodes more frequently increases a user's exposure, and some users will wish to pick uniformly from all routers. Others may be willing to expose themselves even more than the current Tor design in order for increased performance. We propose giving users control over this tradeoff by letting them select a point on the anonymity. performance scale either globally (i.e., in the Tor configuration _le), or depending on the task. Providing such flexibility not only helps existing Tor users, but attracts new users to the network as well, improving anonymity for all by increasing the anonymity set [9]. However, care must be taken to avoid partitioning attacks. If it is easy to identify what level of privacy a user is aiming for, the anonymity set may be in fact reduced.

To choose a router given a selection function fs, a list of routers and their rankings must _rst be obtained; while this ranking can be based on any metric9, we propose the opportunistically probed available bandwidth metric described in Section 3.1. This list can be of all routers in the Tor network, or only those matching certain criteria (fast, stable, exit, etc.). If this list is indexed from 0 to n□1, then the router selected is that with the index bn _ fs(x)c, where x is selected uniformly at random from [0; 1). The cumulative distribution function of the probability of choosing any given router is shown in Figure 5 for different values of s; a similar CDF of router selection for the current Tor router selection is included for comparison.10 This procedure is then repeated for any other routers to be selected, enforcing the restriction that duplicate selections are not allowed, nor are nodes within the same /16 subnet or node family. There are several features to note about this algorithm. First, the chance of a router being selected is based on the ranking of its metric, rather than on the metric itself. This means that an attacker cannot simply add a router with a very large amount of available bandwidth to the network and attract a large fraction of all circuits; instead, many routers must be added, each with enough bandwidth to rank highly. Second, fs is well defined for all real s. This means that, should a reason arise for preferring routers with low bandwidth, a negative s can be used.

Also, while there are, in principle, no bounds on the strength of a preference for high bandwidth (i.e., how large an s can be chosen), too high a value can result in nearly always choosing the most-highly ranked router. In this paper, values of s from 0 to 15 are examined for completeness; a value of s = 15 implies that the most highly ranked router in a set of n = 1 700 (a typical number of routers available in the Tor network at any given time) will be chosen 23% of the time11. It should be stressed, however, that a practical upper bound for s is 10, which results in the most highly ranked router being chosen less than 4% of the time in the above scenario. In practice, we observe an additional problem: due to routers frequently joining and leaving the network, a router often lacks any data on the bandwidth of a significant fraction of the total router population. In order to bootstrap data for these routers, we divide the population into those those routers for which we have data (i.e., known routers) and those routers for which we do not (i.e., new routers) as a first step in choosing a router. A population-weighted coin toss is used to choose between these groups; if the population of new routers is chosen, we choose a router uniformly at random, and if the population of known routers is chosen, we use the algorithm described above. This modified algorithm is the one used for the evaluations in Section 4.

III. WHOLE-SYSTEM EVALUATION

In order to evaluate the degree to which the proposed changes meet the dual goals of improving user experience and increasing resistance to subversion, we evaluated them according to two categories of metrics: performance and anonymity.

A. Performance

In order to obtain an accurate picture of what the performance of a Tor network using these proposed improvements would look like, we ran tests using a single client modified to use the Tunable Tor algorithms in the real Tor network and a simulator where all clients use the modified algorithms. The relative agreement between the two sets of results argues for their fidelity.

1. Performance in the real Tor network

To evaluate the performance of the proposed modifications to the Tor protocol, a large number of tests were performed over the Tor network; each trial involved downloading a 1MB file over HTTP using an exit router connected via a high-bandwidth connection to the hosting server.

2. Performance in a Simulated Tor Network

To study the effect of the proposed changes in a network where all clients are choosing paths using the Tunable Tor algorithm and evaluating routers using the Eigen Speed algorithm, we used the flow-level simulator described in earlier. The mix of selection levels is based on the assumption that most users will prefer maximum performance, with a smaller fraction preferring maximum anonymity and a much smaller fraction tuning their performance to each of the intermediate selection levels; under this assumption, the results are relatively insensitive to the exact mix of selection levels used.

Anonymity

We next analyze the effects of tunable path selection on anonymity. One measure of anonymity is how many routers an intelligent attacker must subvert in order to have a high probability of compromising a tunnel. Throughout this section, our threat model is an attacker who can compromise some fraction of the routers in the Tor network, or alternately, eavesdrop on all of their traffic. While these two threats are, for the most part equivalent, compromising routers allows for the .false advertising. attack described below, while eavesdropping eaves dropping does not. Intuitively, it is clear that if routers are chosen uniformly at random, more routers must be compromised in order to achieve a high probability of tunnel compromise, while skewing the selection towards certain routers requires fewer to be compromised (because the attacker can choose to compromise the more popular routers).

IV. RELATED WORK

Whereas our work optimizes tunnel bandwidth, for reasons discussed in earlier, considerable work has been done studying the use of Tor paths optimized for latency as opposed to bandwidth. Sherr et al. propose the use geographic coordinates to create paths that fall within selected bounds and use the family of functions fs described in this paper for a link-based router selection algorithm more suited to optimizing latency. Renner developed a controller for Tor to select paths according to criteria such as avoiding ocean crossings and otherwise minimizing latencies. Reardon and Goldberg show that modifying Tor to run DTLS over each router link and use a single end-to-end TCP session can significantly reduce end-to-end latency and queue lengths and can improve throughput as well.

In general, the problem of measuring and optimizing for latency, and the security implications of doing so, is a complex one and beyond the scope of this work. Our results regarding the variability of Tor performance match a comparative study of Tor and AN. ON performance, which also showed large standard deviations for bandwidth values provided by Tor. Bauer et al. consider distributed probing, perhaps in the style of anonymous auditing, as a means of defending from low-resource attacks. They reject it due to the extra load imposed on the system and the ability of malicious nodes to falsely respond to probes. In our case, the distributed measurements are performed opportunistically and thus impose no extra load on the network, and they correspond to real traffic. Therefore, a node seeking to appear as high-bandwidth has to actually provide good performance to real users.

V. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we have proposed improvements to the existing Tor router bandwidth evaluation and router selection algorithms. We examined these changes individually and in combination, showing that they result in a Tor protocol that is both more secure (since it does not use self-reported bandwidth to choose routers for tunnel creation) and performs better, both in terms of observed performance and in terms of achievable anonymity. Additionally, by allowing the user to select their preferred balance of performance and anonymity, these improvements increases the usability, and therefore the potential user base and security of the Tor network. Evaluations of these changes show that they can result in increasing average throughput by a factor of almost three in exchange for a modest decrease in anonymity, or they can result in drastically improved anonymity while maintaining similar average throughput.

We also show that the improvements we propose can reduce or even eliminate the long tail of the transfer time distribution, greatly improving performance as perceived by the users of the network. We plan to expand on this work in the future in several ways: first, we are currently implementing a more detailed, packet-level simulator of the Tor network; this will increase the fidelity of the simulation by including such effects as variable file sizes, variable intervals between requests, and TCP slow-start behavior. We would also like to examine the other aspects (such as latency) of the tradeoff between performance and anonymity in anonymous networks of varying types. Additionally, we observed a number of interesting characteristics of the Tor network over the course of this study which could provide insight into the observed behavior of the Tor network, and which we would like to study further.

VII.REFERENCES

[1] P. Syverson, G. Tsudik, M. Reed, and C. Landwehr, .Towards an analysis of onion routing security,. in *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, H. Federrath, Ed. Springer-Verlag, LNCS 2009, Jul. 2000, pp. 96.114.

[2] A. Back, I. Goldberg, and A. Shostack, .Freedom systems 2.1 security issues and analysis,. Zero Knowledge Systems, Inc., White Paper, May 2001.

[3] R. Dingledine, N. Mathewson, and P. Syverson, .Tor: The secondgeneration onion router,. in *Proceedings of the 13th USENIX Security Symposium (USENIX Security '04)*, Aug. 2004.

[4] .TorStatus - Tor network status,. http://torstatus.kgprog.com/.

[5] K. Loesing, .Measuring the Tor network,. https: //git.torproject.org/checkout/metrics/master/report/dirreq/ directory-requests-2009-06-25.pdf.

[6] D. Goodin, .Tor at heart of embassy passwords leak,. *The Register*, Sep. 10, 2007.

[7] G. Goodell, S. Bradner, and M. Roussopoulos, .Building a coreless Internet without ripping out the core,. in *Fourth Workshop on Hot Topics in Networks*, College Park, MD, Nov. 2005.

[8] K. Bauer, D. McCoy, D. Grunwald, T. Kohno, and D. Sicker, .Low-resource routing attacks against anonymous systems,. In *Proceedings of the 2007 Workshop on Privacy in the Electronic Society (WPES)*, Oct. 2007.

[9] R. Dingledine and N. Mathewson, .Anonymity loves company: Usability and the network effect,. in *Designing Security Systems That People Can Use*. O'Reilly Media, 2005.

[10] M. Wright, M. Adler, B. N. Levine, and C. Shields, .An analysis of the degradation of anonymous protocols,. in *Proceedings of the Network and Distributed Security Symposium*, Feb. 2002.

[11] .., .Defending anonymous communication against passive logging attacks,. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, May 2003.

[12] D. McCoy, K. Bauer, D. Grunwald, T. Kohno, and D. Sicker, .Shining light in dark places: Understanding the Tor network,. In *Proceedings of the Eighth Privacy Enhancing Technologies Symposium (PETS'08)*, Aug. 2009.

[13] R. Dingledine, .Exit balancing patch,. http://archives.seul. org/or/dev/Jul 2007/msg00022.html, 2007, mailing list post to or-dev.

[14] A. Akella, S. Seshan, and A. Shaikh, .An empirical evaluation of wide area internet bottlenecks,. in *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement (IMC03)*, 2003.

[15] K. Lakshminarayanan and V. N. Padmanabhan, .Some _ndings on the network performance of broadband hosts,. in *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement (IMC03)*, 2003.

[16] J. Reardon and I. Goldberg, .Improving Tor using a TCP-over-DTLS tunnel,. in *Proceedings of the Eighteenth USENIX Security Symposium*, Aug. 2009.

[17] P. Palfrader, .Echolot,. http://www.palfrader.org/echolot/, accessed on 10 August 2009.

[18] R. Gao, C. Dovrolis, , and E. W. Zegura, .Avoiding oscillations due to intelligent route control systems,. in *Proceedings of the 25th IEEE International Conference on Computer Communications. (INFOCOM 2006)*, Apr. 2006. IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING 15

[19] G. Danezis, R. Dingledine, and N. Mathewson, .Mixminion: Design of a Type III anonymous remailer protocol,. in *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, 2003, pp. 2.15.

[20] S. Chen, B. Mulgrew, and P. M. Grant, "A clustering technique for digital communications channel equalization using radial basis function networks," *IEEE Trans. Neural Networks*, vol. 4, pp. 570–578, July 1993.