International Journal of Security and Its Applications
Vol.9, No.11 (2015), pp.391-402
http://dx.doi.org/10.14257/ijsia.2015.9.11.36

A Stealthy Attack Against Tor Guard Selection

Quangang Li[1], Peipeng Liu[2] and Zhiguang Qin[1]

[1]School of Computer Science & Engineering,
University of Electronic Science and Technology of China,
Chengdu 610054, China
[2]Institute of Computing Technology,
Chinese Academy of Sciences, Beijing 100190,China
sophylquang@163.com

Abstract

Tor is a popular low-latency anonymous communication system which could provide anonymity and anti-censorship. Based on previous researches on de-anonymization of Tor, this paper proposes a novel approach to attack users' guard selection which can pose great threat against Tor users' anonymity. Under the current design of Tor, once entry guards are compromised, the probability that an attacker observes both ends of a Tor circuit will be highly improved. Actual and simulated experiments both show that an attacker (e.g., a local or national government which have the power to monitor a Tor user's internet connection) can successfully compromise a specific Tor user's entry guard in about 30 minutes, and this can further help de-anonymize the user's anonymous communication.

***Keywords**: Tor, Attack, Guard Selection, De-anonymization, Stealthy*

1. Introduction

Tor [1] is one of the most popular low-latency anonymous communication systems. Tor has been developing rapidly since it was introduced in late 2003. Up to early 2014, there are about 5000 Tor relays distributed around the world. Nearly 2.5 million users [2] using Tor to protect their communication anonymity.

In order to improve the anonymity and security of Tor, various attacks have been proposed to de-anonymize Tor users. For example, attackers can confirm an anonymous communication from application layer by exploiting browser vulnerabilities or user misuse, [3-4]. Through fingerprinting users' circuit building [5] and interested websites [6-8], researchers can also infer Tor users' hops of paths and destination websites. As Tor can't resist traffic analysis attacks, attacks based on traffic confirmation have attracted more and more attention [9-13], where attackers can inject signals at one end of Tor circuit then detect the signals at the other end.

De-anonymization attacks based on traffic confirmation always start with compromising entry and exit nodes of user circuit, then use traffic analysis to correlate or identify both parties. As the accuracy of traffic correlation becomes more and more acceptable [10-11], this kind of attacks mainly depends on the probability that attackers can observe both ends of user circuit. Since the entry guard mechanism was introduced, guard nodes are compromised more difficultly than exit nodes.

In this paper, we firstly analyze the existing de-anonymization attacks on Tor. Based on the analysis, we propose a novel approach to attack specific Tor users' entry guard selection. Our approach will shorten the time to compromise entry guards without explicitly disturbing users' normal communications,and finally improve the efficiency of deanonymizing Tor. Our contributions can be summarized as follows:

ISSN: 1738-9976 IJSIA

- This paper analyzes the existing de-anonymization researches on Tor, and concludes that compromising entry guards has become an important factor in Tor de-anonymization.
- This paper proposes an attack which can compromise users' entry guards more efficiency, without disturbing their normal communications on Tor.

The rest of the paper is organized as follows. We firstly give an overview of the related work in Section 2. In Section 3, we propose and analyze our approach to attack users' guard selection. We present experimental results in Section 4. Conclusions is discussed in Section 5.

2. Related Work

Tor contains a small set of trusted authoritative directory servers which are responsible for aggregating and distributing signed information about known routers in the network. Tor clients periodically fetch the directory information in order to learn information about other relays in the network, such as their IP addresses, public keys, *etc*.

To build an anonymous connection through the Tor network, a client firstly selects an ordered sequence with (usually) 3 relays. The client then negotiates session keys accompanied by relays. And each relay starts with the first node (*i.e.*, the entry node) in the sequence. The middle node is connected using the encrypted tunnel established with the entry node. The last node(*i.e.*, the exit node) is also connected via the encrypted tunnel established with the middle node. The resulting encrypted tunnel through the Tor network is called a circuit, and then can be used to transmit client application data.

A typical Tor communication is illustrated in Figure 1. And de-anonymization attacks on Tor aim to correlate the two parties which communicate over Tor. According to different technologies and positions adopted, de-anonymization attacks on Tor can be divided into 3 categories (application layer attacks, path or websites inferring attacks, traffic confirmation attacks refered as A, B and C). We use ***exp*** to denote the transmission path between exit node and destination servers, and ***ep*** to denote the entering transmission path between clients and guard nodes.

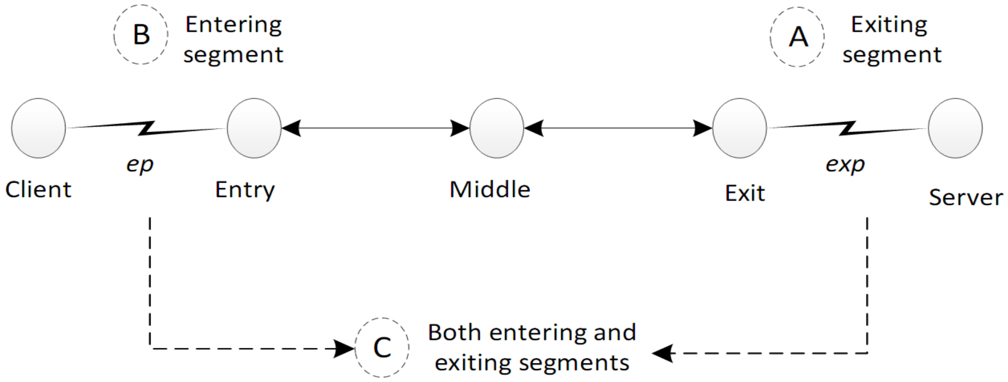


Figure 1. A Typical Tor Communication

Application layer attacks are defined as type A which is illustrated in Figure 2. Attacks are carried out on exiting segment including exit nodes, destination servers and ***exp***. Exploiting vulnerabilities in client browser [3] or user misuse [4], this type of attacks has high accuracy in correlating anonymous communicators. However, due to the strong dependence on specific software vulnerabilities or user configuration, these attacks are not general.

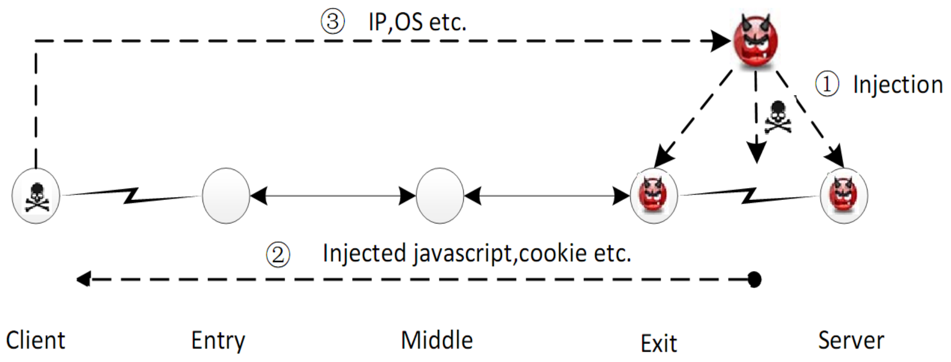


Figure 2. Application Layer Attacks

Path or websites inferring attacks are defined as type B, as shown in Figure 3. Attacks are performed on entering segment including guard nodes and ***ep***. This type of attacks can infer hops of Tor circuits [5] or destination websites [6-8] based on the fingerprinting of sniffed Tor traffic. However, the accuracy of these attacks is highly affected by dynamic network latency, imprecise model detection and the increase of Tor nodes number.

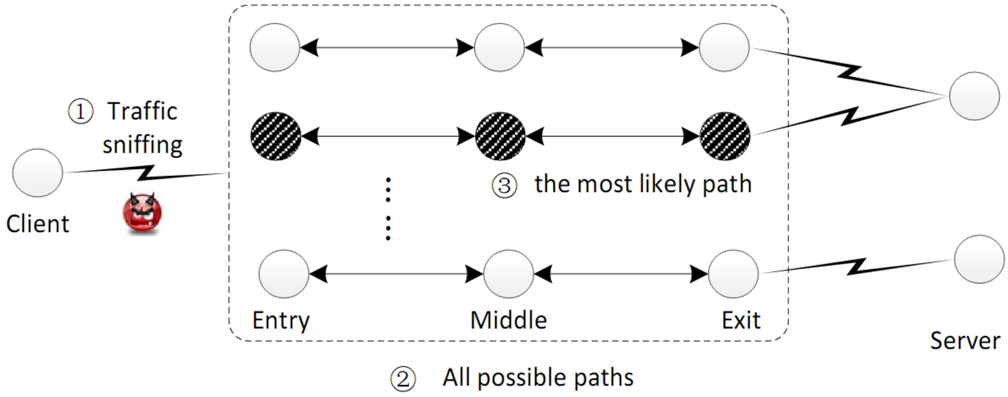


Figure 3. Path or Websites Inferring Attacks

Due to the limitations of the above attacks, attacks of type C base on traffic confirmation, as illustrated in Figure 4. In these attacks, attackers firstly compromise the entering and exiting segments of user circuit, and then use traffic analysis to identify both parties of anonymous communications. The accuracy of such de-anonymization attacks can be calculated by $P_{comp} \times P_{conf}$, where P_{comp} is the probability that attackers can control or monitor both segments of entering and exit, P_{conf} is the accuracy of traffic analysis. As accuracy of traffic correlation becomes more and more acceptable [10-11], researchers are seeking methods to improve P_{comp} in an efficient way.

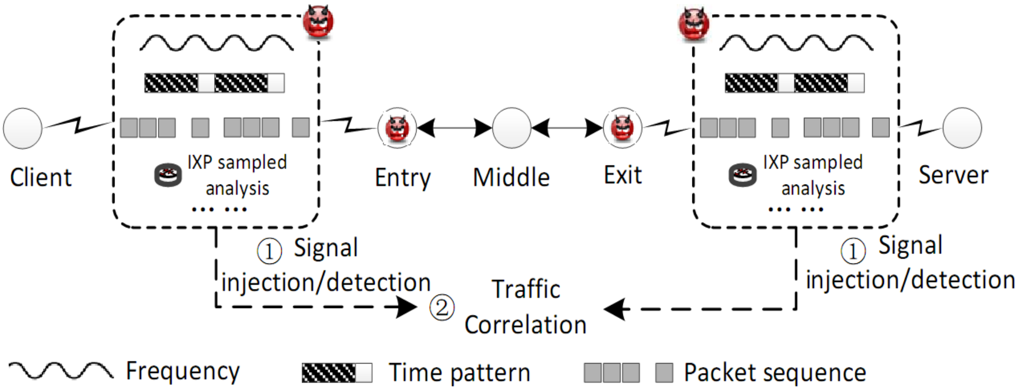


Figure 4. Traffic Confirmation Attacks

P_{comp} can further be split as $P_{comp} = P_{entering} \times P_{exiting}$, where $P_{entering}$ is the probability that attackers compromise user entering segment (***ep*** or guard nodes), while $P_{exiting}$ represents the probability that attackers compromise user exiting segment (exit nodes, ***exp*** or destination servers). As taking control of Tor nodes can access their detailed operation and logs which can help to improve the accuracy of traffic confirmation, it's more attractive to compromise guard and exit nodes than ***ep***, ***exp*** and servers. In Tor's original design, both entry nodes and exit nodes of circuits are selected randomly weighted by their bandwidths. However, the introduction of guard mechanism makes it different for entry and exit nodes. For every new circuit, Tor client will select a new exit node, thus $P_{exiting}$ can be improved by building more circuits. But the client will select just 1 of n guard nodes that he currently holds, and the n guard nodes will not change in 30-60 days as long as they are reachable. This change makes compromising a guard node more challenging and attractive, because once an attacker successfully compromises a user's guard node, then she will have a higher probability of compromising the user's circuits in the following 30-60 days. Some researches have been done to improve the security of guard nodes [14-15].

As analyzed above, P_{comp} is an important factor in deanonymizing Tor. In this paper, we propose a novel method to shorten the interval between two rounds of guard selection without disturbing user's normal usage, thus fasten the procedure of compromising entry guards.

Specifically, to mount an attack of type A, attackers should have the ability to manipulate traffic (*e.g.*, injecting malicious code) exiting from Tor, which requires to control an exit node, or part of ***exp***, or a destination server. While attack of type B needs attackers to passively observe Tor traffic at ***ep*** or a guard node. With wider availability, attack of type C needs attacker to manipulate/observe Tor traffic at both ends of a circuit. For example, in order to inject time signals at ***ep***, attackers have to actively control part of ***ep***, and in order to detect the injected signals, attackers also have to passively observe Tor traffic at exit node or part of ***exp***.

As Tor becomes popular, it attracts more and more attentions of adversaries who control part of the Internet, *e.g.*, local network administrator, ISP and national government. They can not only deploy malicious Tor nodes, but also actively manipulate users' connections between Tor client and entry guards. So in addition to the deployment of several Tor nodes, this paper also assumes that the attacker can actively control part of ***ep***, that is he can generate, modify, delete, or delay Tor traffic at ***ep***. According to [1], these assumptions are consistent with Tor's threat model.

3. Attacking Against Entry Guard Selection

3.1. Basis

To prevent certain profiling attacks, such as predecessor attack [16] and hidden service locating attack [17], entry guards are introduced into Tor. When entry guards are enabled, Tor maintains an ordered list of entry nodes, and stores this list persistently to disk. When choosing the first hop of a circuit, client chooses at random from the first n (default $n = 3$) usable guards on the list. Without entry guards, if an attacker controls C out N relays (ignoring bandwidth), then the attacker will control both the entry and exit nodes of any given circuit with probability $(C / N)^2$. We will finally be compromised as more and more circuits are built. While using entry guards, we would have probability $(N - C) / N$ to choose a good entry and not be compromised until the next round of entry guards. However, it's also obvious that if a client does pick a malicious guard she will have a higher probability of being compromised in the following 30-60 days.

It's necessary to note that, in order to recognize the malicious guards which intensionally fail or choke circuits that extend to non-colluding exit nodes, Tor measures both the path construction success rate and path usage success rate. However, it's obvious that these measures aim at finding malicious guards from the existing reachable guards which client has already selected and don't affect client's choice of a new guard. Thus it's possible to affect the update of a user's guard list without disturbing the user's normal communication.

3.2. Attack

The presented attack in this paper aims to compromise users' entry guards by shortening the time interval between two guard selection rounds. We assume the attacker has several guard nodes deployed in Tor network, and he can identify and manipulate Tor traffic on ***ep*** (*e.g.*, local ISP or network administrator). The assumptions are validsince Tor is operated in a volunteer manner to provide communication anonymity while not hiding the fact that someone is using Tor. The basic idea is as follows:

1. an attacker, with several guard nodes deployed, monitors Tor traffic of a particular user (called Alice). The attacker will get Alice's guard list after a short time, as the first hop of all Alice's Tor circuits is randomly selected from her guard list.
2. By taking advantage of Tor's path specification, the attacker keeps only one of Alice's guard nodes reachable by blocking the connections to her other guard nodes, thus increases the rounds of Alice's guard selection while not disturbing Alice's normal Tor usage.
3. The attacker continues this process until one of his guard nodes is selected by Alice. With compromising 1 of Alice's 3 guard nodes, first hops of 1/3 Alice's circuits are under the attacker's control.
4. With careful operation, the attacker can further compromise Alice's remaining 2 guard nodes, which makes all first hops of Alice's circuits under his control.

For simplicity, we focus on analyzing how to compromise one of Alice's guard nodes in the following parts. In order to improve efficiency of the above attack, two issues need to be concerned. First, the attacker should increase the probability of being selected during Alice's one guard selection round. Second, the attacker should make the rounds of Alice's guard selection (also the number of blocking) as little as possible. Corresponding analysis will be made in the following sections.

3.3. Analysis

3.3.1. Hit Probability in One Guard Selection Round: According to Tor's path selection algorithm [18-19], with deploying k malicious guard nodes, the probability that a guard node with bandwidth b indicates that hit probability in one guard selection round depends on both the number of injected guard nodes and the claimed bandwidth of each injected node. We evaluate guard bandwidth distribution in real Tor network and actual selected guard list by a Tor user (based on 2014-05-05-09-00-00-consensus). The results are shown in Figure 5 and Figure 6, where the ratio of guards which have a bandwidth more than 10MBps in Tor network is 8.96%, while the ratio of user selected guards which have a bandwidth more than 10MBps is 31.33%. The result proves that deploying more high bandwidth guard nodes can increase the hit probability in one guard selection round. But since there are more and more nodes in Tor network, the number of deployed guards should also be increased to maintain a desired hit probability, which makes the attack more and more expensive.

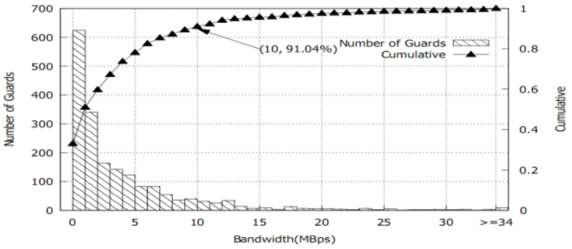


Figure 5. Distribution of Guard Bandwidth in Tor Network

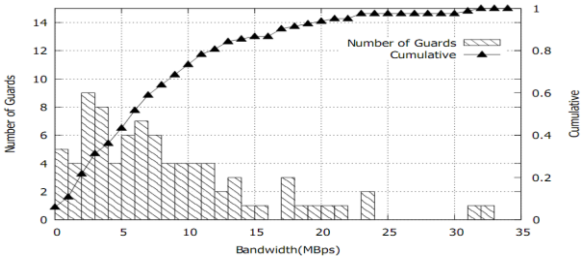


Figure 6. Distribution of Guard Bandwidth Selected by Tor User

With the introduction of entry guards, the round of guard selection is limited to once per 30-60 days. That is, if Alice doesn't select any of guard nodes under attacker's control as her guard nodes this time, Alice's next guard selection will be 30-60 days later. This heavily slow guard's compromise. The proposed attack in this paper aims to shorten the time interval between Alice's two guard selection rounds by blocking particular Tor connections, and thus accelerating the compromise of Alice's entry guards. The number of blocking, which is also the rounds of Alice's guard selection, needed for Alice to first select a guard node under our control in this attack will be analyzed following.

3.3.2. Expectation of Blocking: Denoting the number of normal and malicious guards in Tor network with N and M respectively, and their bandwidth is denoted by $B_1, B_2, \cdots, B_N, B_{N+1}, \cdots, B_{N+M}$, with a total bandwidth B. And S is used to denote the normal guards that have been selected before the first selection of a malicious guard. The probability of firstly selecting a malicious guard after $n-1$ tries can be given by

$$\begin{cases} P_1 = \dfrac{\sum_{N+1}^{N+M} B_i}{B} & n = 1, \\ P_n = \dfrac{\sum_{N+1}^{N+M} B_i}{B - \sum_{k=1}^{S} B_k} \cdot \prod_{i=1}^{n-1} (1 - P_i) & n > 1 \end{cases}$$

For simplicity, we assume bandwidths of all guard nodes are same, so Tor will select every guard with same probability. Then the problem above becomes to select some particular balls from a bucket of balls without putting back. In this situation, the probability of selecting a malicious guard for the first time at the nth guard selection round can be given as follows,

$$P_n = \frac{A_N^{n-1}}{A_{N+M}^{n-1}} \cdot \frac{M}{N + M - (n-1)}$$

where A is the permutation operation. Then the aimed expectation can be computed by

$$E = \sum_{n=1}^{N=1} n \cdot P_n$$

According to the statistics on 24 June 2014 [2], there are currently about 2000 guard nodes, that is $N + M = 2000$. In Tor network, the relationship between the number of deployed guards M and the expectation of the number of blocking E is illustrated in Figure 7. As it shows, when the number of deployed guards reaches 30, the expectation of blocking is about 60. Then the decrease of expectation is no more obvious even with more guard nodes deployed. Given the total number of guard nodes, even with a few deployed guard nodes, the proposed attack can still compromise entry guards of a given user in dozens of blocking.

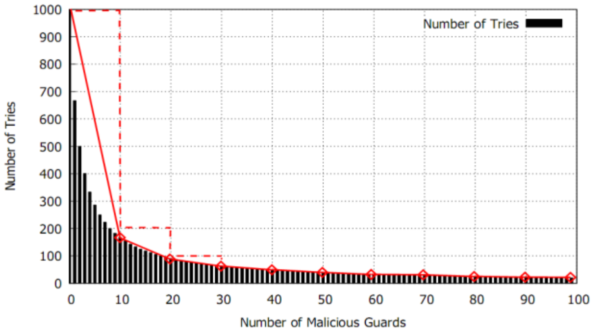


Figure 7. Relationship Between Number of Malicious Guards and Blocking Expectation

3.3.3. Actual Time Needed: As specified in Tor path specification [18], when running as a client, Tor tries to maintain at least a certain number of clean circuits, so that new streams can be handled quickly. Specifically, on startup Tor tries to maintain one clean fast exit circuit that allows connection with port 80, and at least two fast clean stable internal circuits in case we get a resolve request or hidden service request. Based on this, we make the following measures to evaluate the actual time interval between two guard selection rounds. We run a Tor client with no application data proxied, block all its current guard nodes, and wait until one of our guard is selected. In our 10 experiments, the mean interval between two selection rounds is about 1.5 minutes.

Besides, different user behaviors also affect the time interval between two selection rounds. For example, sending many streams over Tor leads to higher rate of circuit creation, which brings a quick selection of next new guard. In this case, we run an application with proxy data through Tor every minute, which causesthe building of a new circuit every minute. As a result, about 19 minutes are needed to complete 19 selections, which means the time interval is about 1 minute.

3.3.4. Attack Detectability and Impacts: Several measures are taken to harden the detectability of the proposed attack. First, as attackers always keep at least one guard reachable for the victim and no blocked entry guards are selected to build circuits, it's difficult for the victim to be aware of the ongoing attack. However, at the very beginning, if the attacker chooses to block a guard node which has already been used as the first hops of some circuits, the user would suffer some latency due to the circuits switch.

Second, aiming to force victims to quickly select malicious guard nodes from the set of all nodes labeled ***GUARD***, we can deploy proper guard nodes to enlarge the malicious subset. And we can also make the deployed guard nodes run as long-time as possible and configure them with high-bandwidth, so as to increase the probability of our guard nodes being selected, and thus accelerate the ongoing attack.

Besides, complex network environments make it difficult for Tor to detect the proposed attack, as it's hard to distinguish whether the failure is due to an attack or a network failure. The proposed guard selection attack may help to dramatically degrade the anonymity of Tor . This attack aims to defeat the security offered by entry guard mechanism, and further help correlate the two parties of an anonymous communication. Under such a successful attack, a Tor user will have all her circuits built with a compromised guard in the following 30-60 days. With more circuits being built, the user will finally select a compromised exit node controlled by the attacker, thus lose her communication anonymity.

4. Experimental Results

In this section, we evaluate the number of blocking needed for Alice to select one of our nodes as her guard node through experiments based on TorPS. In order to evaluate the efficiency of the proposed attack, we select 10 and 50 guard nodes from the consensus file (2014-05-05-09-00-00-consensus) respectively. And for each number, we select corresponding amount of guard nodes by random and high-bandwidth wayseparately. Then, we do guard selection repeatedly until 1 of the pre-selected 10 or 50 nodes is selected. We analyze the number of blocking needed for a given client to successfully select one of our "malicious" guard nodes as his entry guards for the first time. The results are shown in Figure 8 and Figure 9.

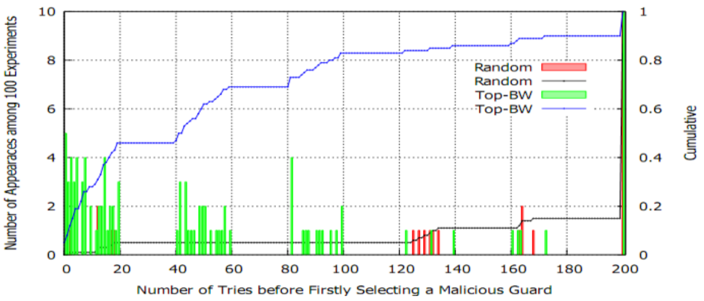


Figure 8. Distribution of Interruptions with 10 Malicious Guard Nodes

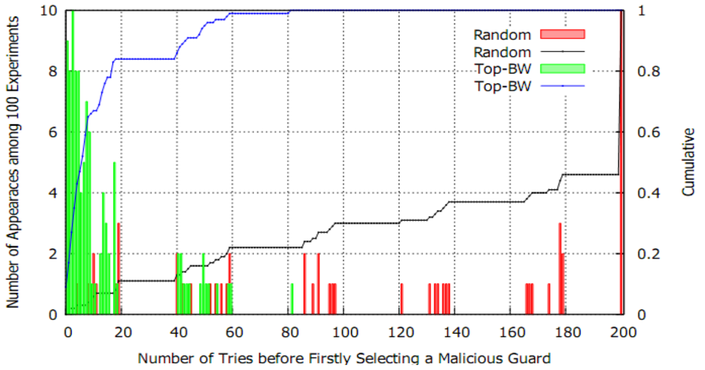


Figure 9. Distribution of interruptions with 50 malicious guard nodes

We limit the upper bound of interruptions to 200 in the experiments. From Figure 8 and Figure 9, we conclude that a moderate attacker who could observe and block traffic can compromise one guard node of a given user with dozens of blocking. Particularly, with 50 high-bandwidth malicious guards, the attacker can compromise more than 80% of given users with less than 20 blocking. According to section 3.3.3, the actual time completing 20 blocking is about 30 minutes.

5. Conclusions

The proposed attack can accelerate the compromise of specific user's entry guards without disturbing their normal Tor communication. But the large number of failures are still doubtful due to the blocking of guard connections. However, Tor can modify its source code to detect this anomaly (similar with path construction success rate and path usage success rate). Besides, Tor can also present these anomalies to users explicitly and indicate that Tor users are probably under attack. However, inspired by the finding about natural churn of guard nodes in [20], attackers may explore to make the guards blocking appear as guards' natural churn, which will make it more difficult for Tor to distinguish whether a guard node is down by itself or blocked. Obviously, the reduced detectability may result in a low efficiency of the proposed attack which needs to be researched deeply.

In this paper, we first analyze previous researches on deanonymizing Tor, and find that guard compromise has become a key issue in deanonymizing Tor. Then, considering an attacker who can generate, modify, delete, or delay Tor traffic at ***ep***, we introduce a novel attack against Tor to accelerate the compromise of guard nodes without disturbing normal Tor communication. Under the current design of Tor, attackers have a large probability to observe both ends of user circuit once the guard nodes are compromised, and can further break Tor's anonymity. Extensive theoretical analysis and experiments on TorPS are performed to validate the effectiveness and feasibility of the proposed attack.

Acknowledgements

This work is supported by the National Science Foundation of China (Grant No. 61133016), the Fundamental Research Funds for the Central Universities (Grant No. ZYGX2012J067), and the National High Technology Joint Research Program of China (863 Program, Grant No. 2011AA010706).

References

[1] R. Dingledine, N. Mathewson, and P. Syverson, Tor: The second generation onion router. DTIC Document, Tech. Rep., **(2004)**
[2] http://metrics.torproject.org
[3] T. G. Abbott, K. J. Lai, M. R. Lieberman, and E. C. Price. Browser based attacks on tor. Privacy Enhancing Technologies. **(2007)** pp.184-199.

[4] S. L. Blond, P. Manils, C. Abdelberi, M. A. D. Kaafar, C. Castelluccia, A. Legout, and W. Dabbous. "One bad apple spoils the bunch: exploiting p2p applications to trace and profile tor users", arXivpreprintarXiv: **(2011)** , pp.1103.1518.

[5] S. Zhioua. Tor traffic analysis using hidden markov models. Security and Communication Networks, Vol.6, No.9, **(2013)** pp.1075-1086.

[6] X. Cai, X. C. Zhang, B. Joshi, and R. Johnson. "Touching from a distance: Website fingerprinting attacks and defenses", Proceedings of the 2012 ACM conference on Computer and Communications Security. **(2012)** pp.605-616.

[7] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel. "Website fingerprinting in onion routing based anonymization networks", Proceedings of the 10th annual ACM workshop on privacy in the electronic society. **(2011)** pp.103-114.

[8] T. Wang and I. Goldberg. "Improved website fingerprinting on tor", Proceedings of the 12th ACM workshop on privacy in the electronic society, **(2013)** pp.201-212.

[9] M. Edman and P. Syverson. "As-awareness in tor path selection", Proceedings of the 16th ACM conference on Computer and communications security. **(2009)** pp.380-389.

[10] X. Fu, Z. Ling, J. Luo, W. Yu, W. Jia, and W. Zhao. "One cell is enough to break tors anonymity", Proc. Black Hat DC. **(2009)**

[11] Z. Ling, J. Luo, W. Yu, X. Fu, D. Xuan, and W. Jia. "A new cell counter based attack against tor", Proceedings of the 16th ACM conference on Computer and communications security. **(2009)** pp.578-589.

[12] P. Liu, J. Shi, L. Wang, X. Wang, and Q. Tan. "Ix-level adversaries on entry-and exit-transmission paths in tor network", IEEE Eighth International Conference on Networking, Architecture and Storage (NAS). **(2013)** pp.166-172.

[13] S. J. Murdoch and P. Zielinski. "Sampled traffic analysis by internet-exchange-level adversaries", Privacy Enhancing Technologies. **(2007)** pp.167-183.

[14] K. Bauer, D. McCoy, D. Grunwald, T. Kohno, and D. Sicker. "Low resource routing attacks against tor", Proceedings of the 2007 ACM workshop on Privacy in electronic society. **(2007)** pp.11-20.

[15] R. Jansen, F. Tschorsch, A. Johnson, and B. Scheuermann. "The sniper attack: Anonymously deanonymizing and disabling the tor network", Network and Distributed Systems Security Symposium (NDSS). **(2014)** SanDiego,CA,USA

[16] M. K. Wright, M. Adler, B. N. Levine, and C. Shields. "The predecessor attack: Ananalysis of a threat to anonymous communications systems", ACM Transactions on Information and System Security (TISSEC) **(2004)**, Vol.7, No.4, pp.489-522.

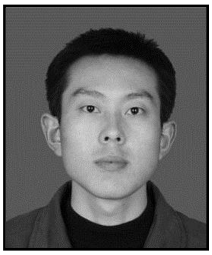
[17] L. Overlier and P. Syverson. "Locating hidden servers", IEEE Symposium on Security and Privacy. **(2006)** pp.15-pp.

[18] https://gitweb.torproject.org/torspec.git/blob/HEAD:/path-spec.txt

[19] Z. Ling, J. Luo, W. Yu, M. Yang, and X. Fu. "Extensive analysis and large-scale empiricale valuation of tor bridge discovery", INFOCOM. **(2012)** pp.2381-2389.

[20] T. Elahi, K. Bauer, M. AlSabah, R. Dingledine, and I. Goldberg. "Changing of the guards: A framework for understanding and improving entry guard selection in tor", Proceedings of the 2012 ACM workshop on Privacy in the electronic society. **(2012)** pp.43-54.

Authors



Quangang Li, he has been working toward Ph.D. at the School of Computer Science & Engineering, University of Electronic Science and Technology of China. His research interests include pattern recognition, social network and machine learning.

Peipeng Liu, he received his BS degree in Software Engineering from Shandong University, China in June 2009. And he is currently working towards his PhD degree in Information Security at Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. His current research interest mainly includes anonymous communication.

Zhiguang Qin, he received Ph.D. degree from University of Electronic Science and Technology of China in 1996. Now, he is a professor and Ph. D. supervisor in the School of Computer Science & Engineering, University of Electronic Science and Technology of China. His research interests mainly include information security and complex network.