Poster: Application-Layer Routing Attacks on Tor

Katharina Kohls
Ruhr University Bochum
Bochum, Germany
katharina.kohls@rub.de

Christina Pöpper
New York University Abu Dhabi
Abu Dhabi, United Arab Emirates
chrstina.poepper@nyu.edu

***Abstract*—Application layer routing attacks allow to force Tor entry traffic through an area under adversarial control. In contrast to previous routing attacks that focus on layer-three or -four mechanisms, the application-layer attack exploits Tor's DoS mitigation features that were implemented to prevent from bursty connection and circuit establishments. The proposed poster documents the general attack concept and summarizes the results of preliminary experiments. Furthermore, it raises questions about several open challenges and provides an overview of the next steps of this work in progress.**

***Index Terms*—Tor, Routing Attacks, Application Layer**

I. Introduction

Successful attacks against Tor affect more than 2 million daily users [1] and enable adversaries to de-anonymize connections or critical services of the network. One important class of attacks in this context are traffic analysis attacks, which benefit from the metadata side channel of low-latency transmissions. Different passive [2], [3] and active [4]–[6] traffic analysis attacks help to identify accessed websites [7], [8] or match the endpoints of a connection.

There are two main influencing factors for the success of a traffic analysis attack. First, the technical capabilities of the adversary define the ability to detect similarities in monitored network transmissions. Second, the organizational capabilities define the number of nodes that can be accessed for recording traffic, i. e., they influence the probability of analyzing related streams. Recent work demonstrates that an AS-level adversary has access to approximately $40\,\%$ of nodes in the Tor infrastructure [9], which increases to up to $85\,\%$ coverage for nation-state adversaries [10].

Routing attacks improve the organizational capabilities of the adversary by directing traffic through areas under adversarial control. Examples of this are attacks on the Border Gateway Protocol (BGP) [10]–[12] or manipulations of Tor routing features [13].

Prior work on routing attacks most often assumes an AS-level adversary that can access transmissions up to the transport or network layer of the protocol stack.

While prior work focuses on transport- or network-layer attacks, we exploit the Denial of Service (DoS) mitigation features on the application layer [14]. In the following, we introduce the general attack concept, overview the results of preliminary experiments, and discuss several challenges of routing attacks on the application layer. Finally, we conclude the current status of this work in progress.

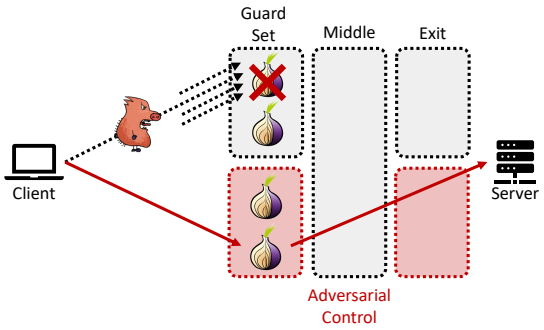

Fig. 1. General attack concept. The adversary triggers the DoS mitigation features of an entry relay to provoke the blocking of connection attempts. Consequently, the client moves to alternative relays from the guard set and forces traffic through an area under adversarial control.

II. Attack

A. General Attack Procedure

Figure 1 provides an overview of the general attack procedure. On startup, the user's Tor client establishes several circuits for maintenance and the transmission of data. In case of standard data circuits, these consist of three relays including one entry guard, one middle relay, and one exit relay picked from the current consensus. The selection of entry guards follows a fixed guard set selection procedure.

The adversary can block the connection to a specific entry guard by triggering one or more DoS mitigation features in the relay, respectively. By sending repeated connection attempts to the relay, the maximum number of parallel connections or requests is exceeded and the client's IP address is blocked. After triggering the DoS mechanism in a relay, the client cannot establish a circuit anymore and needs to move to another node of the guard set. If the new guard relay runs in an area under adversarial control, this allows monitoring the entry traffic of a connection for a website fingerprinting attack. In cases where also the exit traffic traverses an adversarial area, end-to-end confirmation attacks become possible.

B. DoS Mitigation Features

Out of overall ten configuration options for DoS mitigation features, we focus on four specific options handling the number of created circuits and connections, as documented in Listing 1. The `circuit` options cover the circuit creation rate of a single client IP address and refuse new circuits, if the `DoSCircuitCreationRate` is exceeded,

n concurrent connections exist, and the creation burst rate is violated. In the case of such a violation, new circuits are refused for a defined amount of time. Furthermore, the relay keeps track of the number of concurrent connections (`DoSConnectionMaxConcurrentCount`) established and reacts with closing new connections in case the threshold was exceeded.

Listing 1. Denial of Service Mitigation Options

```
DoSCircuitCreationMinConnections NUM
DoSCircuitCreationRate NUM
DoSCircuitCreationBurst NUM
DoSConnectionMaxConcurrentCount NUM
```

C. Attacker Model

We assume an adversary with the ability to establish multiple Tor connections between clients and Tor entry relays. These connections impersonate the user towards the entry relay and trigger DoS mitigation features through repeated connection and circuit creation requests. One option to achieve this behavior is the use of a TCP Man-in-the-Middle (MitM) that uses the client's IP address towards the entry relay.

The attack is time-sensitive in a sense that client addresses are blocked for a limited time in which the redirection must take place. Therefore, targeted attacks that force a client into using a specific entry relay require the synchronization of multiple MitM attacks.

III. Preliminary Experiments

In our preliminary experiments, we test how to trigger the `circuit` and `connection` parameters and the resulting behavior. To verify the triggered behavior, we use the debug and info logs of Tor and compare the occurrences of reject or closing messages. To this end, we use a minimal private Tor setup consisting of a client and server, two directory authorities, and four relays, two of which receive a manual guard flag and two are configured as exit relays. Furthermore, one of the guard relays is configured with the above DoS mitigation features.

First, we focus on creating a critical number of concurrent connections by running multiple Tor instances in parallel. Each Tor instance establishes multiple circuits on startup, incorporating the guard relay with DoS mitigation features. With these instances running in parallel, we find the `DoSConnectionMaxConcurrentCount` threshold to be exceeded, resulting in closed connections (cf. Listing 2). Using the Tor control port, we continue by creating multiple new circuits in all of the running Tor instances, which eventually triggers the `DoSCircuitCreation` features of the guard.

Listing 2. DoS Features Triggered

```
DoS mitigation since startup:
0 circuits killed with too many cells.
5535 circuits rejected, 2 marked addresses.
3390 connections closed.
```

IV. Challenges and Next Steps

While the results of our preliminary experiments indicate the desired behavior, different challenges need to be considered in the next steps of the project. One challenge addresses the establishment of a TCP Man-in-the-Middle, which allows to trigger the DoS mitigation features through multiple concurrent connections and repeated circuit establishments. Furthermore, we need to take the characteristics of the guard set generation into account and match them with the average coverage of an AS-level or nation-state adversary.

In the next steps of our work, we continue the preliminary experiments to fully understand the behavior in case of triggered DoS mitigation features. The insights of these experiments serve as a starting point to design different experimental setups that enable us to analyze the impact of a routing attack on the application layer.

Acknowledgment

This work was supported by Intel as part of ICRI-CARS.

References

[1] The Tor Project, "Tor Metrics," Jan. 2019, https://metrics.torproject.org.

[2] A. Kwon, M. AlSabah, D. Lazar, M. Dacier, and S. Devadas, "Circuit Fingerprinting Attacks: Passive Deanonymization of Tor Hidden Services," in *USENIX Security Symposium*, USENIX '15. Washington, DC, USA: USENIX Association, Aug. 2015.

[3] M. Nasr, A. Bahramali, and A. Houmansadr, "DeepCorr: Strong Flow Correlation Attacks on Tor Using Deep Learning," in *ACM Conference on Computer and Communications Security*, CCS '18. ACM, 2018, pp. 1962–1976.

[4] F. Rezaei and A. Houmansadr, "Tagit: Tagging Network Flows Using Blind Fingerprints," PoPETS '17, vol. 2017, no. 4. De Gruyter, 2017, pp. 290–307.

[5] A. Houmansadr and N. Borisov, "The need for Flow Fingerprints to Link Correlated Network Flows," in *Privacy Enhancing Technologies Symposium*, PETS '13. Bloomington, IN, USA: Springer, Jul. 2013, pp. 205–224.

[6] A. Biryukov, I. Pustogarov, and R.-P. Weinmann, "Trawling for Tor Hidden Services: Detection, Measurement, Deanonymization," in *IEEE Symposium on Security and Privacy*, SP '13. San Francisco, CA, USA: IEEE, May 2013, pp. 80–94.

[7] A. Panchenko, F. Lanze, A. Zinnen, M. Henze, J. Pennekamp, K. Wehrle, and T. Engel, "Website Fingerprinting at Internet Scale," in *Network and Distributed System Security Symposium*, NDSS '16. San Diego, CA, USA: The Internet Society, Feb. 2018.

[8] V. Rimmer, D. Preuveneers, M. Juarez, T. Van Goethem, and W. Joosen, "Automated Website Fingerprinting through Deep Learning," in *Network and Distributed System Security Symposium*, NDSS '18. San Diego, CA, USA: The Internet Society, Feb. 2018.

[9] A. Shah, R. Fontugne, and C. Papadopoulos, "Towards Characterizing International Routing Detours," in *Asian Internet Engineering Conference*, AINTEC '16. Bangkok, Thailand: ACM, Nov. 2016, pp. 17–24.

[10] R. Nithyanand, O. Starov, A. Zair, P. Gill, and M. Schapira, "Measuring and Mitigating AS-level Adversaries Against Tor," in *Network and Distributed System Security Symposium*, NDSS '16. San Diego, CA, USA: The Internet Society, Feb. 2016.

[11] Y. Sun, A. Edmundson, L. Vanbever, O. Li, J. Rexford, M. Chiang, and P. Mittal, "RAPTOR: Routing Attacks on Privacy in Tor," in *USENIX Security Symposium*, USENIX '16. Washington, DC, USA: USENIX Association, Aug. 2016, pp. 271–286.

[12] H. Ballani, P. Francis, and X. Zhang, "A Study of Prefix Hijacking and Interception in the Internet," Aug. 2007.

[13] K. Bauer, D. McCoy, D. Grunwald, T. Kohno, and D. Sicker, "Low-Resource Routing Attacks Against Tor," in *Workshop on Privacy in the Electronic Society*, WPES '07. Alexandria, VA, USA: ACM, Oct. 2007, pp. 11–20.

[14] dgoulet, "Request for denial of service mitigation subsystem," Mar. 2019, https://trac.torproject.org/projects/tor/ticket/24902#no2.