POLITECHNICA UNIVERSITY OF BUCHAREST

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

COMPUTER SCIENCE DEPARTMENT

|  |  |
| --- | --- |
| upb.png | cs.png |

Review on

The Effect of DNS on Tor’s Anonymity

|  |  |
| --- | --- |
| Thesis supervisor:  Prof. Mihai Chiroiu | Gabriel Bădilă |

BUCHAREST

2018

Table of Contents

1. Introduction 4
2. Understanding the attacks 4
3. 1 Fingerprinting attacks 4
4. 2 Correlation attacks (DefecTor attacks) 4
5. 3 Different types of a DNS resolver 5
6. Defending against attacks 5
7. 1 Short-term solutions 5
8. 2 Long-term solutions 5
9. Similar articles 5
10. 1 Touching from a Distance: Website Fingerprinting Attacks and Defenses 5
11. 2 Website Fingerprinting in Onion Routing Based Anonymization Networks 6
12. 3 Remove global client-side DNS caching 7
13. Conclusion 7

A lot of research has gone into improving the Tor network, but its use of DNS has received little attention. An attacker can use DNS requests to mount highly precise website fingerprinting attacks. Results show that DNS requests from Tor exit relays traverse numerous autonomous systems that subsequent web traffic does not traverse. A set of exit relays, at times can compromise 40% of Tor’s exit bandwidth, uses Google’s public DNS servers — an alarming high number for a single organizations.

1. Introduction

Until now we have not learn how to build an anonymity network that can resist to global adversaries, provide low latency and scale well. The Tor browser did not have to considerate any of these global adversaries because his scheme was not designed to be able to identify and defend from adversaries that can observe the traffic for extended periods of time and in multiple network locations.

DNS traffic is highly relevant for correlation attacks because it traverses so many paths and autonomous systems (ASes) and if an attacker, by any chance, observe occasional DNS requests he may still be able to link both ends of the communication, identify the path through the network and alter or steal the information.

The Tor network is an overlay network that anonymizes TCP streams such as web traffic. Information are transmitted through three relays randomly selected — called guard, middle, and exit — that form a virtual tunnel called a circuit. The guard relay learns client IP address, through middle relay the information are sent and the exit relay send all the DNS requests to identify the address of the destination.

1. Understanding the attacks
2. 1 Fingerprinting attacks

The Tor browser always encrypts the traffic from the client to the exit relay so that the Internet Service Provider (ISP) cannot read the content of the packets. Even if ISP cannot see the information sent, he received statistics about the network traffic, about the frequency and the interval of the packets that are sent or received. An attacker can analyze the information stored by the Internet Service Provider and determinate the destination of the packets sent by the user, this procedure is called *website fingerprinting*.

This type of attack divides in two settings: a *closed world* (the attacker chooses a set of n monitored websites, with the restriction that the user is visiting only one of them, and tries to find which of the websites the user is visiting) and an *open world* (besides the n monitored websites, the user can browse to other sites that are not monitored by the attacker. There are two metrics in the open world setting: recall and precision. Recall represents the probability that a visit on a monitored website will be detected and precision the probability that a classification of a visit on a monitored website is the correct one).

1. 2 Correlation attacks (DefecTor attacks)

After tests have been performed the conclusion was that if an attacker succeeds to intercept the DNS requests made on exit relay and analyze the statistics retained by the ISP he can take control of the both endpoints. More than that, if he uses the tools, TorPS (Tor Path Simulator) and AS Path Prediction he will have access to all information sent by the client and he can manipulated in any way he wants. The DNS requests represent a great help for an unknown instance to identify and use the client traffic. We can say that when precision has great value, DefecTor attacks pose an even bigger threat that website fingerprinting attacks.

1. 3 Different types of a DNS resolver

All the DNS requests are managed by a DNS resolver. An exit relay can either run its own resolver or rely on a third-party resolver (like the one provided by the ISP). If an exit relay uses his own resolver the attacker wait for a DNS request from the exit relay’s IP address, but if the exit relay uses a third-party resolver the DNS request will come from an IP address unknown to the attacker. Here is the question: use your own resolver that has a greater chance to be hacked or use a third-party resolver that has a lower change to be hacked, but shared your all information with that third-party software? The tests show that the first option for users was a third-party resolver (Google, 40%) and the second option was the local resolver (12 %).

1. Defending against attacks
2. 1 Short-term solutions

Exit relay operators must make a decision between using their own resolver and expose DNS queries to network adversaries and using a third-party resolver which exposes DNS queries to a third-party. The goal is to minimize the exposure and the resolvers that are the most used are Google, local resolver and the resolvers provided by their ISP. To choose the Google resolver, the Google will have access to all statistics what contradicts the main goal of Tor, anonymity. If we choose the option with a local resolver each DNS query will be exposed to a diverse set of ASes in the network and the attackers will have an easy mission to retrieve the queries. The last option will be to use a resolver provided by the ISP. The risk in this scenario come from the possibility of misconfiguration and censoring of DNS resolver that could turn the host in a very centralized data sinks.

Taking in consideration all the above, the best solution will be to choose between a local resolver and an ISP resolver, the Google solution being out of the question.

1. 2 Long-term solutions

It was proposed T-DNS which brings several TCP optimizations to transport the DNS protocol over TLS and TCP. TLS layer provides security and privacy between the exit relay and the resolver. Another option will be that websites operators that are concern for the safety of their users should offer an onion service as alternative.

DefecTor attacks are based on increasing precision of website fingerprinting attacks so the based the way to defend in this situation is to focus on reducing the recall of the website fingerprinting attacks.

1. Similar articles
2. 1 Touching from a Distance: Website Fingerprinting Attacks and Defenses

The current defensive systems have a lot of deficiencies and this article presents a set of tests for detecting these deficiencies and the steps taken to correct them. Results show that ad hoc defenses against website fingerprinting attacks fail in almost 90% of the cases.

The purpose of these tests is to show the vulnerability of user’s data, even though they use some of the most secure mechanisms for sending their information (IPSec tunnels, SSH tunnels, Tor browser). The results are based on different types of attacks coming from different angles: active attacks that require subverting nodes in the anonymity network, active attacks that require injecting traffic into the network, and attacks based on subverting web servers visited by anonymous users. The focus of these attacks is directed to quite different results: discovering the identity of the anonymous user, discovering the destination server or discover the path through the anonymizing network (all of these are possible because of the instability of the DNS traffic).

An attacker can recognize web pages pretty easily. Although there is inherent stability in the ordering requests, browsers cannot request an object until they have received the portion of the page that references it. Another issue is that web proxies, such as Tor and SSH multiplex the data traffic over a single, encrypted channel and an attacker can see the size, the timing and the direction of the packets. Moreover, Tor send all the information in 512-byte cells so the size of every packet is limited. If we take in consideration that every TCP message is usually followed by a confirmation message (ACK) then all traces look even more similar.  
  
 Like I said before it is not very hard to determine a page that a user is visiting, but an attacker wants to know if the user is visiting any of the pages on a banned web site. To find the answer to this question, the attacker can build a Hidden Markow Model (HMM) for each target web site and compute an algorithm that calculate the probability that a packet trace is made by a user that activates on a target web site. If the probability is below a certain threshold that the user certainly visiting that web site. Huge web sites have over a thousand pages so it would be very difficult to build an algorithm for this kind of sites. But a vulnerability of this sites is that most of them are build based on templates and an attacker could create a model based on those templates. Another idea is to watch the behavior of users, when they search something directly in the URL search bar, when they press the “Back” and “Forwards” buttons. Based on this behavior and using HMM an attacker can model page loads and determinate the visited web site.

These tests indicates that the probability to identify a page loaded over an SSH tunnel is greater than 90% and the probability to identify a page from Tor is around 80%. These results shows how defenseless are even the most secure methods for sending data against the website fingerprint attacks and the exploitation of the DNS requests.

1. 2 Website Fingerprinting in Onion Routing Based Anonymization Networks

Anonymous communication aims at hiding the information send between two instances in a network. Tor network claims that can offer anonymity by routing data through several overlay nodes and using layered encryption of the content. What this article aims to achieve is to check how effective Tor’s security is against website fingerprinting attacks (WFP). The website fingerprinting attacks are build base on the assumption that web objects can be differentiated by examining the TCP connection between the user and the proxy, which only holds true for HTTP/1.0.

The tests were performed on two different scenarios: closed-world and open-world. We want to test with which probability an attacker can retrieve a certain amount of relevant web pages using the anonymization network that he assumes his victim uses as well. After testing the closed-world scenario we achieved a 54.61% which are much higher than it was estimated initially. For the open-world scenario were used from a list of the 1.000.000 most popular Internet pages, 4000 uncensored URLs randomly selected (the censored pages contain real illegal websites: sexually explicit content, the most popular pages from the Alexa list and pages randomly chosen from Alexa Urls). The open-world scenario should have been more challenging then the closed-world scenario, but in this case alarming result were obtain — positive rate of 73%.

As countermeasures, padding is the most common technique which is employed to impede traffic analysis. The sender appends a certain amount of dummy data to the actual data before applying the encryption. Usually, padding is used to achieve a fixed packet size or a random one. Of course, the padding produces overhead and the user is the one that will set the overhead-anonymity ratio.

1. 3 Remove global client-side DNS caching

Even from the earliest versions, Tor has kept a client-side DNS cache. The are several security issues with DNS cache, a malicious exit note can affect the behavior of the circuit in the future. If the site’s DNS provides different answers to clients all over the world, then the client’s cached choice of IP will reveal where it first learned about the website which will be a great advantage for the attacker to track packets.

I agree that the DNS cache has a great purpose because using it, your network can speed up considerably. Why try to identify a website using a DNS resolver when you already visited that website and you saved it in a local database from you can access faster. A great danger to which users are exposed is *DNS cache poisoning*, an attacker can insert unauthorized domain names or IP addresses into DNS’s cache. Not only that the attacker can intercept the packets that user send, but he can redirect client requests to wrong destination, malicious websites that pose a great threat on user data’s security.

The solution will be that clients stop cache DNS results in the global address map and cache them in per-circuit DNS caches instead. But where is an upside there is also a downside, entries in the address map which relied on happening after the DNS cache entries can no longer work so well. This problem can be solved by introducing a few extra steps of address mapping after the DNS cache is applied. The performance will not be heavily affected, only the case where the exit node and its resolver do not have the answer cached will add a little strain on the performance.

In the initial article none of the solutions include removing DNS cached from the global address map, there the focus was on choosing a better resolver for DNS requests or adding another layer for security between the exit relay and the resolver because they thought that DNS cache it is very important to maintain the security-performance ratio.

1. Conclusion

The above exposure had the purpose to demonstrate how attacker can use DNS traffic from Tor relays to launch more effective website fingerprinting attacks. Also, we proposed defenses against these attacks, such as choosing a local resolver or an ISP resolver instead the Google one, T-DNS, padding or removing the global DNS cache.

bibliography

[1] B. Greschbach, T. Pulls, L. M. Roberts, P. Winter, N. Feamster, “The Effect of DNS on Tor’s Anonymity”, in NDSS, 2017. URL: http://wp.internetsociety.org/ndss/wp-content/uploads/sites/25/2017/09/ndss2017\_06B-2\_Greschbach\_paper.pdf

[2] X. Cai, X. C. Zhang, B. Joshi, and R. Johnson, “Touching from a distance: Website fingerprinting attacks and defenses”, in CCS. ACM, 2012. URL: https://nymity.ch/tor-dns/pdf/Cai2012a.pdf

[3] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel, “Website fingerprinting in onion routing based anonymization networks”, in WPES. ACM, 2011. URL: https://nymity.ch/tor-dns/pdf/Panchenko2011a.pdf

[4] N. Mathewson, “Remove global client-side DNS caching”, Jul. 2012. URL: https://gitweb.torproject.org/torspec.git/tree/proposals/205-local-dnscache.txt