University of Tennessee, Knoxville

Trace: Tennessee Research and Creative Exchange

Masters Theses

Graduate School

5-2018

The Effect of Modern Web Content and Caching on The Tor Onion Router

Joseph Parker Diamond
University of Tennessee, jdiamon3@vols.utk.edu

To the Graduate Council:

I am submitting herewith a thesis written by Joseph Parker Diamond entitled "The Effect of Modern Web Content and Caching on The Tor Onion Router." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Science.

Maxfield J. Schuchard, Major Professor

We have read this thesis and recommend its acceptance:

Mark E. Dean, Audris Mockus

Accepted for the Council:
Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

The Effect of Modern Web Content and Caching on The Tor Onion Router

A Thesis Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Joseph Parker Diamond

May 2018

For My Family and My Community

Because the right to privacy is universal

Acknowledgments

I would like to thank Dr. Maxfield Schuchard for his faith in my abilities and for his patience and instruction when things are not moving smoothly. I would like to thank the VolSec research lab for their assistance, both in and out of work. I would like to thank my research committee, Dr. Mark Dean and Dr. Audris Mockus, for listening to my work despite the last minute notice and preparations. Finally, I would like to thank UT and all of the generosity of its donors that have made college possible for me and many others.

Abstract

This work evaluates Tor users' risk of de-anonymization in the presence of a network-level adversary. We evaluate the likelihood that a Tor user, who is consuming modern web content, will be susceptible to a traffic analysis or watermarking attack. This work shows that the previously studied point-to-point model for Tor connections is not realistic and does not fully capture the risk of de-anonymization for Tor users. We show these results by measuring network paths along key parts of a Tor circuit. First, we measure the paths between the Tor exit relays and web resources requested when accessing the Alexa Top 1000 websites. Then, we use available and trusted traceroute data to approximate paths between Tor users and likely guard nodes. Then, the intersection of these paths at an autonomous system level is examined to determine if they share any elements. If the intersection of the paths is non-empty, then a Tor user making a request with those paths is susceptible to de-anonymization.

Results from weighted selection of Tor exit and guard relays indicate that a Tor user visiting a random Alexa Top 1000 website is susceptible to de-anonymization with 20% probability for almost half of the Alexa Top 1000. Multiple resources account for significant additional de-anonymization risk over the point-to-point model, and shorter network paths to content distribution nodes do not effectively compensate. Moreover, examining the intersection of paths to resources in the top-level domains of a website does not full eliminate the risk of de-anonymization under the AS-Aware Tor problem.

Table of Contents

List of Figures

Chapter 1

Introduction

Onion routing systems, specifically the Tor project, provide anonymity for end-users by means of a telescoping encryption system with VPN-like request passing. A Tor client will create a ”circuit” of three relays; each relay has an associated public key. The Tor client will encrypt a message with each relay’s key and address the encrypted message to the next hop relay in the circuit. The message is then sent to the first relay in the circuit, the **guard relay**, which decrypts it and forwards it to the next relay. This process continues until the message reaches the **exit relay**, which removes the last layer of encryption and sends the request contained in the message on behalf of the client. Each relay only knows the previous hop and next hop of the message – in this way the client and destination are un-linkable to any single hop on the message path [2].

An adversary that can observe both ends of the communication chain – the client-guard and exit-destination paths – can link the client and destination via packet-size analysis and traffic watermarking attacks [6]. Individuals can perform such attacks by controlling a the guard and exit relays in a client’s circuit while an autonomous system (AS) can de-anonymize users by observing traffic at any point along the two paths at the ends of the Tor communication. The ability of an AS on the client-guard and exit-destination path to de-anonymize Tor users is called the **AS-Aware Tor problem**, and it has been the subject of many studies and attack efforts [7].

Recent studies like [7], however, model Tor connections as point-to-point tunnels and use tools like **wget** to retrieve items from a single IP address through a Tor connection. In reality, though, modern Internet content, especially HTTP web pages, consist of resources from multiple sources that are retrieved by the client at the time of the request or upon page load. Moreover, modern web resources extensively request external resources after the initial request by embedding requests in JavaScript through AJAX or other technologies. Since these resources are heavily used and shared throughout the web, they are often hosted on content distribution node(s) (CDN(s)) to make them more readily available to end-users.

This work analyzes the privacy implications of the previously unexamined effects that CDNs and multi-domain web resources have on Tor connections with respect to the AS-Aware Tor problem. To do this, we construct a system to request web content from each Tor exit node, traceroute the AS-level path to the web resources for the Alexa Top 1000 from each exit, and determine the likelihood of the de-anonymization based on the likely paths to guard nodes as generated by TorPS [5]. To do this, we use Python's Stem library to construct Tor circuits through various exit nodes. We then request all Alexa Top 1000 websites, record the resolved IP addresses for each resource requested, and then approximate the AS-level path from each resolved IP to the exit node by tracerouting the IP addresses with a RIPE ATLAS probe in the same AS as the exit node. Likely guard nodes are selected for each exit node through TorPS, and the AS-level paths between guards and clients are constructed through CAIDA traceroutes over the preceding four months. The intersection of these AS-level paths are then calculated, showing the likelihood of an AS being on both the client-guard and exit-destination path.

The rest of the paper is organized as follows: Section 2 details how an AS can de-anonymize Tor users through various watermarking techniques and the state of the art in threat mitigation. Section 3 will describe how the relevant data was collected through Tor and how the effects of CDN and multi-domain web content are measured. Section 4.2 will discuss the measurement results and how these results are interpreted. Section 4 provides a model for the likelihood of de-anonymization of Tor users given the results of the previous section. Finally, related work is discussed in Section 5 and the contributions are summarized in Section 6.

Chapter 2

Background

2.1 Onion Routing and the Tor Project

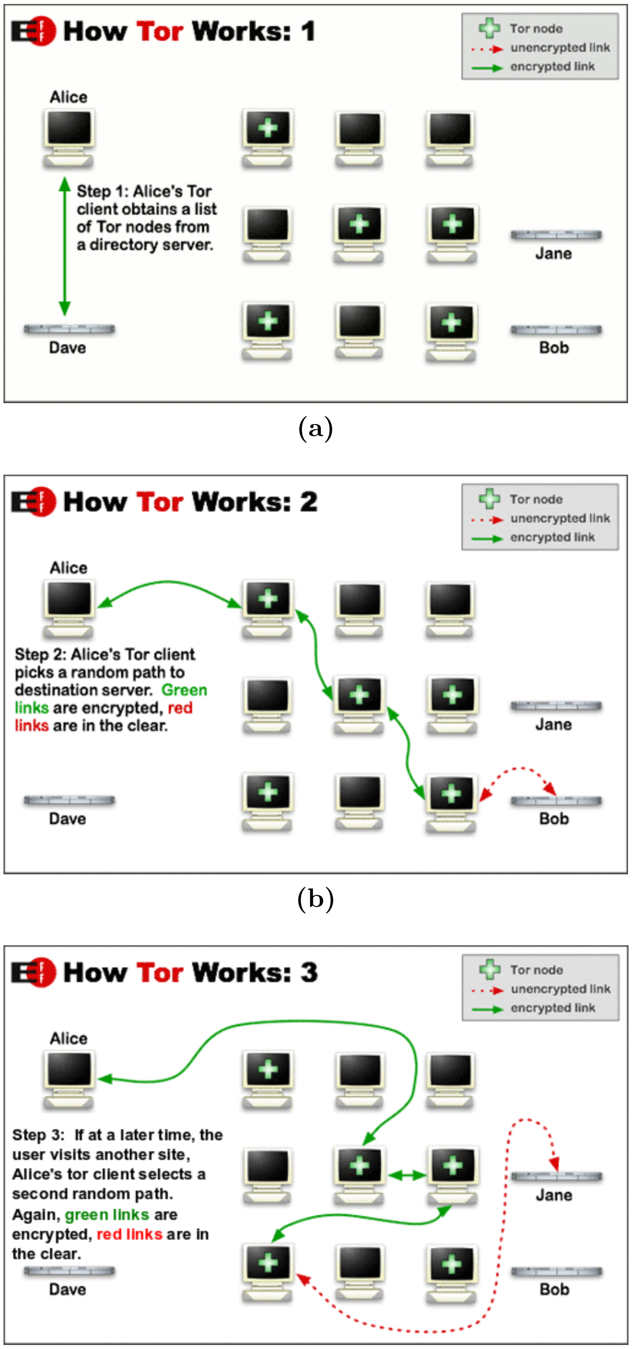
Onion routers are network systems that pass messages via multiple hops through multiple network hosts from a client to a destination. Typically, they also provide layered encryption and additional privacy-preserving features. The Tor Project is one of the most prolific onion routing systems, and is the system this work focuses on. Thus, when we refer to 'Tor', we mean the onion routing system proposed by [2] and implemented by The Tor Project.

Tor provides un-linkability between a client and a server via a multiple hop message passing system with layered encryption. This process is illustrated in Figure 2.1. The client forms a 'circuit' of 3 relays – a guard, middle, and exit – and obtains a public-key for each of the relays. The client then encrypts its message in layers, first with the public key of the exit relay, then with the public key of the middle relay, and finally again with the public key of the guard relay. The client then sends its triple-encrypted message to the guard node. The guard node decrypts the first layer of encryption on the message, but cannot remove the next layer of encryption since it does not know the middle relay's private key. After removing the first layer of encryption, though, the guard now sees that the message is destined for the middle relay selected by the client. The guard relay therefore forwards the message to the middle relay, which repeats this process until the message finally reaches the exit relay. The exit relay removes the final layer of encryption, revealing the plain text message. The exit relay then sends the message to the end destination and repeats the process in reverse back to the client. The server, however, only sees a message coming from the exit node and hence does not know that the client sent the message. Additionally, the exit relay only knows that the message came from the middle relay, so it too does not know the identity of the client. The only relay that knows the identity of the client is the guard node, but the guard node does not know the identity of the destination server. In this way, the server does not know the identity of the client, nor do the relays know the identity of both the client and destination. Therefore, this system provides client-destination un-linkability, or anonymity.

2.2 De-Anonymization via Watermarking and Flow Analysis

There were originally only a few theorized de-anonymization attacks on Tor. One obvious attack would be if an adversary controlled all of the relays in use by a given client. Then, the adversary would be able to decrypt all of the messages and find out who sent them by tracing the hops back from the exit relay. A less obvious attack involves an adversary that controls only the guard and exit node in use by a client, illustrated in Figure 2.2. In such a scenario, the adversary can de-anonymize the client via latency watermarking and traffic flow analysis.

In a latency watermarking attack, the adversary adds a unique pattern of brief delays between packets when the guard node receives packets from the client. The adversary then looks for a the same pattern of delays as the traffic is passed to the exit and out to the server. If the adversary sees this same latency pattern as traffic is forwarded from the exit to a server, then the adversary knows which client is attempting to talk to the server that is the recipient of the watermarked traffic. The adversary can also perform traffic flow analysis by counting the number of packets received by the guard node and sent by the exit node in a given time period. Over a long enough period of time the adversary can correlate the number of packets in particular traffic streams and likewise link a client and destination, successfully de-anonymizing the Tor user.


How Tor Works: 1

Step 1: Alice's Tor client obtains a list of Tor nodes from a directory server.

(a)

How Tor Works: 2

Step 2: Alice's Tor client picks a random path to destination server. Green links are encrypted, red links are in the clear.

(b)

How Tor Works: 3

Step 3: If at a later time, the user visits another site, Alice's tor client selects a second random path. Again, green links are encrypted, red links are in the clear.

(c)

Figure 2.1: How Tor Works from [4]

2.3 AS-Aware Tor Problem

The scenario where an adversary controls the guard and exit nodes is just a particular instance of a more generalized attack scenario. If any network-level entity ever simultaneously lies on the path between the client and guard and between the exit and server, then that entity can de-anonymize the client via the same watermarking attack described in 2.2. In terms of de-anonymization, controlling a part of the client-guard and exit-destination paths is equivalent to controlling a guard and/or exit relay, respectively. Since the network entities that transit traffic are typically internet service providers (ISP), or collections thereof, this attack is considered feasible only at the level of autonomous systems (AS). An AS is a collection of network routing entities, like ISPs or a singular ISP, that advertises routes to a collection of IP address prefixes that it maintains. The feasibility of ASes to de-anonymize Tor users if they control part of the client-guard and exit-destination routes is consequently referred to as the AS-Aware Tor problem; it is illustrated in Figure 2.3. This attack has been well studied by [7] and [5] and has been shown to be effective under a point-to-point connectivity model.

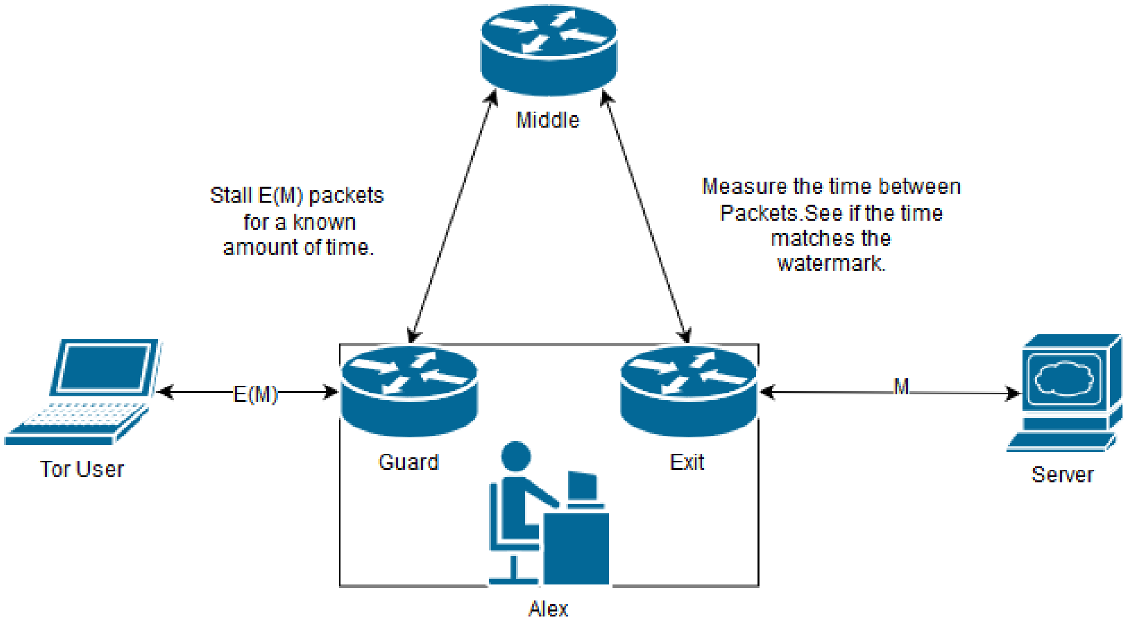

Middle
Stall E(M) packets for a known amount of time.
Measure the time between Packets.See if the time matches the watermark.
Tor User
E(M)
Guard
Exit
M
Server
Alex

Figure 2.2: Watermarking when an Adversary Controls Guard and Exit

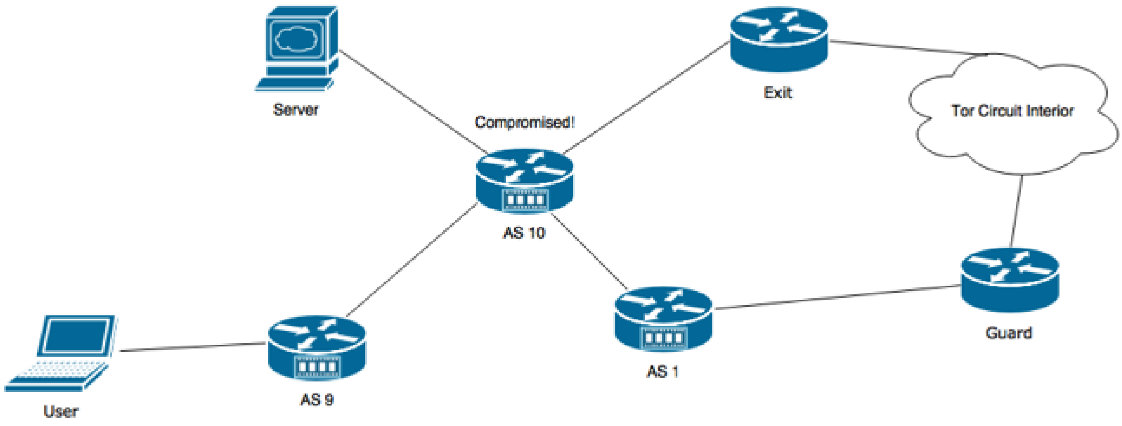


Figure 2.3: AS-Aware Tor Attack Diagram

Chapter 3

System Design

3.1 The Multiple-Destination Model

Previous research on the AS-Aware Tor problem considers Tor connections as point-to-point communications – a single client connects to a single destination over Tor. Modern web traffic, however, does not conform to this model, and our intuition suggests that most Tor user traffic is web traffic as opposed to FTP or some other form. Modern websites frequently load in resources from different domains. These resources are then retrieved by separate requests made by the user to different domains and they could happen at different times during or after the page load via HTML events or AJAX. Therefore, there are typically multiple destinations per website visit, so the number of exit-destination connections is greater than one for a single client. This increases the likelihood of an overlap in the client-guard and exit-destination network paths over the point-to-point model. Popular websites are typically cached in content distribution nodes (CDN), though, so these paths might be shorter than those under the point-to-point model. The shorter paths would consequently decrease the likelihood of an overlap in these paths. An illustration of this multi-destination model is given in Figure 3.1, which also examines the measurement components of this work. This work examines the effects that these two modern web content aspects have on the AS-Aware Tor problem.

3.2 Measuring Destinations per Website

The destinations are the resources that are contacted to complete a website load. Since these resources could be requested via HTML events or JavaScript, a full-browser is needed to emulate a normal Tor user. Therefore, the Selenium browser automation framework was used to visit each website in the Alexa Top 1000. Since resource requests and IP addresses will vary by the location of the request, each website must be requested through exit nodes in distinct ASes. To do this, Python's Stem library was used to programatically construct Tor circuits through exit relays in distinct ASes. The websites are then requested by URL through the constructed Tor circuits, allowing the exit relays to handle DNS resolution as they normally would. The IP addresses that are resolved for each resources are then logged. This process is repeated for each website and exit AS combination, and is illustrated in Figure 3.1a.

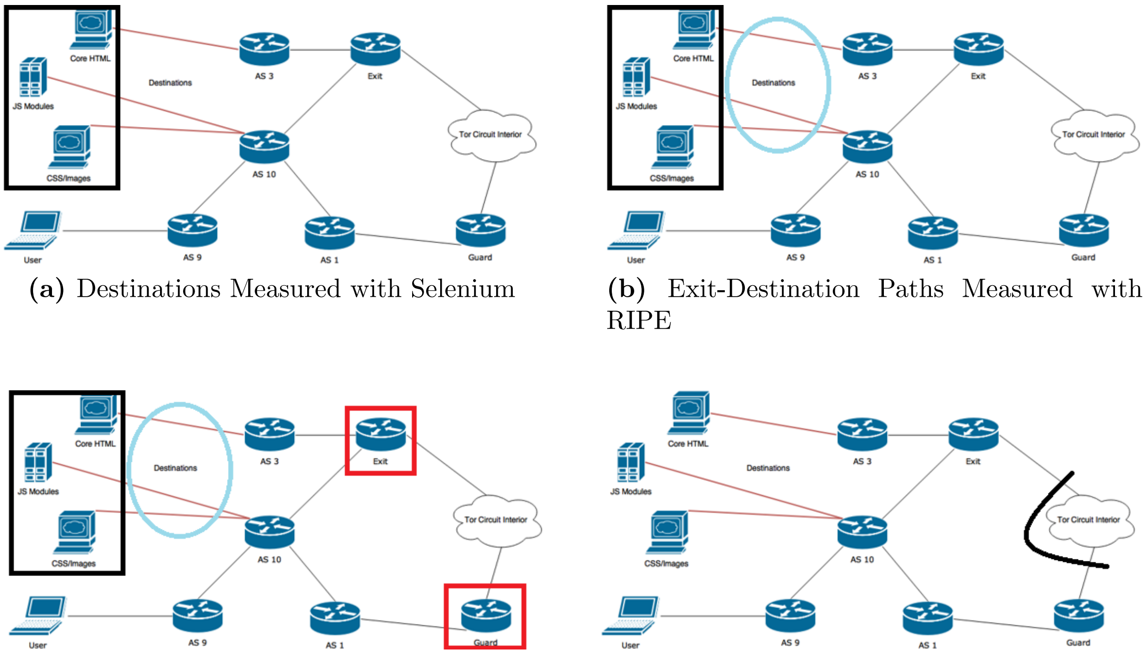
3.3 Measuring Exit-Destination Paths

Since the destinations of each website are measured in 3.2, the paths between the exit relays and these destinations must be measured to determine if they overlap with the client-guard paths. The network path between two hosts can be determined by performing a traceroute from one to the other. Unfortunately, Tor offers no way to perform traceroutes from exit nodes, so the exit-destination paths must be approximated. The AS-level path between the exit and destinations will determine the risk of de-anonymization in a scenario, so any host with the same AS-level path as an exit relay will provide a sufficiently close traceroute to the destinations. Since the finest-grained unit of routing on the internet is defined by the /24 of the IP address space, a host in the same /24 as an exit relay will have roughly the same AS-level path.

To approximate this path, RIPE Atlas probes are used to traceroute the destinations from the same ASes as the exit relays. RIPE Atlas is a network measurement infrastructure of volunteer-hosted probes around the world that allow individuals to request network measurements at the cost of credits. For each exit relay, the RIPE Atlas probe in the same AS as the exit relay conducts a traceroute to the IP addresses of the resources loaded for each website in the Alexa Top 1000. The IP-level traceroutes are then resolved to the AS-level path between the exit relay and the destination resources using BGP data from Routeviews. This provides the AS-level paths between the exit and destination resources as shown in Figure 3.1b.

3.4 Client-Guard Paths and Relay Selection

To measure the Client-Guard paths, a similar AS-level path approximation is obtained like that described in 3.3. CAIDA Ark traceroutes are used instead of RIPE Atlas traceroutes since the amount of guard relays vastly outnumbers the number of potential exit relays for a Tor circuit. The traceroutes to the same /24 address spaces of guard relays is used as an approximation of client-guard paths, where the client is the CAIDA Ark performing the traceroute. With the client-guard paths known, Tor relays must be selected based on available measurement vantage points. Therefore, only exit relays that have a RIPE Atlas probe in the same AS are considered. Likewise, only guard relays that share an AS with a CAIDA traceroute target are considered in this measurement. Once the potential guard and exit relays are known, like in Figure 3.1c, the client-guard paths can be approximated from the CAIDA traceroutes, resulting in full knowledge of the path except for the middle relay path. The known measurement components are summed up in Figure 3.1d.



(a) Destinations Measured with Selenium

(b) Exit-Destination Paths Measured with RIPE

(c) Known Relays

(d) Complete Path Sans Middle Relay

Figure 3.1: Measurement Components to Evaluate Risk of De-Anonymization

Chapter 4

Evaluation

4.1 Tor Circuit Simulation

To evaluate the risk of de-anonymization, the likelihood of an overlap in client-guard and exit-destination paths must be determined for each combination of client AS, guard AS, exit AS, and domain. Tor circuit construction is not random, however. The exit relay is first selected based on its bandwidth, the middle is selected based on the bandwidth of the exit, and likewise the guard is selected based on the bandwidth of the middle. Additional caveats are made based on the reputation of certain relays. Fortunately, prior work on simulating this path selection algorithm has already been done by [5] with their path simulator TorPS. TorPS, Tor Path Simulator, takes as input a consensus document, which details the Tor relays available for circuits and generates circuits based on the Tor path selection algorithm.

Using TorPS, the multinomial probability of guard and exit relays was calculated – the likelihood that a relay occurs is directly proportional to its frequency of occurrence in TorPS. Our simulator then used these probabilities to choose simulate Tor circuit construction along with a uniform sampling of all client ASes, which is equivalent to the ASes of the CAIDA Arks. We generate 1000 circuits, consisting of a client AS, guard relay, and exit relay, and determine if the client-guard and exit-destination paths intersect for each website in the Alexa Top 1000. Using the CAIDA Ark traceroutes between the client AS and guard relay, we determine the client-guard path. Then, for each resource loaded in from a website, the path from the exit to that resource is retrieved from the RIPE Atlas probes' traceroutes.

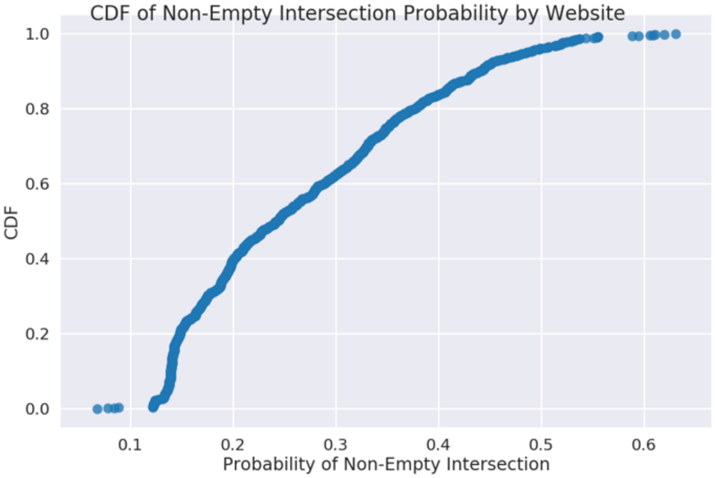
Now that we have both the client-guard and exit-destination paths, we take the intersection of the client-guard path and the path between th exit and each resource. If any of these intersections are non-empty, then that client is susceptible to de-anonymization when visiting that website.

4.2 Results of Path Intersections

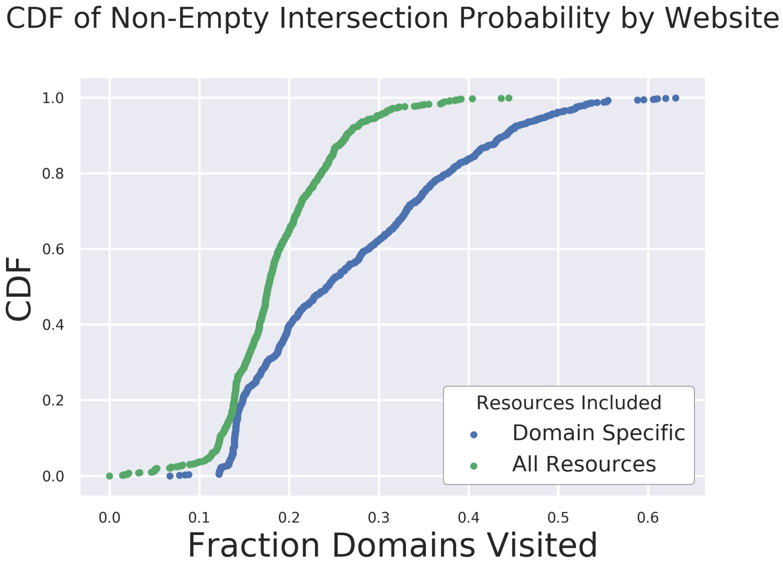
There are 114 distinct ASes that contain both a Tor exit relay and a RIPE Atlas probe, 130 distinct ASes that were the target of a CAIDA traceroute, and 203 distinct ASes that host a CAIDA Ark monitor. This corresponds to 114 potential exit ASes, 130 possible guard ASes, and 203 possible client ASes in this simulation. There are roughly 3 million possible combinations of these data points. We generate 1000 circuits, consisting of a client, guard, and exit AS, selected by the probabilities described in the prior section. For each domain, the exit-destination paths are intersected with the client-guard paths to determine if there is any intersection. The results of these intersections suggest that the risk of de-anonymization is greater than 20% for more than half of all the websites in the Alexa Top 1000, as indicated by the CDF illustrated in Figure 4.1a.

However, some resources are loaded in for nearly all websites; gstatic, for instance, is loaded into roughly 90% of the websites in the Alexa Top 1000, so being able to link a client to gstatic does not reveal much about the website the client is visiting. To determine the risk of de-anonmyization of more identifying resources, we examine the intersection of the client-guard paths with the exit-destination paths of resources with the same top-level domain (TLD) of the website that the client is visiting. The CDF of these TLD path intersections in Figure 4.1b indicate that there is still a non-trivial risk of de-anonymization. Examining only paths to resources with the same TLD only reduces the number of websites with greater than 20% chance of de-anonymization to roughly 400 websites.

Moreover, these distinct resources are typically located in distinct /24 IP address spaces, which are likely to have distinct AS-level paths. As shown in Figure 4.2, the average number of resources loaded in a website are almost directly proportional to the number of distinct /24s. Most content is not co-cached in the same CDN, if at all, so there are



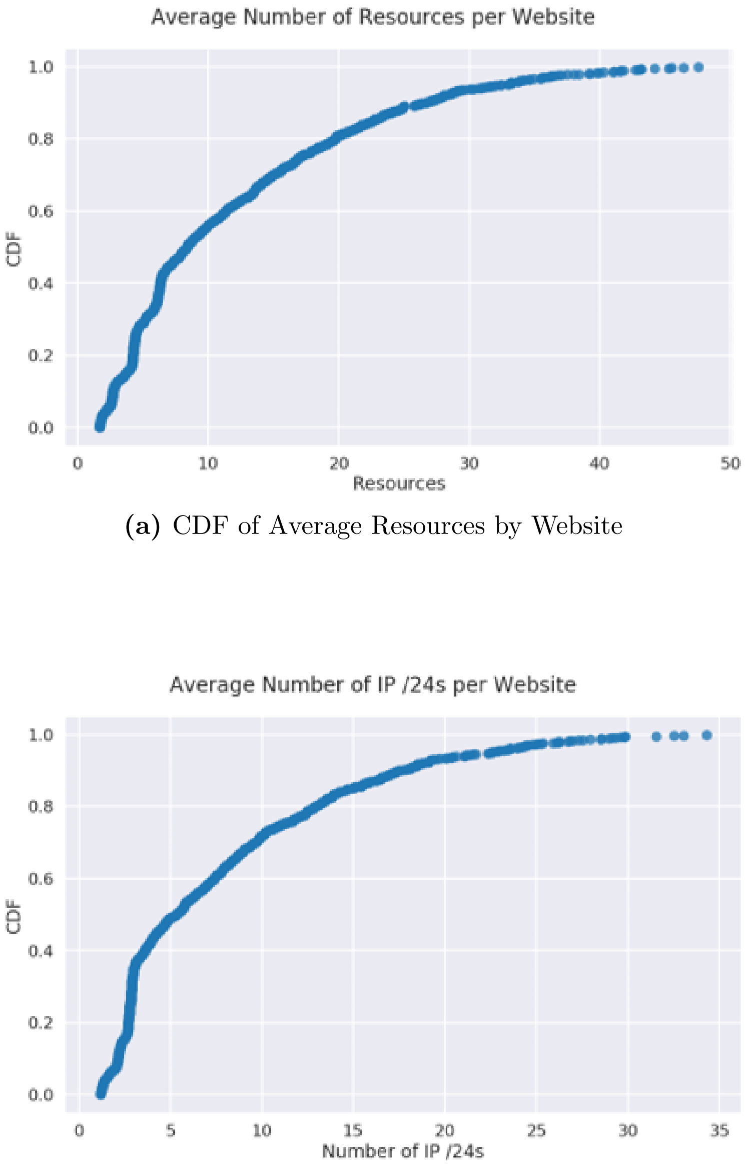
(a) CDF of Non-Empty Intersection by Website



(b) CDF of Non-Empty Intersection of Top-Level Domain Paths vs All

Figure 4.1: Risk of De-Anonymization over Websites

not many reductions in the number of paths. Therefore, the multiple resources contribute significantly to the risk of de-anonymization, and the shorter paths to CDNs do not significantly compensate for these additional paths. In comparison to the point-to-point model, a traceroute to the main resource in the NY Post website yielded a 21% chance of de-anonymization against an active AS adversary while loading all resources yields a 35% chance of de-anonymization.



(a) CDF of Average Resources by Website

(b) CDF of Average /24s by Website

Figure 4.2: Average Resources vs /24s

Chapter 5

Related Work

Related work on the AS-Aware Tor problem provides only partial countermeasures and does not reflect the multiple connections to CDNs that server cached content. Work by [3] collected actual Tor data, but did not collect any data on connection context in order to preserve user privacy. The results of the work suggest that client traffic is heavily generated by a few ASes and that there are a few ASes which are nearly unavoidable in terms of servicing client-guard and exit-destination paths. The authors only collected data pertaining to a small number of websites, though, and the modern web topology may have have changed significantly since the time of its writing.

A low-latency solution to AS-Aware Tor path selection was proposed in [1], which accounts for possible multiple destinations per Tor connection. The solution, however, inferred AS relationships based on BGP data, which as the authors point out is not a perfect prediction of the AS-level paths between relays. Additionally, the path selection requires prior knowledge of the destination location, which requires either a hard-coded IP addresses or out-of-band DNS requests. In addition to the un-avoidable ASes pointed out by [3], the path taken to avoid particular ASes may be sufficient information to fingerprint the destination of Tor connections. There are also latency increases as a trade-off for a supposed increase in anonymity.

Chapter 6

Conclusion

In this work we show that the previously studied point-to-point model of Tor connections does not capture the full risk of de-anonymization under the AS-Aware Tor problem. The results suggest that the multiple destinations visited per website request account for a significant portion of the risk of de-anonymization and that the shorter paths to CDNs do not compensate for the increased risk of visiting multiple destinations per website load. Moreover, the number of distinct /24s visited per resource is nearly proportional, suggesting that co-cached content is sparse. The significance of these results across a wide variety of measurement vantage points and relay selections indicate that these risks are very much real to current Tor users.

6.1 Future Work

The measurement system and data collected during this work provides many more opportunities for analysis and possible countermeasures. A network level adversary may not be able to discern a client's destination from a single resource that is widely shared across websites, but there has been significant work in fingerprinting website access via the advertisements and resources loaded. Specifically, some resources may be indicative of particular website accesses. In the future, we would like to determine how the anonymity set of website shrinks as particular resource requests are made known to a network adversary. Locating co-cached content and CDNs in the network topology would also be useful to avoid long paths between exits and destinations. In reference to 5, there are some proposed solutions to the AS-Aware Tor problem that should be vetted by this methodology and system. Specifically, the solutions in [1] should be examined under the multiple destination model and the path selection should be tested under a fingerprinting attack.

Bibliography

[1] Akhoondi, M., Yu, C., and Madhyastha, H. V. (2012). Lastor: A low-latency as-aware tor client. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 476–490. IEEE. 17, 19

[2] Dingledine, R., Mathewson, N., and Syverson, P. (2004). Tor: The second-generation onion router. Technical report, Naval Research Lab Washington DC. 1, 3

[3] Edman, M. and Syverson, P. (2009). As-awareness in tor path selection. In *Proceedings of the 16th ACM conference on Computer and communications security*, pages 380–389. ACM. 17

[4] Foundation, E. F. (N/A). How tor works. [Online; accessed April 1, 2018]. vii, 5

[5] Johnson, A., Wacek, C., Jansen, R., Sherr, M., and Syverson, P. (2013). Users get routed: Traffic correlation on tor by realistic adversaries. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 337–348. ACM. 2, 6, 12

[6] Shmatikov, V. and Wang, M.-H. (2006). Timing analysis in low-latency mix networks: Attacks and defenses. In *European Symposium on Research in Computer Security*, pages 18–33. Springer. 1

[7] Sun, Y., Edmundson, A., Vanbever, L., Li, O., Rexford, J., Chiang, M., and Mittal, P. (2015). Raptor: Routing attacks on privacy in tor. In *USENIX Security Symposium*, pages 271–286. 1, 6

Vita

Joseph ”Parker” Diamond is a 5 Year Master’s candidate from the University of Tennessee, Knoxville, with a focus on privacy, anonymity, and censorship circumvention. During his undergraduate, he completed a double major in computer science and mathematics before applying these degrees towards his current research. He is a co-founder of HackUTK, the first student-led cybersecurity organization at UT Knoxville, where he conducted several skill sessions on a regular basis. Parker worked previously as an undergraduate research at CURENT, a power systems research organization at the University of Tennessee. Parker has also published work on cryptocurrencies and mining pool payout schemes with his colleagues in VolSec, the UT security lab. Their work was recently presented at Financial Cryptography 2018 in Curacao.

Outside of his active research work, Parker has interest in IoT and mobile security. In his spare time, Parker works on CTF problems and learns new programming languages in order to explore developing systems. He is a contributor to several open-source projects. He is currently pursuing research positions at industry locations and national labs in order to continue contributing to the academic community before a possible return to a PhD program.