An Anonymity Vulnerability in Tor - A Summary

Damian Franco
Dept. of Computer Science
University of New Mexico
dfranco24@unm.edu

***Abstract*—Privacy is and will continue to be one of the most concerning issues in cyberspace. Tor, short for "The Onion Router," is free and open-source software for enabling anonymous communication. This paper discusses cybersecurity issues and how Tor is vulnerable to end-to-end traffic correlation attacks. The authors present a set of Trapper Attacks that can deanonymize user activities in a Tor network, which can be used by both Autonomous System-level (AS-level) and Node-level adversaries. Autonomous System-level adversaries can exploit censored networks and control entry guards, while Node-level adversaries can exploit poor reliability to compromise existing nodes and the anonymous path. The authors implement a tool that can launch these attacks to automatically reveal communication relationships between a Tor user and its destinations, and present a formal analysis framework to evaluate Tor's security vulnerability. The proposed Trapper Attacks in this paper are designed to scale up in real-world Tor networks and can deanonymize Tor clients in real-time with a 100% accuracy rate and a false positive rate close to 0.**

***Index Terms*—Tor, anonymity, cybersecurity, Networks, deanonymization, infiltration, cyber attacks, honey relay**

I. Introduction

Today, we have the most access and expansion of all things technology. Emergence of technologies such as Internet-of-Things, cyber physical systems and virtual reality seem to be continuing the drastic change of everyday life. Along with those technologies come with its own risks and challenges and most of those challenges involve security and privacy issues. User data is becoming increasingly accessible to individuals who have ill-intent and personal information can be leaked or compromised. Privacy is important and has led to the development of anonymous browsing. Tor or "The Onion Router" is a free and open-source software that allows users to browse the internet anonymously by routing their traffic through a network of volunteer-operated servers. Tor has become increasingly popular for protecting their communication anonymity. Tor and other anonymous communication techniques have good intentions in design, but are heavily exploited for many criminal activities.

The deanonymization of the Tor network involves identifying the communication relationship between a user and their destination. Previous research has proposed attacks on Tor through controlling guard relays or correlating network communications, but these attacks assume that an adversary can monitor traffic on the forward or reverse path, and do not account for practical scenarios, such as the increasing scale and geographical distribution of Tor's nodes across different countries. To protect against deanonymization attacks, Tor uses entry guards as the first-hop router in its core network, where each client maintains a fixed list of guard relays and chooses one as the first hop for new circuits, and the guard list remains unchanged for 120 days as long as all guards are reachable, making it less likely for a compromised relay node to be selected. The number of guard relays that an adversary can control is the key factor to controlling Tor deanonymization attacks. This leads to the introduction of a "Honey" relay. A "Honey" relay refers to a relay node that has been intentionally compromised by an adversary in order to monitor or manipulate Tor traffic. The term "honey" implies that the node is used to attract and deceive unsuspecting Tor clients, as opposed to being a legitimate node in the Tor network.

Tor assumes that adversaries can manipulate some onion routers and monitor network traffic, but it is difficult for them to observe all relays on a path. However, adversaries can inject fake onion routers into paths intended for censored networks, affecting the reliability of Tor circuits. By injecting honey relays, the Trapper Attack can compromise circuits and be misperceived as poor communication reliability, and combined with other attacks. The goal of the research is to investigate the practicality of Trapper Attacks on a live Tor network, test its effectiveness by implementing it with honey relays, and improve efficiency and accuracy over existing attacks.

The study investigates ways to alter Tor's guard selection algorithm to force a user's traffic towards honey relays that are injected into the Tor network, by proposing techniques to select nodes that are known to the adversary and deny users' connections to trustworthy onion routers. The proposed Trapper Attacks aim to modify Tor's path selection algorithm by injecting honey relays, forcing Tor users' traffic to move towards them, allowing the adversary to inspect and correlate the traffic to achieve deanonymization by identifying each traffic entering or leaving the honey relays. Trapper Attacks can accurately and quickly deanonymize Tor users with a 100% accuracy rate and almost 0 false positive rate within 40 seconds, as well as increase the likelihood of compromising Tor users by up to 90% within 4 hours by deploying honey relays. The Advanced Trapper Attacks, when combined with Tor's relay selection algorithm, can shorten the time interval a client updates its guard relays from 3-6 months to 3 minutes, greatly accelerating the process of compromising the guard-relays-list of a targeted client.

This paper investigates the Trapper Attacks on the Tor network, comprehensively studies its efficiency, accuracy, and

feasibility, and validates it with experiments on a live Tor network, with the main contributions being presented in the following sections.

(i) Introduce Trapper Attacks which are a set of new attacks that can give an adversary the power to control Tor's routing path, leading to a shift from probabilistic to deterministic node selection for a Tor user.
(ii) Test their survivability against Sybilhunter and DannerDetector.
(iii) Proof-of-concept of an implementation of Trapper Attacks that results in communication deanonymization.
(iv) Large-scale simulations and real-world experiments on live Tor networks, and the evaluation results are presented.

A. Relevant Work

Various existing methods of deanonymization attacks have been proposed against Tor. These include Path Selection Attacks which aim to help Tor clients avoid an adversary to monitor any direction of the traffic at both endpoints of the communication and Traffic Analysis Attacks which make use of metadata to correlate the flows of the communicating end points.

II. Methods: Threat Model

The Tor network, which is designed for anonymous communication, is vulnerable to adversaries that can monitor network traffic entering and exiting the Tor communication channel. The adversaries in this model are assumed to be organizations such as ISPs that can control Tor relays as honey relays and monitor all targeted Tor user's incoming and outgoing connections. It is assumed that the underlying network protocols are secure, but the types and amounts of honey routers that an adversary can control to launch a deanonymizing attack are limited.

There is an assumption that adversaries can observe, alter, or drop a communication connection in the Tor network. Examples of such adversaries include China, which has blocked access to known entry nodes in Tor, allowing it to selectively block entry nodes and force targeted user traffic through its entry nodes. The paragraph notes that routing-capable adversaries can control the routing path between two endpoints of the communication, enabling them to launch various end-to-end traffic correlation attacks. The paragraph ends by stating that the discussion will consider the types and amounts of adversary resources under a set of assumptions.

Importance of an adversary that controls two endpoints of Tor users' connections for end-to-end traffic correlating attacks is not to be underestimated. The assumption is made that the adversary has bandwidth and computing resources, and the node-level adversary functions can run their own Tor routers or collude with some Tor routers. The network-level adversary function is assumed to control one or several ASes and to monitor, alter, or drop any traffic entering or exiting the Tor network. In this scenario, a user's entry nodes become an attractive target for the Trapper Attack if the network-level adversary function is interested in investigating who is accessing specific destinations. In certain circumstances, an attacker can block Tor connections by observing all TCP connections between the Tor client and their entry guards. They achieve this by blocking the user's entry nodes, thereby forcing the user to select an adversarial relay. This process is repeated until an adversarial relay is finally chosen. The result is a selective Denial-of-Service attack that provides the adversary with improved circuit visibility, significantly enhancing the effectiveness of the proposed attack.

III. Methods: The Proposed Trapper Attacks

This section introduces a set of Trapper Attacks against the Tor network that can compromise users' circuits by forcing them to choose injected honey relays as guards and exit relays. To aid in understanding the attack methodology, the basic Trapper Attacks are described first, which can be mounted by both Node-level adversaries and AS-level adversaries. Next, a much safer variant that further protects honey relays from being detected by existing detection algorithms is described. Finally, the effectiveness of the proposed Trapper Attacks is evaluated in detail.

First, the authors consider two representative experimental scenarios for the Trapper Attack. The assumption is made that the attacker has several honey relays with high-bandwidth and high-uptimes deployed in the Tor network. Adversaries can launch honey relays to selectively affect the reliability of Tor nodes, significantly enhancing the adversary's visibility to deanonymize the Tor network. Furthermore, Trapper Attacks can be combined with AS-level adversaries, which are ASes or organizations that have the cooperation of ASes. The details of Trapper Attacks on the Tor network are presented below.

A. Basic Attacks

An ISP can use Tor's relay selection algorithms to manipulate the routing path of Tor by diverting from the actual guard relays. By hijacking Tor's relay selection, the traffic is automatically routed to the honey relays. This type of attack enables the adversary to control all responsible entry guards of a specific Tor client for a certain period. It is notable that the authors have observed that when a guard relay is unavailable, the entry guards are likely to be updated in Tor's virtual circuit construction. In this scenario, AS-level adversaries can exploit Tor's relay selection algorithms to force a Tor client to choose a honey relay as a guard relay for all their circuits. This attack will result in all Tor client traffic being redirected to the honey guards, which will be visible to the adversary. The GRC attack can be executed in the following steps:

(i) Identify the entry guards of a Tor user.
(ii) Test their survivability against Sybilhunter and DannerDetector.
(iii) Repeat both steps above until the adversarial relay is selected as an entry guard.

The attack persists until a new entry guard node is included in the guard list. The adversary blocks all previous guard

relays, while ensuring that the malicious guard relay is operational, to gain control over all entry guards of the Tor client's circuits used for communication. AS-level adversaries can execute such routing path attacks to manipulate Tor's traffic. Previous research has demonstrated that these attacks may not always succeed. As a result, the GRC attack seems harmless to Tor users, making it highly durable without detection.

In order to conduct a successful deanonymization attack, an adversary must be able to monitor Tor users' traffic at both the entry and exit points of the communication channel. To achieve this, the authors use the Tor Circuit Selective Destruction (TCSD) attack, which forces a Tor client's entry and exit relays onto honey relays. This is done by destroying the circuit and repeating the process until a honey relay is selected as the exit node. The attack enables the adversary to control both the entry and exit guards of the communication and launch an end-to-end traffic correlation attack. The adversary can then monitor a significant amount of targeted client traffic, such as obtaining information about the internet destination and the Tor client's IP address. By forcefully destroying the circuit, the attacker can speed up the reconstruction process, significantly increasing their chances of gaining control of an exit node promptly.

B. Advanced Attacks With Anti-Attack-Detection Mechanisms

The authors discuss more advanced attacks with anti-attack-detection mechanisms. They discuss policies for preventing detection and improving the survival of the basic Trapper Attacks proposed by hiding the injected honey relays. To prevent selective denial-of-service attacks, some studies have been able to detect malicious relays in the Tor network. In addition to the methods represented in Sybilhunter, a deterministic approach has been proposed by DannerDetector. Both approaches are well-known existing approaches to this problem. Sybilhunter is a system that detects Sybil relays by analyzing their appearance and behavior, such as their configuration and uptime sequences. The more honey relays that are injected into the Tor network, the easier it is for Sybilhunter to detect them. On the other hand, DannerDetector aims to detect selective denial-of-service (DoS) relays in the Tor network. It can identify all honey relays within a few rounds of execution. In each round, the detector establishes multiple testing circuits, according to its policy, through every relay to download a large file. If the circuit is interrupted during the file download, the detector would suspect the existence of honey relays in the circuit. Nevertheless, the authors have identified an anti-detection policy that can bypass the aforementioned detection methods. The anti-detection policy consists of three tactics:

(i) Obfuscating Deployment
(ii) Deferral Decision
(iii) Collaborative Filtering

The obfuscating deployment tactic aims to bypass Sybilhunter by varying the configuration of the Tor instance. The deferral decision and collaborative filtering tactics are proposed to conceal the honey relays from DannerDetector.

In order to simplify the management of multiple honey relays, detection operators often administer them simultaneously. For example, they may reboot all of the honey relays at the same time or use the same configuration file for each honey relay. A method to bypass the Sybilhunter is proposed in which the method can effectively obscure the appearance and uptime patterns of the honey relays in the Tor network. The main idea of obfuscating deployment is to randomly introduce honey relays into the Tor network while maintaining the attribute distribution of the network.

To detect honey relays in the Tor network, DannerDetector conducts multiple testing rounds to improve its algorithm's accuracy. In each round, it selects two relays from the consensus as the guard and exit nodes and rotates the rest of the nodes as middle nodes to construct the testing circuit. For each circuit, a large file is downloaded through it. Honey relays are likely to interrupt the uncompromised circuit, causing the download to fail, while normal relays perform well during the download. Hence, DannerDetector can identify honey relays by analyzing the download result. This led to the creation of a Collaborative Detection (CD) algorithm to protect honey relays from being detected by DannerDetector. In short, the algorithm has instances of honey relays work together and dynamically change the value of their parameters to bypass the testing circuit, using a history pattern list and circuit fingerprints. There are three scenarios in which practical applications occur when using this algorithm. The first is when both the guard and exit nodes are honey nodes, the next is when both the guard and exit nodes are both normal nodes and last is when one of the guard and exit nodes is a honey node.

C. Deanonymization in Tor

Lastly, there are multiple approaches to deanonymization in Tor and how the attacks would function on a real Tor network. The Trapper Attacks experiment aims to demonstrate how an adversary can selectively affect the reliability of Tor nodes, including guard and exit relays, using honey routers deployed in Tor network, and potentially in cooperation with AS-level adversaries. In other words, existing traffic correlation analysis only considers scenarios where an adversary can monitor traffic in both directions between Tor clients and Internet destinations, but the paths between the client-to-entry may cross censored networks, making it difficult for adversaries to monitor both paths, and the use of fixed entry guards by Tor clients also decreases adversary visibility. The system architecture for deanonymizing Tor network consists of a circuit controller module, a fast traffic analysis module, and a traffic correlation module, where the circuit controller selectively affects Tor circuits to increase visibility of honey relays, the fast traffic analysis extracts protocol features of circuits to create Tor users' feature vectors, and the traffic correlation correlates traffic entering or leaving honey relays to find user-destination pairs.

The Circuit Controller in a Tor network uses control and relay cells to communicate between nodes and users, with

control cells issuing commands and relay cells carrying end-to-end TCP stream data, while the Tor control protocol enables communication between the controller and the locally running Tor instance. The Circuit Controller tests for a Pearson correlation coefficient, and if it is below a certain threshold. It then sends an interrupt command to a honey relay to interrupt the circuit in one of two ways.

The fast traffic analysis techniques used by an adversary involve monitoring high-level protocol features and translating raw cells into pattern vectors to determine whether two circuits share a common path. Lastly, to address the difficulty of correctly correlating the flows of Tor users, the traffic correlation module utilizes a clustering algorithm to group together Tor users who exhibit similar traffic patterns, thereby reducing the number of potential user-destination pairs to be correlated.

The approach the authors use uses a two-stage correlation to improve accuracy: the first stage flags potential origin-destination pairs based on Tor circuit construction and fingerprints, and the second stage uses throughput fingerprinting to compute the number of cells per unit time and relies on the Pearson Correlation Coefficient to identify matching flows for correlation analysis. The Pearson Correlation Coefficient (1) is a parametric measure of the linear correlation between two variables X and Y, and it is widely used to assess the degree of linear dependence between two random variables.

$$p = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2(y_i - \overline{y})^2}} \quad (1)$$

The approach involves calculating the Pearson's correlation coefficient between the pattern vectors of circuit-based fingerprinting features for each pair of observed Tor clients and adding the IP address pair to the final origin-destination pair if the correlation coefficient exceeds a threshold value of t (0.7 is suggested as the best threshold value), which helps recognize the communication relationship between Tor users and servers.

IV. Evaluation and Results

The proposed Trapper Attacks were evaluated through a series of experiments to assess their cost and the survivability of the honey relays.

A. Effectiveness of Trapper Attacks

To evaluate the effectiveness of Trapper Attacks, the authors modified the source code of Tor-0.4.0.5 to log circuit information, conducted experiments at 10 vantage points with 20 Tor clients each, and forced them to choose honey relays as entry nodes using Linux's iptables (iptables is a command-line firewall utility that uses policy chains to allow or block traffic) rules to simulate relay selection hijacking, recording the guard and exit nodes selected by clients on each circuit with 10, 20, 50, 100, 150, and 200 guards and exit nodes sampled from Tor consensus as controlled honey relays. The catch probability in the theoretical analysis is based on the assumption of m honey relays injected into the Tor network, with different tags assigned by directory servers, including guard, exit, both guard and exit, and neither guard nor exit routers, denoted as BG, BE, BGE, and BNGE respectively, and the bandwidth advertised by adversarial guards as b, which allows the calculation of the probability of Tor clients selecting adversarial Tor routers as guards.

The authors analyze the time it takes for adversaries to deny services to trustworthy onion routers and force Tor clients to select honey relays. It is assumed that adversaries have a budget for bandwidth and can run a certain percentage of decoy routers. The probability that a honey relay is selected as the guard after n updates for choosing the entry guard by a Tor client trying to create a three-hop circuit can be calculated, given the total bandwidth of Tor guard routers (B).

To compromise Tor's routing path, colluding the first and last router in a three-hop circuit is required, leading to the probability of honey routers being chosen as both entry guard and exit routers for a three-hop circuit, which can be approximately derived. Based on the theoretical analysis, the authors obtained the probability that a Tor client chooses the guards and exits.

The impact of the number of attacker-controlled guard nodes and their bandwidth ratio on the probability of compromising the first guard node is shown in Figure 1 and 2, while the impact of the number of attacker-controlled exit nodes on the number of circuit rotations required to compromise the first exit node is shown in Figure 3. Figure 4 shows the probabilities of successfully compromising both the guard and exit nodes, and the number of circuit recreations required to achieve this, revealing that an adversary can compromise the routing path of a Tor client with a 90% probability after blocking 400 circuit creation attempts when 100 guards and 20 exits are controlled, which can lead to deanonymization within 4 hours against roughly 100% of users in locations under the adversary's control.

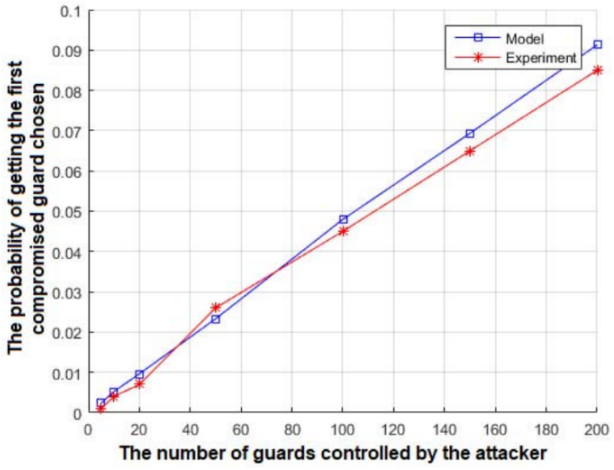

Fig. 1. How the probability of selecting the first compromised node as a guard node is affected by the number of guards under control of the attacker.

B. Advanced Trapper Attacks Survivabliity

Next, the authors tested and evaluated results for the ad-

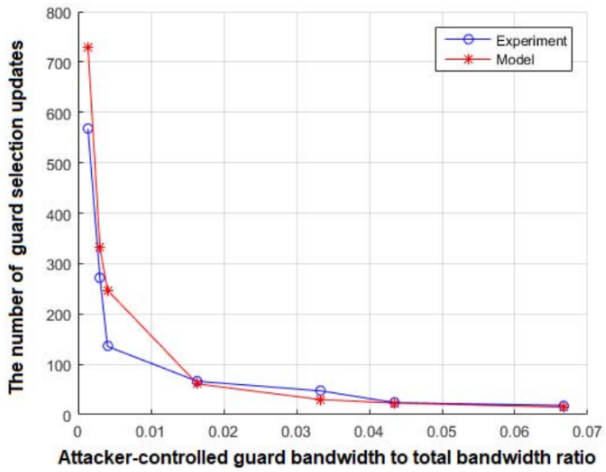

Fig. 2. How the ratio of attacker-controlled guard bandwidth affects the number of updates necessary to select the first compromised guard.

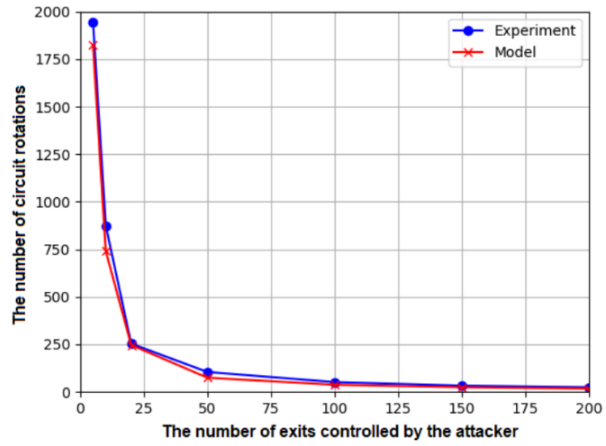

Fig. 3. How the number of exits controlled by the attacker affects the amount of circuit rotations, i.e., destruction and reconstruction of circuits, needed to select the first compromised node as an exit.

vanced Trapper Attacks survivability such as the Sybilhunter and DannerDetctector. They conducted experiments to evaluate the effectiveness of the anti-detection policy against the Advanced Trapper Attack under the current detection policy. The simulation implementation is used to compare the performance of the attack under Sybilhunter and DannerDetector detections in this section. Initially, the network churn rate of the joining or leaving relays is calculated using a formula from [6] on three data sets. The two figure that I did not include on this write up show the network churn rates during ten days in August 2020. The study found that without applying the obfuscating method, there was an unexpectedly high churn rate on 2020-08-20, which indicates that many relays joined or left the Tor network. Additionally, there was no significant

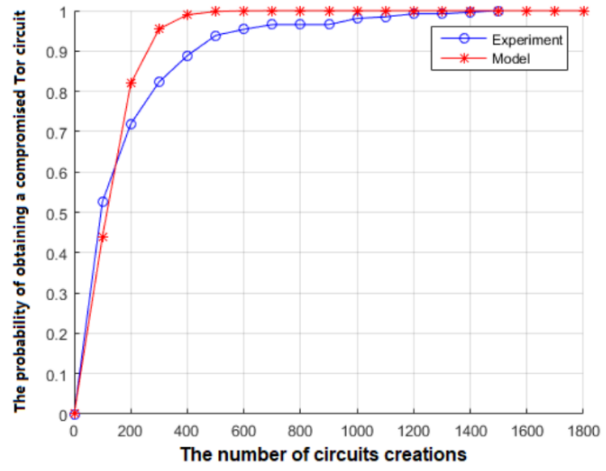

Fig. 4. The probability of successfully obtaining a compromised Tor circuit, including a guard and an exit node, while varying the number of circuit creations, given that 100 guard nodes and 20 exit nodes are under the control of attackers.

difference in the network churn rate between the first and third data sets, indicating that the injected honey relays did not significantly affect the network churn rate.

By running Sybilhunter on origin and honey datasets, the effectiveness of Trapper Attacks was evaluated before and after applying the obfuscating method. Results showed that with the proposed obfuscating method, 209 honey relays per month could be injected on average, with a survival rate of 94.28%. Without the obfuscating method, the survival rate was only 5.58%. Under DannerDetector, three kinds of honey relays with different behavior parameters were tested, and the lifetime of honey relays with deferral decision and collaborative filtering was found to be about 85 times longer than honey relays with random interrupt policy and about 235 times longer than honey relays that interrupt each uncompromised circuit. These experiments demonstrate that the anti-detection policy significantly increases the anti-detection ability of the injected honey relays under both Sybilhunter and DannerDetector.

C. Accuracy of Deanonymization Attacks

Lastly, evaluation on the accuracy of deanonymization attacks was conducted in this research. This was done by simulating three common types of Tor network users at seven vantage points on the live network. These included a web browsing client, an IRC client, and a bulk transfer client, each with their own distinct functionality. They deployed honey relays by setting up 3 guard and 5 exit nodes and directing Tor to use specific nodes for certain destinations on all three types of communication on the Tor network. To collect data, a modification to the source code of Tor-0.4.0.5 to log information about Tor circuits and built a vector of unique circuit identifiers was added. The authors then calculated the total lifetime and active time of each circuit and removed outliers using the interquartile range.

In conclusion, the analysis of the results is presented in Figure 7 and Figure 8, which demonstrate that the source IP and destination IP of a user can be accurately detected within 30 seconds with a false positive rate of less than 1%.

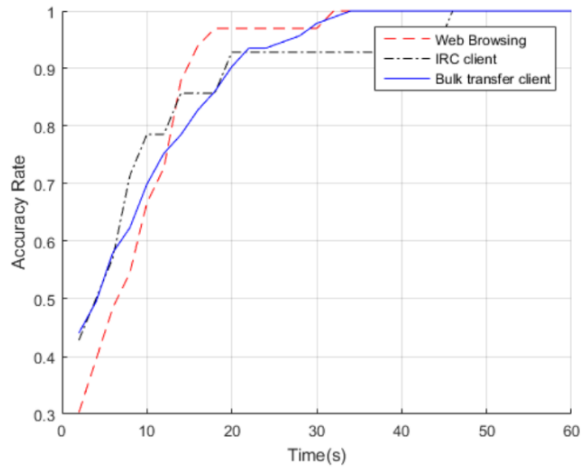

Fig. 5. The accuracy of the deanonymization attack over time.

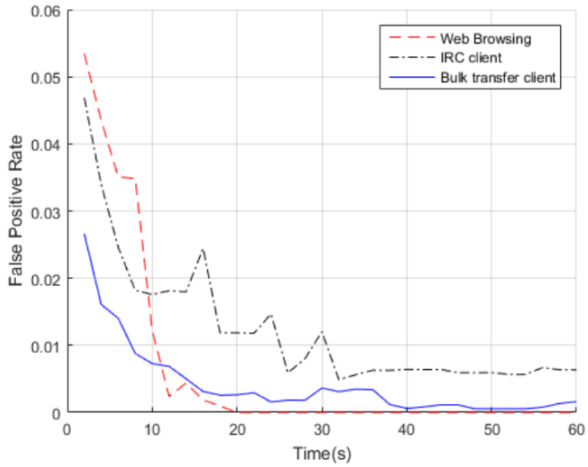

Fig. 6. The false positive rate of the deanonymization attack over time.

V. Discussion and Conclusions

This research demonstrates the impact of Trapper Attacks on the anonymity of other systems and suggests potential countermeasures after demonstrating their threat against the Tor network. Low-latency anonymous communication systems like Tor are vulnerable to end-to-end traffic analysis attacks, but the probability of a user being chosen by an attacker-controlled entry and exit node is low, and the guard mechanism was designed to mitigate various deanonymizing attacks. Trapper Attacks degrade Tor's path selection algorithm from probabilistic to deterministic, posing a privacy and security risk to Tor users by allowing adversaries to selectively control a Tor's routing path if a client has chosen a compromised guard. Trapper Attacks could also impact other low-latency anonymous communication systems, such as I2P and Freenet, if AS-level adversaries control a fraction of relay nodes, compromising the client's entry and exit nodes, and launching end-to-end traffic analysis attacks, and thus, the severity and security implications on these networks need to be assessed.

There are two possible countermeasures to make Tor path selection more robust that the authors offer. The first is to limit the number of exposed guard nodes. The Trapper Attack exploits the weakness of Tor's path selection algorithm by directing traffic to honey relays controlled by adversaries. To prevent this attack, the number of exposed guard nodes should be limited, and circumvention tools such as Tor over VPN should be used to prevent attackers from seeing Tor traffic. Two ways to minimize the chance of attacker-controlled guards becoming a Tor user's entry guard are limiting the number of updated guard nodes and using circumvention tools. The second countermeasures is to consistently monitor the behavior of Tor relays. If manipulated Tor circuits are detected by monitoring, the effectiveness of an attack is significantly decreased. A system reputation system that monitors the behavior of each Tor router will increase the chances of detecting the presence of an unreliable router or present attacker.

In this paper, Trapper Attacks are introduced, which enable adversaries to identify anonymous communication over Tor accurately. Through experiments, the attacks are shown to allow adversaries to control the routing path by selectively affecting the reliability of Tor circuits or hijacking Tor's relay selection, thereby deanonymizing user communications. Trapper Attacks deny service on trustworthy onion routers to redirect users' data towards adversary-controlled honey relays, resulting in a significant increase in the adversary's knowledge. The feasibility, survivability, and effectiveness of the attacks are demonstrated through an experiment on a live Tor network, which shows that the attacks can deanonymize the network in near real-time with a high survival rate even in the presence of honey relay detection software. The paper also presents a formal analysis framework to quantitatively assess the severity of the attacks and their security implications on the live Tor network.

References

[1] Qingfeng Tan, Xuebin Wang, Wei Shi, Jian Tang, Zhihong Tian, "An Anonymity Vulnerability in Tor", IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 30, NO. 6, DECEMBER 2022

[2] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in Proc. 13th Conf. USENIX Secur. Symp., Berkeley, CA, USA, vol. 13, 2004, p. 21.

[3] T. T. Project. (Jun. 2020). Tor Metrics Portal. [Online]. Available: https://metrics.torproject.org/

[4] T. Elahi, K. Bauer, M. AlSabah, R. Dingledine, and I. Goldberg, "Changing of the guards: A framework for understanding and improving entry guard selection in Tor," in Proc. ACM Workshop Privacy Electron. Soc., 2012, pp. 43–54.

[5] K. Kohls, K. Jansen, D. Rupprecht, T. Holz, and C. Pöpper, "On the challenges of geographical avoidance for Tor," in Proc. 26th Symp. Netw. Distrib. Syst. Secur. (NDSS), Feb. 2019.