Exploring the Threats Posed by Botnets on Onion-Routing

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Justin Fabrizio

Spring, 2022.

Technical Project Team Members

Justin Fabrizio

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Daniel Graham, Department of Computer Science

Abstract
Tor Browser, the most popular internet anonymity system to date, relies on a network of Tor nodes to provide layered encryption for traffic. The security of Tor browser relies on the fact that a malicious party does not control both the entry and exit nodes on any given path through its network, as doing so would allow for traffic correlation analysis to deanonymize users communicating on that path. Likewise, botnets pose a serious risk to anonymity systems like Tor Browser that use onion-routing, since Tor Browser's protocol is very lenient in allowing machines to voluntarily operate as nodes in the Tor network. Making more discriminatory modifications to Tor Browser's path selection protocol such as incorporating a relay's publish date could help better mitigate such an attack.

1. Introduction

With over 92% of American households owning a computer and 85% with broadband internet access, data privacy has never been a greater concern to western society [3]. While internet data collection can be financially beneficial to private corporations, it has also sparked fears of large-scale government surveillance. These fears were magnified by a 2013 NSA leak of a project codenamed "PRISM" which indicated that the federal government had partnered with major tech companies to conduct data surveillance on the general public [4].

One of the most effective ways to preserve data privacy on the internet is through the use of internet anonymity systems such as Tor Browser. While Tor Browser has remained a viable internet anonymity system for the better part of two decades, constantly evolving technology will always present new threats to its ability to provide internet anonymity. This paper will explore the unique threats posed to Tor Browser by large-scale botnets and propose possible actions Tor Browser can take to mitigate such threats.

2. Background

Tor Browser provides anonymity to users by constructing a path of relays in their network through which all of the user's traffic is routed prior to reaching the destination address. Tor browser then uses AES encryption at the application layer in addition to "layered" TLS/SSLv3 encryption between relays to ensure that no single relay has access to both the user's IP address and their unencrypted traffic. In general, a path consists of three distinct relays; the first relay (closest to the client) is known as "entry" or "guard" node, the second is the "middle node," and the final relay is referred to as the "exit node." As such, the entry node has access to the user's IP address and other forms of identifying information, and the exit node has access to the user's unencrypted traffic [5]. A commonly theorized attack on this system is a traffic correlation analysis attack in which an attacker could deanonymize users by correlating attributes of the traffic, such as timestamps and download sizes, between the entrance and exit relays of the of the network [2]. However, this attack is generally not feasible due to the number of distinct Tor relays in the network, which generally fluctuates between six and eight thousand. Additionally, Tor Browser's path selection protocol imposes constraints on what relays can be selected for a given path to help lower the risk of one or more compromised relays being placed on the same path.

A botnet is a group of "zombie" computers that have been compromised through the use of malware and can be controlled by a single entity. Botnets can range from a couple dozen to upwards of a million individual machines. One of the most prominent botnets discovered was the

Mirai botnet which was estimated to contain over 587,000 machines [6].

3. Related Works

A widely-researched threat to Tor Browser and onion-routing systems as a whole is the distributed denial of service attack (DDoS). This attack is conducted by having numerous machines repeatedly attempt to connect to a targeted service or node and prevent legitimate traffic from being processed. This concept is closely related to this research as botnets are frequently used for such attacks due to their large size and distributed nature. A proposed solution to this particular attack is detailed in *Mitigating Distributed Denial of Service Attacks in an Anonymous Routing Environment: Client Puzzles and Tor* [1]. Fraser suggests having all clients attempting to make a connection through the Tor network first solve a puzzle to eliminate automated connections. The problem addressed in this paper is similar to the one being addressed in this research in the sense that the problem stems from automated actions on a large number of distributed machines; yet the solution proposed by Fraser loses relevance when the increased presence of "CAPTCHA farms" since the paper has been published is considered.

A second paper of relevance to this research is *Correlation Attacks on Tor* [2]. The author details the execution of traffic correlation analysis attack over the Tor network and demonstrates its success by capturing packet traces from both the client-side and server-side of a connection. It is important to note that this attack would not be possible if the attacker did not have access to the packet traces between the client and the entry node as well as between the exit node and the server. This topic lays the groundwork for this research as it provides motivation for an attacker to gain control over the vast majority of Tor relays.

4. Proposed Design

The current Tor path construction protocol imposes the following constraints on what nodes can be chosen as relays for a given path through the network: 1) the same node is never selected more than once in the same path; 2) multiple nodes from the same family are never placed in the same path; 3) multiple nodes from the same subnet are never placed in the same path; and 4) non-running/non-valid nodes are never selected. A "family" of nodes is defined as a group of nodes that list each other in the family field of their descriptors; this means that nodes must self-identify as belonging to the same family. The size of modern botnets (100,000+ machines) poses a risk to Tor's path construction protocol as it can be inferred that the vast majority of member machines within a botnet are not on the same subnet and, if running Tor nodes, they would not identify as being in the same family. Combined with the relatively small size of the Tor network (6,000-8,000 machines) it is reasonable to assume if a large botnet were to host Tor nodes on all its member machines, it would not be irregular for two machines from the same botnet to act as the entry and exit relays for a given path. Considering this scenario, even adding more nodes to the path would be futile in increasing security as an attacker would still only need to gain access to the entry and exit nodes of the path.

The proposed attack would be to combine the malware techniques behind botnet creation with the distributed, open-source, nature of Tor Browser to effectively gain access to a large number packet traces which could later be used to deanonymize Tor users through traffic correlation analysis.

A potential solution for this problem would be to modify the Tor path construction protocol such that it incorporates the existing "published" field in each node's descriptor that stores the date and time at which the node was activated. From there a threshold could be set such that the protocol

specifies a minimum difference between the creation time of the entry and exit nodes for them to be placed in the same path. Building on this concept, a more aggressive mitigation technique could be to set large time difference minimum in the protocol which could then be lowered incrementally on a timescale of minutes if no suitable nodes were found.

5. Anticipated Outcomes

It is anticipated that the first steps in the deanonymizing attack described above would go smoothly as the distributed nature of onion-routing relies on the software being open source, accessible, and volunteer-driven. However, one must also consider the amount of data that would have to be processed to go through with a traffic correlation analysis. In addition to automating the packet collection process, software would also have to be written to compare each exit node capture with every entry node capture on a regular basis to effectively and reliably deanonymize users. This process would have a high computational overhead, yet would still be more feasible than attempting to crack the layered TLS encryption used over the Tor network.

On the other hand, the solution described in the previous section would not have a significant impact on a client's connection time as it only adds a single additional decision to the path construction protocol. Moreover, this change only affects the path construction prior to connection, so the speed of connection would be indistinguishable from the old protocol after the connection with the server has been established. Despite this, the change would still only act as a mitigation technique as it would only prevent botnets that activated Tor nodes on member machines in a short amount of time. Overall, this small change could be effective in mitigating the threats posed by botnets or other large entities.

6. Conclusions

By nature, internet anonymity systems are particularly vulnerable to attacks from botnets as any unique identifiers that could be saved to filter malicious machines would hinder the anonymity of genuine clients and nodes. More specifically, the proposed attack would be for a large botnet of ideally 20,000+ machines to host Tor nodes. This would provide a suitable base for a typical traffic correlation analysis attack as the malicious nodes would account for the vast majority of nodes in the Tor network. Ideally, a mitigation technique or solution in response to this attack should only rely on the existing fields in a Tor node's descriptor, as the addition of new fields would require a high level of scrutiny to ensure that they do not lower the anonymity of clients.

A proposed mitigation technique to this attack is to modify Tor such that the path construction protocol has an additional path constraint related to the "published" field in a node's descriptor. More specifically, setting a minimum time difference threshold between the current node's and potential next node's "published" fields would help prevent multiple nodes hosted by the same entity from being put in a given path.

7. Future Work

In addition to broader exploration of other mitigation techniques, future work will focus heavily on the feasibility of the proposed Tor modifications regarding the "published" field and the path construction protocol. Due to the open-source nature of Tor, testing of the modifications could be performed with multiple machines on a private subnet to ensure that the changes do not have a significant effect on the speed of the path construction protocol as well as round-trip time once the connection is established. Additionally, it will be necessary to perform a more in-depth analysis of how the proposed mitigation technique could potentially be addressed by malicious entities attempting to perform the described attack.

References

[1] Nicholas A. Fraser. 2006. *Mitigating Distributed Denial of Service Attacks in an Anonymous Routing Environment: Client Puzzles and Tor*. Air Force Institute of Technology, Wright-Patterson AFB, OH.

[2] Jan Fajfer. 2018. *Correlation Attacks on Tor*. Czech Technical University, Prague, Czech Republic.

[3] United States Census Bureau. 2018. *Computer and internet use in the United States: 2018.*

[4] Adam Florek. 2014. *The Problems with PRISM: How A Modern Definition of Privacy Necessarily Protects Privacy Interests in Digital Communications*. J. Marshall J. Info. Tech. & Privacy Law. University of Illinois Chicago, Chicago, IL.

[5] Tor Project. 2021. *The Tor Project: Privacy & Freedom Online*.

[6] Manos Antonakakis. 2017. *Understanding the Mirai Botnet*. Georgia Institute of Technology, Atlanta, GA.

[7] Yoshimichi Nakatsuka, Ercan Ozturk, Andrew Paverd, Gene Tsudik. 2021. *CACTI: Captcha Avoidance via Client-side TEE Integration*. University of California Irvine, Irvine, CA.