Dude, where's that IP? Circumventing measurement-based IP geolocation

Phillipa Gill *Yashar Ganjali*
Dept. of Computer Science
University of Toronto

Bernard Wong
Dept. of Computer Science
Cornell University

David Lie
Dept. of Electrical and Computer Engineering
University of Toronto

Abstract

Many applications of IP geolocation can benefit from geolocation that is robust to adversarial clients. These include applications that limit access to online content to a specific geographic region and cloud computing, where some organizations must ensure their virtual machines stay in an appropriate geographic region. This paper studies the applicability of current IP geolocation techniques against an adversary who tries to subvert the techniques into returning a forged result. We propose and evaluate attacks on both delay-based IP geolocation techniques and more advanced topology-aware techniques. Against delay-based techniques, we find that the adversary has a clear trade-off between the accuracy and the detectability of an attack. In contrast, we observe that more sophisticated topology-aware techniques actually fare worse against an adversary because they give the adversary more inputs to manipulate through their use of topology and delay information.

1 Introduction

Many applications benefit from using IP geolocation to determine the geographic location of hosts on the Internet. For example, online advertisers and search engines tailor their content based on the client's location. Currently, geolocation databases such as Quova [22] and MaxMind [16] are the most popular method used by applications that need geolocation services.

Geolocation is also used in many security-sensitive applications. Online content providers such as Hulu [13], BBC iPlayer [22], RealMedia [22] and Pandora [20], limit their content distribution to specific geographic regions. Before allowing a client to view the content, they determine the client's location from its IP address and allow access only if the client is in a permitted jurisdiction. In addition, Internet gambling websites must restrict access to their applications based on the client's location or risk legal repercussions [29]. Accordingly, these businesses rely on geolocation to limit access to their online services.

Looking forward, the growth of infrastructure-as-a-service clouds, such as Amazon's EC2 service [1], may also drive organizations using cloud computing to employ geolocation. Users of cloud computing deploy VMs on a cloud provider's infrastructure without having to maintain the hardware their VM is running on. However, differences in laws governing issues such as privacy, information discovery, compliance and audit require that some cloud users to restrict VM locations to certain jurisdictions or countries [6]. These location restrictions may be specified as part of a service level agreement (SLA) between the cloud user and provider. Cloud users can use IP geolocation to independently verify that the location restrictions in their cloud SLAs are met.

In these cases, the target of geolocation has an incentive to mislead the geolocation system about its true location. Clients commonly use proxies to mislead content providers so they can view content that is unauthorized in their geographic region. In response, some content providers [13] however, have identified and blocked access from known proxies; but this does not prevent all clients from circumventing geographic controls. Similarly, cloud providers may attempt to break location restrictions in their SLAs to move customer VMs to cheaper locations. Governments that enforce location requirements on the cloud user may require the geolocation checks to be robust *no matter what* a cloud provider may do to mislead them. Even if the cloud provider itself is not malicious, its employees may also try to relocate VMs to locations where they can be attacked by other malicious VMs [24]. Thus, while cloud users might trust the cloud service provider, they may still be required to cd ..have independent verification of the location of their VMs to meet audit requirements or to avoid legal liability.

IP geolocation has been an active field of research for almost a decade. However, all current geolocation techniques assume a benign target that is not trying to intentionally mislead the user, and there has been limited work on geolocating malicious targets. Castelluccia *et al.* apply Constraint-Based Geolocation (CBG) [12] to the problem of geolocating fast-flux hidden servers that use a layer of proxies in a botnet [5] to conceal their location. Muir and Oorschot [18] describe limitations of passive geolocation techniques (e.g., `whois` services) and present a technique for finding the IP address of a machine using the Tor anonymization network [28]. These previous works focus on de-anonymization of hosts behind proxies, while our contribution in this paper is to answer fundamental questions about whether current geolocation algorithms are suitable for security-sensitive applications:

- **Are current geolocation algorithms accurate enough to locate an IP within a certain country or jurisdiction?** We answer this question by surveying previously published studies of geolocation algorithms. We find that current algorithms have accuracies of 35-194 km, making them suitable for geolocation within a country.

- **How can adversaries attack a geolocation system?** We propose attacks on two broad classes of measurement-based geolocation algorithms – those relying on network delay measurements and those using network topology information. To evaluate the practicality of these attacks, we categorize adversaries into two classes – a simple adversary that can manipulate network delays and a sophisticated one with control over a set of routable IP addresses.

- **How effective are such attacks? Can they be detected?** We evaluate our attacks by analyzing them against models of geolocation algorithms. We also perform an empirical evaluation using measurements taken from PlanetLab [21] and executing attacks on implementations of delay-based and topology-aware geolocation algorithms. We observe the simple adversary has limited accuracy and must trade off accuracy for detectability of their attack. On the other hand, the sophisticated adversary has higher accuracy and remains difficult to detect.

The rest of this paper is structured as follows. Section 2 summarizes relevant background and previous work on geolocation techniques. The security model and assumptions we use to evaluate current geolocation proposals is described in Section 3. We develop and analyze attacks on delay-based and topology-aware geolocation methods in Sections 4 and 5, respectively. Section 6 presents related work that evaluates geolocation when confronted by a target that leverages proxies. We present conclusions in Section 7.

2 Geolocation Background

IP geolocation aims to solve the problem of determining the geographic location of a given IP address. The solution can be expressed to varying degrees of granularity; for most applications the result should be precise enough to determine the city in which the IP is located, either returning a city name or the longitude and latitude where the target is located. The two main approaches to geolocation use either active network measurements to determine the location of the host or databases of IP to location mappings.

Measurement-based geolocation algorithms [9, 12, 14, 19, 30, 31] leverage a set of geographically distributed *landmark* hosts with known locations to locate the *target* IP. These landmarks measure various network properties, such as delay, and the paths taken by traffic between themselves and the target. These results are used as input to the geolocation algorithm which uses them to determine the target's location using methods such as: constraining the region where the target may be located (geolocalization) [12, 30], iterative force directed algorithms [31], machine learning [9] and constrained optimization [14].

Geolocation algorithms mainly rely on `ping` [7] and `traceroute` [7] measurements. Ping measures the round-trip time (RTT) delay between two machines on the Internet, while `traceroute` discovers and measures the RTT to routers along the path to a given destination. We classify measurement-based geolocation algorithms by the type of measurements they use to determine the target's location. We refer to algorithms that use end-to-end RTTs as delay-based [9, 12, 31] and those that use both RTT and topology information as topology-aware algorithms [14, 30].

An alternative to measurement-based geolocation is geolocation using databases of IP to location mappings. These databases can be either proprietary or public. Public databases include those administered by regional Internet registries (e.g., ARIN [3], RIPE [23]). Proprietary databases of IP to geographic location mappings are provided by companies such as Quova [22] and Maxmind [16]. While the exact method of constructing these databases is not public, they are sometimes based on a combination of `whois` services, DNS LOC records and autonomous system (AS) numbers [2]. Registries and databases tend to be coarse grained, usually returning the headquarters location of the organization that registered the IP address. This becomes a problem when organizations distribute their IP addresses over a wide geographic region, such as large ISPs or content providers. Mislead-

Table 1: Average accuracy of measurement-based geolocation algorithms.

Class	Algorithm	Average accuracy (km)
Delay-based	GeoPing [19]	150 km (25th percentile); 109 km (median) [30]
	CBG [12]	78-182
	Statistical [31]	92
	Learning-based [9]	407-449 (113 km less than CBG [12] on their data)
Topology-aware	TBG [14]	194
	Octant [30]	35-40 (median)
Other	GeoTrack [19]	156 km (median) [30]

ing database geolocation is also straightforward through the use of proxies.

DNS LOC [8] is an open standard that allows DNS administrators to augment DNS servers with location information, effectively creating a publicly available database of IP location information. However, it has not gained widespread usage. In addition, since the contents of the DNS LOC database are not authenticated and are set by the owners of the IP addresses themselves, it is poorly suited for security-sensitive applications.

Much research has gone into improving the accuracy of measurement-based geolocation algorithms; consequently, they provide fairly reliable results. Table 1 shows the reported average accuracies of recently proposed geolocation algorithms. Based on the reported accuracies, we believe that current geolocation algorithms are sufficiently accurate to place a machine within a country or jurisdiction. In particular, CBG [12] and Octant [30] appear to offer accuracies well within the size of most countries and may even be able to place users within a metropolitan area. Measurement-based geolocation is particularly appealing for secure geolocation because if a measurement can reach the target (e.g., using application layer measurements [17]), even if it is behind a proxy (e.g., SOCKS or HTTP proxy), the effectiveness of proxying will be diminished.

3 Security Model

We model secure geolocation as a three-party problem. First, there is the geolocation *user* or *victim*. The user hopes to accurately determine the location of the target using a geolocation algorithm that relies on measurements of network properties[1]. We assume that; (1) the user has access to a number of landmark machines distributed around the globe to make measurements of RTTs and network paths, and (2) the user trusts the results of measurements reported by landmarks. Second, there is the *adversary*, who owns the target's IP address. The adversary would like to mislead the user into believing that the target is at a *forged location* of the adversary's choosing, when in reality the target is actually located at the *true location*. The adversary is responsible for physically connecting the target IP address to the Internet, which allows them to insert additional machines or routers between the target and the Internet. The third party is the *Internet* itself. While the Internet is impartial to both adversary and user, it introduces additive noise as a result of queuing delays and circuitous routes. These properties introduce some inherent inaccuracy and unpredictability into the results of measurements on which geolocation algorithms rely. In general, an adversary's malicious tampering with network properties (such as adding delay), if done in small amounts, is difficult to distinguish from additive noise introduced by the Internet.

This work addresses two types of adversaries with differing capabilities. We assume in both cases that the adversary is fully aware of the geolocation algorithm and knows both the IP addresses and locations of all landmarks used in the algorithm. The first, *simple adversary* can tamper only with the RTT measurements taken by the landmarks. This can be done by selectively delaying packets from landmarks to make the RTT appear larger than it actually is. The simple adversary was chosen to resemble a home user running a program to selectively delay responses to measurements. The second, *sophisticated adversary*, controls several IP addresses and can use them to create fake routers and paths to the target. Further, this adversary may have a wide area network (WAN) with several gateway routers and can influence BGP routes to the target. The sophisticated adversary was chosen to model a cloud provider as the adversary. Many large online service providers already deploy WANs [11], making this attack model feasible with low additional cost to the provider.

We make two assumptions in this work. First, while aware of the geolocation algorithm being used, and the location and IP addresses of all landmarks, the adversary cannot compromise the landmarks or run code on them. Thus, the only way the adversary can compromise the integrity of network measurements is to modify the properties of traffic traveling on network links directly connected to a machine under its control.

The second assumption is that network measurements made by landmarks actually reach the target. Otherwise, an adversary could trivially attack the geolocation system by placing a proxy at the forged location that responds to all geolocation traffic and forwards all other traffic to the true location. To avoid this attack, the user can either combine the measurements with regular traffic or protect it using cryptography. For example, if the geolocation user is a Web content provider, Muir and Oorschot [18] have shown that even an anonymization network such as Tor [28] may be defeated using a Java applet embedded in a Web page. Users who want to geolocate a VM in a compute cloud may require the cloud provider to support tamper-proof VMs [10, 25] and embed a secret key in the VM for authenticating end-to-end network measurements. In this case, the adversary would need to place a copy of the VM in the forged location to respond to measurements. Given that the adversary is trying to avoid placing a VM in the forged location, it is not a practical attack for a malicious cloud provider.

4 Delay-based geolocation

Delay-based geolocation algorithms use measurements of end-to-end network delays to geolocate the target IP. To execute delay-based geolocation, the landmarks need to calibrate the relationship between geographic distance and network delay. This is done by having each landmark, L_i, ping all other landmarks. Since the landmarks have known geographic locations, L_i can then derive a function mapping geographic distance, g_{ij}, to network delay, d_{ij}, observed to each other landmark L_j where $i \neq j$ [12]. Each landmark performs this calibration and develops its own mapping of geographic distance to network delay. After calibrating its distance-to-delay function, it then pings the target IP. Using the distance-to-delay function, the landmark can then transform the observed delay to the target into a predicted distance to the target. All landmarks perform this computation to triangulate the location of the target.

Delay-based geolocation operates under the implicit assumption that network delay is well correlated with geographic distance. However, network delay is composed of queuing, processing, transmission and propagation delay [15]. Where only the propagation time of network traffic is related to distance traveled, and the other components vary depending on network load, thus adding noise to the measured delay. This assumption is also violated when network traffic does not take a direct ("as the crow flies") path between hosts. These indirect paths are referred to as "circuitous" routes [30].

There are many proposed methods for delay-based geolocation, including GeoPing [19], Statistical Geolocation [31], Learning-based Geolocation [9] and CBG [12]. These algorithms differ in how they express the distance-to-delay function and how they triangulate the position of the target. GeoPing is based on the observation that hosts that are geographically close to each other will have delay properties similar to the landmark nodes [19]. Statistical Geolocation develops a joint probability density function of distance to delay that is input into a force-directed algorithm used to geolocate the target [31]. In contrast, Learning-based Geolocation utilizes a Naïve Bayes framework to geolocate a target IP given a set of measurements [9]. CBG has the highest reported accuracy of the delay-based algorithms, with a mean error of 78-182 km [12]. The remainder of this section therefore focuses on CBG to model and evaluate how an adversary can influence delay-based geolocation techniques.

CBG [12] establishes the distance-delay function, described above, by having the landmarks ping each other to derive a set of points (g_{ij},d_{ij}) mapping geographic distance to network delay. To mitigate the effects of congestion on network delays, multiple measurements are made, and the 2.5-percentile of network delays are used by the landmarks to calibrate their distance-to-delay mapping. Each landmark then computes a linear ("best line") function that is closest to, but below, the set of points. Distance between each landmark and the target IP is inferred using the "best line" function. This gives an implied circle around each landmark where the target IP may be located. The target IP is then predicted to be in the region of intersection of the circles of all the landmarks. Since the result of this process is a *feasible region* where the target may be located, CBG determines the centroid of the region and returns this value as the geolocation result. Gueye *et al.* observe a mean error of 182 km in the US and 78 km in Europe. They also find that the feasible region where the target IP may be located ranges from 10^4 km^2 in Europe to 10^5 km^2 in North America.

4.1 Attack on delay-based geolocation

Since delay-based geolocation techniques do not take network topology into account, the ability of a sophisticated adversary to manipulate network paths is of no additional value. Against a delay-based geolocation algorithm, the simple and sophisticated adversaries have equal power.

To mislead delay-based geolocation, the adversary can manipulate distance of the target computed by the landmarks by altering the delay observed by each landmark. The adversary knows the identities and locations of each landmark and can thus identify traffic from the landmarks and alter the delay as necessary. To make the target at the true location, t, appear to be at forged location, τ, the adversary must alter the perceived delay, d_{it}, be-

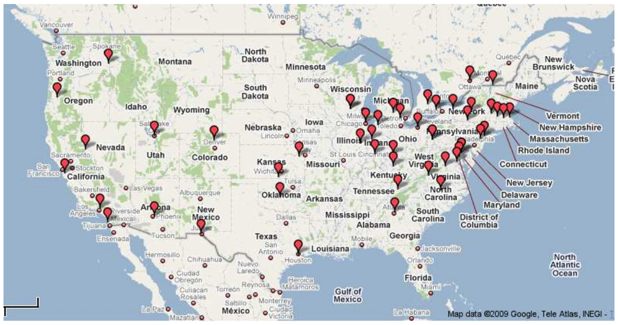

Figure 1: Landmarks (PlanetLab nodes) used in evaluation.

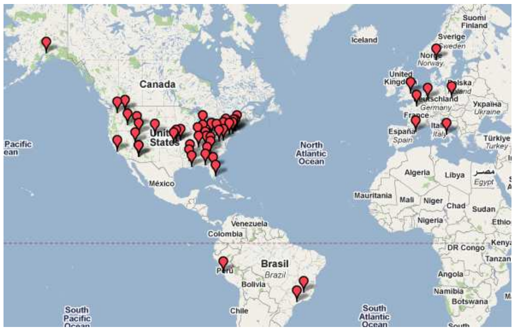

Figure 2: Forged locations (τ) used in the evaluation.

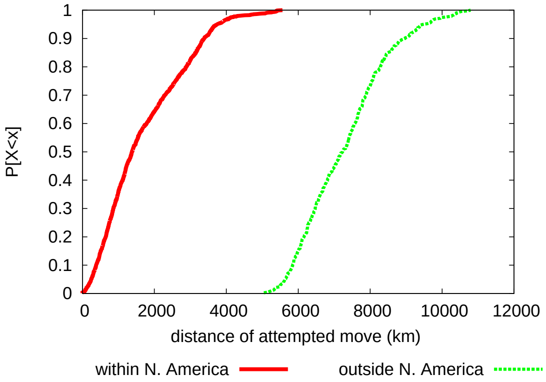

Figure 3: CDF of the distance the adversary tries to move the target.

tween each landmark, L_i and t to become the delay, $d_{i\tau}$, each landmark should perceive between L_i and τ. To do this, two problems must be solved. The adversary must first find the appropriate delay, $d_{i\tau}$, for each landmark and then change the perceived delay to the appropriate delay.

If the adversary controls a machine at or near τ, she may directly acquire the appropriate $d_{i\tau}$ for each landmark by pinging each of the landmarks from the forged location τ. However, pings to all the landmarks from a machine not related to the geolocation algorithm may arouse suspicion. Also, it may not be the case that the adversary controls a machine at or near τ.

Alternatively, with knowledge of the location of the landmarks, the adversary can compute the geographic distances g_{it} and $g_{i\tau}$ between each landmark L_i and the true location t as well as the forged location τ. This enables the adversary to determine the additional distance a probe from L_i would travel ($\gamma_i = g_{i\tau} - g_{it}$) had it actually been directed to the forged location τ. The next challenge is to map γ_i into the appropriate amount of delay to add. To do this, the adversary may use 2/3 the speed of light in a vacuum (c) as a lower-bound approximation for the speed of traffic on the Internet [14]. Thus, the required delay to add to each ping from L_i is:

$$\delta_i = \frac{2 \times \gamma_i}{2/3 \times c} \tag{1}$$

The additional distance the ping from L_i would travel is multiplied by 2 because the delay measured by `ping` is the round-trip time as opposed to the end-to-end delay. This approximation is the lower bound on the delay that would be required for the ping to traverse the distance $2 \times \gamma_i$ because the speed of light propagation is the fastest data can travel between the two points.

Armed with this approximation of the appropriate $d_{i\tau}$ for each landmark, the adversary can now increase the delay of each probe from the landmarks. The perceived delay cannot be decreased since this would require the adversary to either increase the speed of the network path between t and L_i, or slow down probes from L_i during its calibration phase. Since the adversary cannot compromise the landmarks and does not control network paths that are not directly connected to one of her machines, she is not able to accomplish this. As a result, the adversary may only modify landmark delays that need to be increased (i.e., $d_{i\tau} > d_{it}$). For all other landmarks, she does not alter the delays. Thus, even with perfect knowledge of the delays $d_{i\tau}$, neither a simple nor sophisticated adversary will be able to execute an attack perfectly on delay-based geolocation techniques.

4.2 Evaluation

We evaluate the effectiveness of our proposed attack against a simulator that runs the CBG algorithm proposed by Gueye *et al.* [12]. We collected measurement inputs for the algorithm using 50 PlanetLab nodes. Each node takes a turn being the target with the remaining 49 PlanetLab nodes being used as landmarks. Figure 1 shows the locations of the PlanetLab nodes. Each tar-

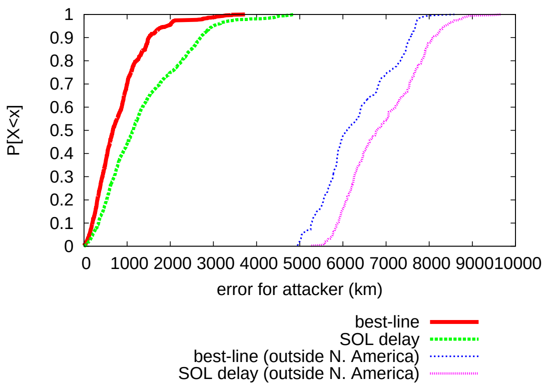

Figure 4: CDF of error distance for the adversary when attacking delay-based geolocation using speed of light (SOL) or best line delay.

get is initially geolocated using observed network delays. The target is then moved to 50 forged locations using the delay-adding attack, shown in Figure 2. We select 40 of the forged locations based on the location of US universities and 10 based on the location of universities outside of North America. This results in a total of 2,500 attempted attacks on the CBG algorithm.

In the delay adding attack, the adversary cannot move a target that is not within the same region as the landmarks into that region. For example, if the target is located in Europe, moving it to a forged location in North America would require reducing delay to all landmarks, which is not possible. This implies that if a geolocation provider wants to prevent the adversary from moving the target into a specific region, it should place their landmarks in this desired region.

Figure 3 shows the CDF of the distances the adversary attempts to move the target. In North America, the target is moved less than 4,000 km most of the time moved moved less than 1,379 km 50% of the time. Outside of North America, the distance moved consistently exceeds 5,000 km.

We evaluate the delay-adding attack under two circumstances: (1) when the adversary knows exactly what delay to add (by giving the adversary access to the “best line” function used by the landmarks), and (2) when the adversary uses the speed of light (SOL) approximation for the additional delay.

4.2.1 Attack effectiveness

Since the adversary is only able to increase, and not decrease, perceived delays, there are errors between the forged location, τ, and the actual location, r, returned by the geolocation algorithm. To understand why these errors exist, consider Figure 5. The arcs labeled g_1, g_2,

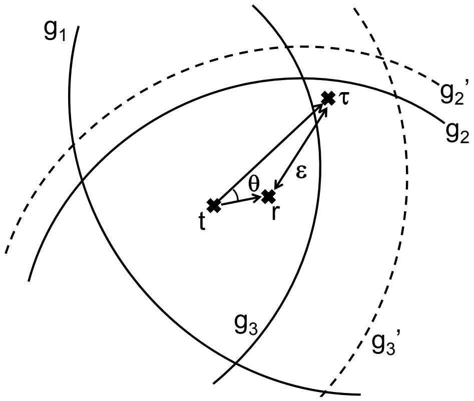

Figure 5: Attacking delay-based geolocation.

and g_3 are the circles drawn by 3 landmarks when geolocating the target. The region enclosed by the arcs is the feasible region, and the geolocation result is the centroid of that region. To move t to τ, the adversary should increase the radii of g_2 and g_3 and decrease the radius of g_1. However, as described earlier, delay can only be added, meaning that the adversary can only increase the radii of g_2 and g_3 to g_2' and g_3', respectively (shown by the dotted lines). Since the delay of g_1 cannot be decreased, this results in a larger feasible region with a centroid r that does not quite reach τ. We call the difference between the geolocation result (r) and forged location (τ) the error distance (ε) for the adversary. The difference between the intended and actual direction of the move is the angle θ.

We begin by evaluating the error distance, ε. Figure 4 shows the CDF of error for the adversary over the set of attempted attacks in our evaluation. Within North America, an adversary using the speed of light approximation has a median error of 1,143 km. When the adversary has access to the best line function,their error decreases to 671 km. As a reference, 671 km is approximately half the width of Texas. This indicates that when moving within North America, it is possible for an adversary with access to the best line function to be successful in trying to move the target into a specific state. We note that three of the targets used in our evaluation were located in Canada. Using the speed of light approximation these Canadian targets are able to appear in the US 65% of the time. Using the best line function, they are able to move into the US 89% of the time.

Outside of North America, the delay-adding attack has poor accuracy with a minimum error for the adversary of 4,947 km. As a reference, the distance from San Francisco to New York City is 4,135 km. Error of this magnitude is not practical for an adversary attempting to place the target in a specific country. For the remainder of this section, we focus on attacks where the adversary tries

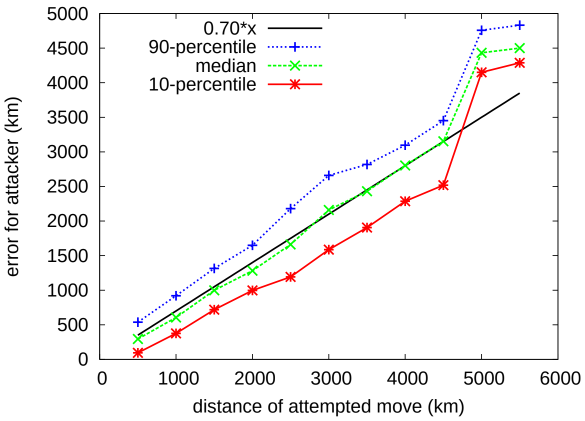

Figure 6: Error observed by the adversary depending on distance of their attempted move for the delay-adding attack.

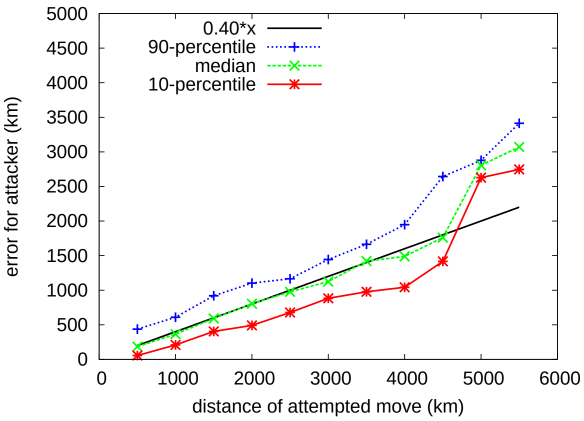

Figure 7: Error observed by the adversary depending on distance of their attempted move for the delay-adding attack when they have access to the best line function.

to move within North America because the error for the adversary is more reasonable.

We next consider how the distance the adversary tries to move the target affects the observed error. Figure 6 shows error for the adversary depending on how far the adversary attempts to move the target when using the speed of light approximation. Figure 7 shows the same data for an adversary with access to the best line function. We note that the error observed by the adversary grows with the magnitude of the attempted move by the adversary. Specifically, for each 1 km the adversary tries to move the median error increases by 700 meters when she does not have access to the best line function. With access to the best line function, the median error per km decreases by 43% to 400 km. Thus, the attack we propose works best when the distance between t and τ is relatively small and the error observed by the attacker grows linearly with the size of the move.

Given the relatively high errors observed by the adversary, we next verify whether the adversary moves in her chosen direction. Figure 8 shows the CDF of θ, the difference between the direction the adversary tried to move and the direction the target was actually moved. While lacking high accuracy when executing the delay-adding attack, the adversary is able to move the target in the general direction of her choosing. The difference in direction is less than 45 degrees 74% of the time and less than 90 degrees 89% of the time. The attack where the adversary has access to the best line function performs better with a difference in direction of less than 45 degrees 91% of the time.

4.2.2 Attack detectability

We next look at whether a geolocation provider can detect the delay-adding attack and thus determine that the geolocation result has been tampered with.

When CBG geolocates a target, it determines a feasible region where the target can be located [12]. The size of the feasible region can be interpreted as a measure of confidence in the geolocation result. A very large region size indicates that there is a large area where the target may be located, although the algorithm returns the centroid. As we saw in Figure 5, the adversary, able only to add delay, can only increase the radii of the arcs and thus only increase the region size. As a result, the delay-adding attack always increases the feasible region size and reduces confidence in the result of the geolocation algorithm. We consider the region size computed by CBG before and after our proposed attack to determine how effective region size may be for detecting an attack.

Figure 9 shows the region size for CBG when the delay-adding attack is executed in general, when the attack only attempts to move the landmark less than 1,000 km, and where the adversary has access to the best line function. We observe that the region size becomes orders of magnitude larger when the delay-adding attack is executed. The region size grows even larger when the adversary uses the best line function. An adversary that moves the target less than 1,000 km is able to execute the attack without having much impact on the region size distribution.

The region size grows in proportion to the amount of delay added. This explains why the adversary creates a larger region size when using the best line function, which adds more delay than the speed of light approxi-

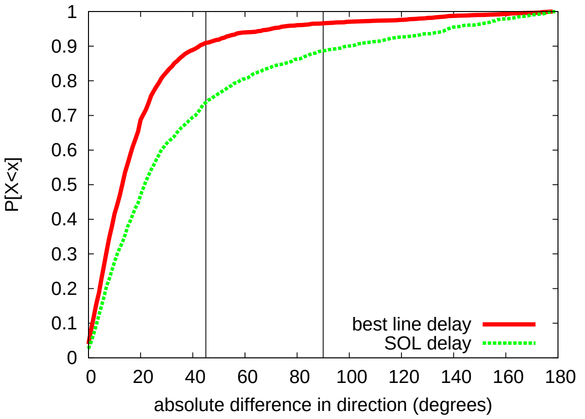

Figure 8: CDF of change in direction for the delay-adding attack.

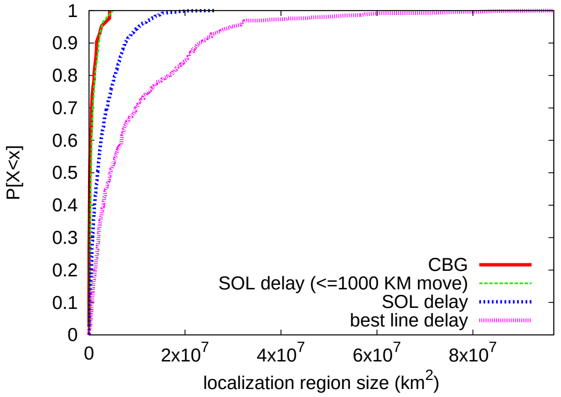

Figure 9: CDF of region size for CBG before and after the delay-adding attack.

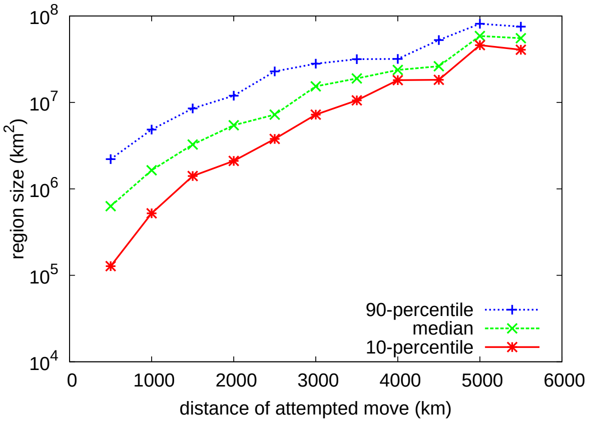

Figure 10: Region size depending on how far the adversary attempts to move the target using the best line function.

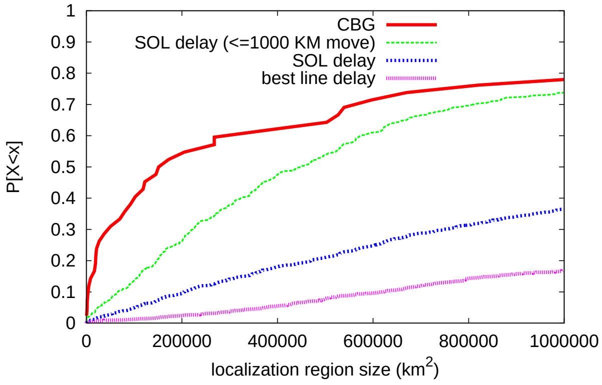

Figure 11: CDF of region size for CBG before and after delay-adding, limited to points less than 1,000,000 km^2.

mation. Figure 10 illustrates this case. As the adversary attempts to move the target further from its true location, the amount of delay that must be added increases. This in turn increases the region size returned by CBG. Thus, while there may be methods for adding delay that improve the adversary's accuracy, they will only increase the ability of the geolocation provider to detect the attack.

Given the increased region sizes observed when the delay-adding attack is executed, one defense would be to use a region size threshold to exclude geolocation results with insufficient confidence. Increased region sizes may be caused by an adversary adding delays, as we have observed or by fluctuations in the stochastic component of network delay. In either case, the geolocation algorithm observes a region that is too large for practical purposes. Suppose we discard all geolocation results with a region size greater than 1,000,000 km^2 (this is approximately the size of Texas and California combined). Figure 11 shows the CDF of region size below this threshold. The adversary using the speed-of-light approximation will be undetected only 36% of the time. However, if the adversary attempts to move less than 1,000 km she will remain undetected 74% of the time. An adversary with access to the best line for each of the landmarks is more easily detectable because of the larger region sizes that result from the larger injected delays. With a threshold of 1,000,000 km^2, the adversary using the best line function will have her results discarded 83% of the time. Thus, using a threshold on the region size is effective for detecting attacks on delay-based geolocation except when the attacker tries to move the target only a short distance.

5 Topology-aware geolocation

Delay-based geolocation relies on correlating measured delays with distances between landmarks. As we saw previously, these correlations or mappings are applied to landmark-to-target delays to create overlapping confidence regions; the overlap is the feasible region, and the estimated location of the target is its centroid. When inter-landmark delays and landmark-to-target delays are not similarly correlated with physical distances (e.g., due to circuitous end-to-end paths) the resulting delay-to-distance relationships to the target can deviate significantly from the pre-computed correlations.

Topology-aware geolocation addresses this problem by limiting the impact of circuitous end-to-end paths; specifically, it localizes all intermediate routers in addition to the target node, which results in a better estimate of delays. Starting from the landmarks, the geolocation algorithm iteratively estimates the location of all intermediate routers on the path between the landmark and the target. This is done solely based on single-hop link delays, which are usually significantly less circuitous than multi-hop end-to-end paths, enabling topology-aware geolocation to be more resilient to circuitous network paths than delay-based geolocation.

There are two previously proposed topology-aware geolocation methods, topology-based geolocation (TBG) [14] and Octant [30]. These methods differ in how they geolocate the intermediate routers. TBG uses delays measured between intermediate routers as inputs to a constrained optimization that solves for the location of the intermediate routers and target IP [14]. In contrast, Octant leverages a "geolocalization" framework similar to CBG [12], where the location of the intermediate routers and target are constrained to specific regions based on their delays from landmarks and other intermediate routers [30]. These delays are mapped into distances using a convex hull rather than a linear function, such as the best line in CBG to improve the mapping between distance and delay.

Octant leverages several optimizations that improve its performance over other geolocation algorithms. These include: taking into account both positive and negative constraints; accounting for fixed delays along network paths, and decreasing the weight of constraints based on latency measurements. Wong *et al.* find that their scheme outperforms CBG, with median accuracies of 35-40 km [30]. In addition, the feasible regions returned by Octant are much smaller than those returned by CBG. They also observe that their scheme is robust even given a small number of landmarks with performance leveling off after 15 landmarks.

When analyzing and evaluating attacks on topology-aware geolocation, we consider a generic geolocation framework. Intermediate routers are localized using constraints generated from latencies to adjacent routers. The target is localized to a feasibility region generated based on latencies from the last hop(s) before the target, and the centroid of the region is returned.

5.1 Delay-based attacks on topology-aware geolocation

Topology-aware geolocation systems localize all intermediate routers in addition to the target node. We begin by analyzing how a simple adversary, one without the ability to fabricate routers, could attack the geolocation system, and then move onto how a sophisticated adversary could apply additional capabilities to improve the attack. Since the simple adversary has no control over the probes outside her own network, any change made can only be reflected on the final links of the path towards the target.

Most networks are usually connected to the rest of the Internet via a small number of gateway routers. Any path connecting nodes outside the adversary's network to the target (which is inside the network) will go through one of these routers. Here, we start with a simple case where all routes towards the target converge on a single gateway router; we then consider the more general case of multiple gateway routers.

CLAIM: 1 *If the network paths from the landmarks to the target converge to a single common gateway router, increasing the end-to-end delays between the landmarks and the target can be detected and mitigated by topology-aware geolocation systems.*

To verify this claim, we first characterize the effect of delay-based attacks on topology-aware geolocation. Delay-based attacks selectively increase the delay of the probes from landmarks. The probe from landmark L_i is delayed for an additional δ_i seconds. Given that all network paths to the target converge to a single common gateway router h, the end-to-end delay from each landmark, L_i, to the target can be written as:

$$d_{it} = d_{ih} + d_{ht} + \delta_i \tag{2}$$

The observed latency from the gateway to the target is $d_{it} - d_{ih}$, which is the sum of the real last-hop latency and the attack delay. However, since the delay-based attack relies on selectively varying the attack delays, δ_i, based on the location of L_i, the observed last-hop latency between the gateway and the target will be inconsistent across measurements initiated from different landmarks.

The high-variance in the last-hop link delay can be used to detect delay-based attacks in topology-aware geolocation systems. The attack can be mitigated by taking the minimum observed delay for each link. The resulting observed link delay from h to the target is:

$$\hat{d}_{ht} = d_{ht} + \min_{L_i \in L} \delta_i \tag{3}$$

This significantly reduces the scope of delay-based attacks, requiring attack delays to be uniform across all measurement vantage points when there is only a single common gateway to the target.

In general, if there are multiple gateway routers on the border of the adversary's network, we can make the following weaker claim:

CLAIM: 2 *Increasing the delay between each gateway and the target can only be as effective against topology-based geolocation as increasing end-to-end delays against delay-based geolocation with a reduced set of landmarks.*

An adversary could attempt to modify delays between each gateway router, h_j, and the target, t. This assumes the adversary knows the approximate geolocation results for all gateway routers [2]. Where there is only a single gateway router with no additional attack delay, topology-based geolocation places the target within a circle centered at h with coordinates (λ_h, ϕ_h):

$$\sqrt{(x - \lambda_h)^2 + (y - \phi_h)^2} = d_{ht} \tag{4}$$

Subjecting the latency measurement to an additional delay, δ, changes the equation to the following:

$$\sqrt{(x - \lambda_h)^2 + (y - \phi_h)^2} = d_{ht} + \delta \tag{5}$$

Thus, for targets with a single gateway router, an adversary can only increase the localization region by introducing an additional delay without changing the location of the region's geometric center.

For targets with multiple gateway routers $H = h_0, h_1, ..., h_n$, targets are geolocated based on the delays between the gateways and t. An adversary can add additional delay, δ_j, between each gateway, h_j, and t based on the location of h_j. This is equivalent to the delay-adding attack, except the previously geolocated gateway routers are used in place of the real landmarks. Therefore, the previous evaluation results for the delay-adding attack on delay-based geolocation can be extended to topology-based geolocation for targets with multiple gateway routers.

5.2 Topology-based attacks

In topology-based geolocation, intermediate nodes are localized to confidence regions, and geographic constraints constructed from these intermediate nodes are expanded by their confidence regions to account for the accumulation of error. However, this does not result in a monotonic increase in the region size of intermediate nodes with each hop. The intersection of several expanded constraints for intermediate nodes along multiple network paths to the target can still result in intermediate nodes that are localized to small regions. A sophisticated adversary with control over a large administrative domain can exploit this property by fabricating nodes, links and latencies within its network to create constraint intersections at specific locations. This assumes that the adversary can detect probe traffic issued from geolocation systems in order to present a topologically different network without affecting normal traffic.

Externally visible nodes in an adversary's network consist of gateway routers $ER = \{er_0, er_1, ..., er_m\}$, internal routers $F = \{f_0, f_1, ..., f_n\}$ and end-points $T = \{\tau_0, \tau_1, ..., \tau_s\}$. Internal routers can be fictitious, and network links between internal routers can be arbitrarily manufactured. The adversary's network can be described as the graph $G = (V, E)$, where $V = F \cup ER \cup T$ represents routers, and $E = \{e_0, e_1, ..., e_k\}$ with weights $w(e_i)$ is the set of links connecting the routers with weights representing network delays.

All internal link latencies, including those between gateways, can be fabricated by the adversary. However, the delay between fictitious nodes must respect the speed-of-light constraint, which dictates that a packet can only travel a distance equal to the product of delay and the speed-of-light in fiber.

CLAIM: 3 *Topology-based attacks require the adversary to have more than one geographically distributed gateway router to its network.*

This claim follows from the analysis of delay-based attacks when all network paths to the target converge to a common gateway router. With only one gateway router to the network, changes to internal network nodes can affect only the final size of the localization region, not the region's geometric center.

CLAIM: 4 *An adversary with control over three or more geographically distributed gateway routers to its network can move the target to an arbitrary location.*

Unlike delay-based attacks that can only increase latencies from the landmarks to the target, topology-based attacks can assign arbitrary latencies from the ingress points to the target. From geometric triangulation, this enables topology-based attacks to, theoretically, triangulate the location of the target to any point on the globe given three or more ingress points.

In practice, there are challenges that limit the adversary from achieving perfect accuracy with this attack. Specifically, the attack requires the adversary to know the estimated location of the gateway routers and to have an accurate model of the delay-to-distance function used by the geolocation system. Such information can be reverse-engineered by a determined adversary by analyzing the geolocation results of other targets in the adversary's network.

Although a resourceful adversary's topology-based attack can substantially affect geolocation results, it can also introduce additional *circuitousness* to all network paths to the target that creates a detectable signature. Circuitousness refers to the ratio of actual distance traveled along a network path to the direct distance between the two end points of a path. Circuitousness can be observed by plotting the location of intermediate nodes as they are located by the topology-aware geolocation system.

5.2.1 Naming attack extension

State-of-the-art, topology-based geolocation systems [14, 30] leverage the structured way in which most routers are named to extract more precise information about router location. A collection of common naming patterns is available through the *undns* tool [27], which can extract approximate city locations from the domain names of routers.

When geolocation relies on `undns`, an adversary can effectively change the observed location of the target even with only a single gateway router to its network. This naming attack requires the adversary is capable of crafting a domain name that can deceive the *undns* tool, poisoning the *undns* database with erroneous mappings or responding to traceroutes with a spoofed IP address. The adversary only needs to use the naming attack to place any last hops before the target at its desired geographic location. The target will then be localized to the same location as this last hop in the absence of sufficient constraints.

Naming attacks exhibit the same increased circuitousness as standard topology-based attacks. Extensive poisoning of the *undns* database could allow an attacker to change the location of other routers along the network paths to reduce path circuitousness.

5.3 Evaluation

We evaluate the topology-based (hop-adding) attack and *undns* naming extension using a simulator of topology-aware geolocation. To perform the evaluation, we developed the fictitious network illustrated in Figure 12. The network includes 4 gateway routers (ER), represented by PlanetLab nodes in Victoria, BC; Riverside, CA; Ithaca, NY, and Gainesville, FL. The network also includes 11 forged locations (T) and 14 non-existent internal routers (F). Three of the non-existent routers are geographically distributed around the US, while the other 11 are placed close to the forged locations to improve the effectiveness of the attack, especially when the adversary can manipulate *undns* entries. Routers in the fictitious network are connected using basic heuristics. For example, each of the 11 internal routers near the forged locations is connected to the 3 routers nearest them to aid in triangulation. We show that even using this simple network design, an adversary executing the hop-adding attack and *undns* extension can be successful.

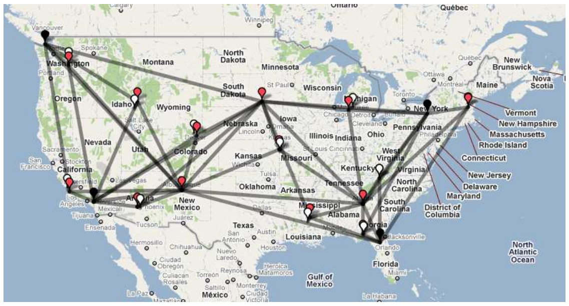

Figure 12: The adversary's network used for evaluating the topology-based attack.

To evaluate the attack, we use the same set of 50 PlanetLab nodes used in evaluating the delay-adding attack (Figure 1), with an additional 30 European PlanetLab nodes that act only as targets attempting to move into North America. We move the targets to the 11 forged locations in the fictitious network. These locations, a subset of the 40 US locations used in evaluating the delay-adding attack, were chosen to be geographically distributed around the US. Each of the 80 PlanetLab nodes takes a turn being the target with the remaining US PlanetLab nodes used as landmarks. Each target is moved to each of the 11 forged locations in turn, for a total of 880 attacks.

When executing the attack, the traceroute from each landmark is directed to its nearest gateway router. The first part of the traceroute is dictated by the network path between the landmark and its nearest gateway router (represented by a PlanetLab node). The second part is artificially generated to be the shortest path between the gateway router and the forged location. The latency of the second part is lower bounded by the speed-of-light delay between the gateway router and the target's true location. When the speed-of-light latency between the gateway router and the target is greater than the latency on the shortest path from the gateway to the forged location, the additional delay is divided across links in the shortest path.

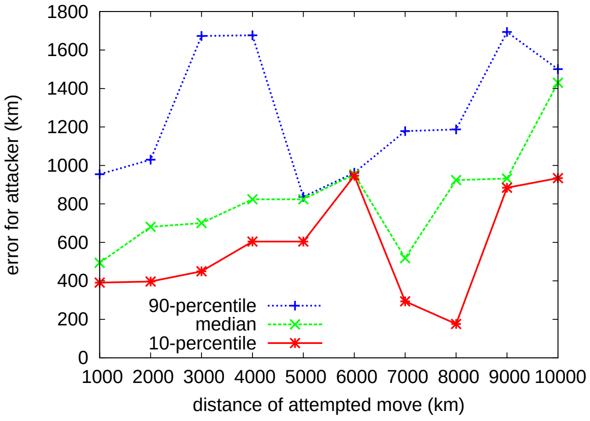

Figure 14: Error observed by the adversary depending on how far they attempt to move the target using the topology-based attack.

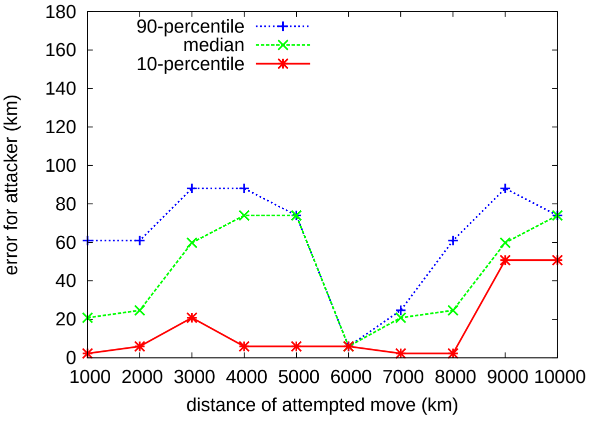

Figure 15: Error observed by the adversary depending on how far they attempt to move the target using the *undns* attack.

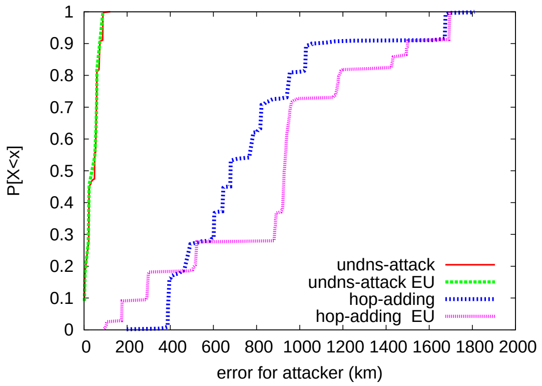

Figure 13: CDF of error distance for the attacker when executing the topology-based and *undns* attacks.

5.3.1 Attack effectiveness

We begin by examining how accurate the adversary can be when attempting to move the target to a specific forged location. Figure 13 shows the error for the adversary when executing the topology-based attack and *undns* extension. Without the *undns* extension, the adversary is able to place a North American target within 680 km of the false location 50% of the time. This is similar to the delay-adding attack in which the adversary has access to the best line function. When moving a target from Europe to North America, the adversary's median error increases by 50% to 929 km. Despite this increase, we observe that the adversary succeeds in each attempt to move a European target into the US. In addition to the overall decrease in accuracy for the adversary, we note that there are some instances where the target in Europe misleads the algorithm with higher accuracy. This is caused by the adversary using the speed-of-light approximation for latencies within their network. Since the speed-of-light is the lower bound on network delay, when additional delay is added to the links to account for the time it would take a probe to reach the target in Europe, the delay approaches the larger delay expected by the landmarks' distance-to-delay mapping. The *undns* extension increases the adversary's accuracy by 93%, with the adversary locating herself within 50 km of the forged location 50% of the time. These results are consistent whether the true location of the target is in North America or Europe.

When analyzing the delay-adding attack, we observed a linear relationship between the distance the adversary attempts to move the target and the error she observes. Figures 14 and 15 show the 10th percentile, median and 90th percentile error for the attacker depending on how far the forged location is from the target for the topology-based attack and *undns* extension, respectively. The observed errors were quite erratic which is a result of the many other factors that affect the accuracy of geolocation beyond the distance of the attempted move. In general, error for the adversary increases slowly as the adversary tries to move the target longer distances. This enables an adversary executing the topology-based attack to move the target longer distances. Error for the adversary using the *undns* extension remains fairly constant regardless of how far they attempt to move the target. In the case of the *undns* attack, the median accuracy fluctuates by less than 60 km whether the adversary moves 500 km or 4,000 km. The slow growth of adversary error stems from the engineered delays in the fictitious network. These delays cause nodes along the paths (including the end point) to

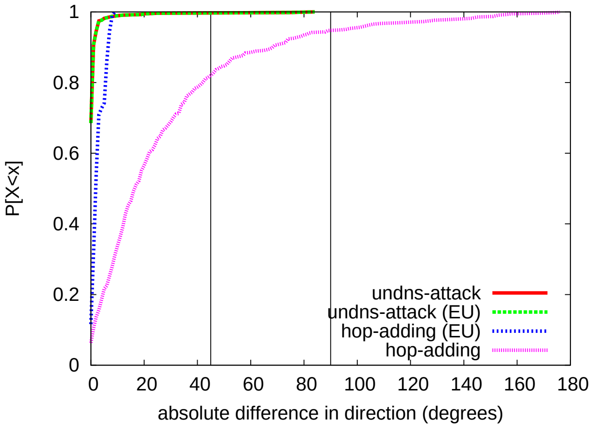

Figure 16: CDF of change in direction for the topology-based and *undns* extension.

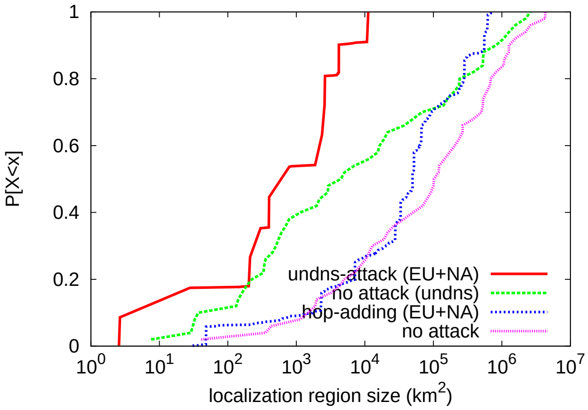

Figure 17: CDF of region size before and after the topology-based attack and *undns* extension.

be geolocated to a similar location regardless of where the target location was originally located.

We next confirm that the adversary is able to move in her chosen direction. Figure 16 shows the difference between the direction the adversary tried to move the target and the direction the target was actually moved (θ in the delay-adding attack). For the general topology-based attack, the adversary is within 36 degrees of her intended direction 75% of the time and within 69 degrees 90% of the time. This improves with the *undns* extension where the adversary is within 3 degrees of their intended direction 95% of the time. When the target attempts to move from Europe to North America, they always move very close to their chosen direction. The adversary always is within 10 degrees of her chosen direction. The smaller change in direction for European nodes stems from the longer distance between the target and the forged location. This causes a smaller change in direction to be observed for similar error values compared to a target that is closer to the forged location.

5.3.2 Attack detectability

We have observed that an adversary executing the topology-based attack and the *undns* extension to the attack can accurately relocate the geolocation target. We next consider whether the victim would be able to detect these attacks and reduce their impacts on geolocation results.

Figure 17 shows the region sizes for topology-aware geolocation and *undns* geolocation before and after the attacks are executed (for both North America and European targets). Unlike the delay-adding attack, the adversary that adds hops to the traceroutes of the victim has region sizes similar to the original algorithms and, in some cases, even smaller region sizes. For topology-aware geolocation, we observe median region sizes of 102,273 km^2 before and 50,441 km^2 after the attack. For the *undns* extension, we observe median region sizes of 4,448 km^2 before and 790 km^2 after the attack. These results indicate that region size is a poor metric for ruling out attacks that add hops to the end of traceroute paths.

Another metric that may be used to rule out geolocation results that have been modified by an adversary is path *circuitousness*. We define circuitousness of a traceroute path between landmark, L_i, and the target as follows, where $r = (\lambda_r, \phi_r)$ is the location returned by the geolocation algorithm, and $h_j = (\lambda_j, \phi_j)$ is the location of intermediate hop j as computed by the geolocation algorithm:

$$C = \frac{d_{ih_0} + \Sigma_{j=1}^{n} d_{h_{j-1}h_j} + d_{h_n r}}{d_{ir}} \qquad (6)$$

Figure 18 shows the distribution of circuitousness for paths between each landmark and the target for topology-aware geolocation before and after the topology-based attack is executed[3]. We observe that when the topology-based attack is executed the circuitousness per landmark increases. One criterion a geolocation algorithm can use for discarding results from the topology-based attack would be to discard results from landmarks where the circuitousness is abnormally high. If a geolocation framework that assigns weights to constraints, such as Octant, is used, constraints from landmarks with high circuitousness could be given a lower weight to limit the adversary's effectiveness. We note that a clever adversary could design her network to use more direct paths, making it more difficult to detect the attack by observing circuitousness.

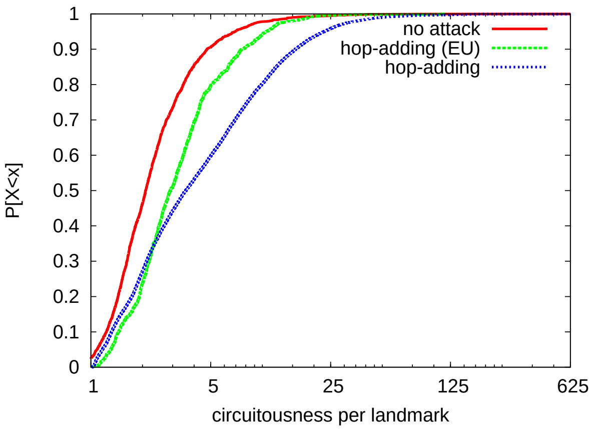

Figure 18: CDF of circuitousness for each landmark before and after the topology-based attack.

6 Related work

While there have been many related works on developing and evaluating geolocation algorithms (e.g., [12, 14, 26, 30]), there has been limited study of IP geolocation given a non-benign target [5, 18].

Castelluccia *et al.* consider the application of CBG [12] to the problem of geolocating hidden servers hosting illegal content within a botnet [5]. The technique used to hide these servers is referred to as “fast-flux”, where a constantly changing set of machines infected by a botnet is used to proxy HTTP messages for a hidden server. Geolocating these servers is important to enable the appropriate authorities to take action against them. Castelluccia *et al.* leverage the fact that the hidden server is behind a layer of proxies to factor out the portion of the observed RTT caused by the proxy layer. They use HTTP connections to measure RTTs (because the hidden servers are unlikely to respond to `ping`) and factor out additional delay caused by the layer of proxies to geolocate hidden servers with a median error of 100 km using PlanetLab nodes as ground truth hidden servers.

Muir and Oorschot survey a variety of geolocation techniques and their applicability in the presence of an adversarial target [18]. Their work is similar to but distinct from ours. Specifically, they emphasize geolocation techniques that leverage secondary sources of information, such as `whois` registries based on domain, IP and AS; DNS LOC [8]; application data from HTTP headers, and data inferred from routing information. They consider delay-based geolocation but do not specify or evaluate any attacks on measurement-based geolocation. Muir and Oorschot discuss the limitations of IP geolocation when an adversary attempts to conceal her IP address through the use of an anonymization proxy and examine how a Web page embedding a Java applet can discover a client’s true identity using Java’s socket class to connect back to the server. They demonstrate this strategy for identifying clients using the Tor [28] anonymization network.

These previous works begin to consider the performance of geolocation algorithms when the target of geolocation may have incentive to be adversarial. However, they generally focus on the issue of geolocating hosts that attempt to deceive geolocation using proxies. In contrast, we develop and evaluate attacks on two classes of measurement-based geolocation techniques by manipulating the network properties on which the techniques rely.

We observe that the problem of geolocating an adversarial target is similar to the problem of secure positioning [4] in the domain of wireless networks. Unlike wireless signals, network delay is subject to additive noise as a result of congestion and queuing along the network path as well as circuitous routes. Multiple hops along network paths on the Internet and the existence of large organizational WANs also enable new adversarial models in the domain of IP geolocation.

7 Conclusions

Many applications of geolocation benefit from security guarantees when confronted with an adversarial target. These include popular applications, such as limiting media distribution to a specific region, fraud detection, and newer applications, such as ensuring regional regulatory compliance when using an infrastructure as a service provider. This paper considered two models of an adversary trying to mislead measurement-based geolocation techniques that leverage end-to-end delays and topology information. To this end, we developed and evaluated two attacks against delay-based and topology-aware geolocation.

To avoid detection, adversaries can leverage inherent variability in network delay and circuitousness of network paths on the Internet to hide their tampering. Since these properties are measured and used by various geolocation techniques, they serve as good attack vectors by which the adversary can influence the geolocation result.

Our most surprising finding is that the more advanced and accurate topology-aware geolocation techniques are more susceptible to covert tampering than the simpler delay-based techniques. For geolocation algorithms that leverage delay, we observed how a simple adversary that only adds delay to probes could alter the results of geolocation. However, this adversary has limited precision when attempting to forge a specific location. We also observed a clear trade-off between the amount of delay an adversary added and her detectability, using the region size returned by CBG [12] as a metric for discarding anomalous results.

Compared to delay-based geolocation, topology-aware geolocation fares no better against a simple adversary and worse against a sophisticated one. Topology-aware geolocation uses more information sources, such as traceroute and *undns* , to achieve higher accuracy than delay-based geolocation. Unfortunately, this advantage becomes a weakness against an adversary able to corrupt these sources. A sophisticated adversary that can leverage multiple network entry points (e.g., an infrastructure as a service provider) can cause the geolocation system to return a result as accurate as the best case simple adversary without increasing the resultant region size. When *undns* entries are corrupted, the adversary is able to forge locations with high accuracy without increasing the region sizes – in some cases, even decreasing them.

Our work reveals limitations of current measurement-based geolocation techniques given an adversarial target. To provide secure geolocation, these algorithms must account for the presence of untrustworthy measurements. This may be in the form of heuristics to discount measurements deemed untrustworthy or through the use of secure measurement protocols. We intend to explore these directions in future work.

Acknowledgements

The authors would like to thank the anonymous reviewers and our shepherd, Steven Gribble, for their feedback, which has helped to improve this paper. This work was supported by the Natural Sciences and Engineering Research Council (NSERC) ISSNet and NSERC-CGS funding.

References

[1] Amazon EC2, 2010. http://aws.amazon.com/ec2/.

[2] ANDERSON, M., BANSAL, A., DOCTOR, B., HADJIYIANNIC, G., HERRINGSHAW, C., KARPLUS, E., AND MUNIZ, D. Method and apparatus for estimating a geographic location of a networked entity, June 2004. US Patent number: 6684250.

[3] American Registry for Internet numbers (ARIN), 2010. http://www.arin.net.

[4] CAPKUN, S., AND HUBAUX, J. Secure positioning of wireless devices with application to sensor networks. In *Proceedings of IEEE INFOCOM Conference* (March 2005).

[5] CASTELLUCCIA, C., KAAFAR, M., MANILS, P., AND PERITO, D. Geolocalization of proxied services and its application to fast-flux hidden servers. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference* (November 2009).

[6] CBC. USA Patriot Act comes under fire in B.C. report, October 2004. http://www.cbc.ca/canada/story/2004/10/29/patriotact_bc041029.html.

[7] CROVELLA, M., AND KRISHNAMURTHY, B. *Internet Measurement: Infrastructure, Traffic and Applications*. John Wiley & sons, 2006.

[8] DAVIS, C., VIXIE, P., GOODWIN, T., AND DICKINSON, I. A means for expressing location information in the domain name system. RFC 1876, IETF, Jan. 1996.

[9] ERIKSSON, B., BARFORD, P., SOMMERS, J., AND NOWAK, R. A learning-based approach for IP geolocation. In *Proceedings of the Passive and Active Measurement Workshop* (April 2010).

[10] GARFINKEL, T., PFAFF, B., CHOW, J., ROSENBLUM, M., AND BONEH, D. Terra: A virtual machine-based platform for trusted computing. In *Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP)* (October 2003).

[11] GILL, P., ARLITT, M., LI, Z., AND MAHANTI, A. The flattening Internet topology: Natural evolution, unsightly barnacles or contrived collapse? In *Proceedings of the Passive and Active Measurement Workshop* (April 2008).

[12] GUEYE, B., ZIVIANI, A., CROVELLA, M., AND FDIDA, S. Constraint-based geolocation of Internet hosts. *IEEE/ACM Transactions on Networking 14*, 6 (December 2006).

[13] Hulu - watch your favorites. anytime. for free., 2010. http://www.hulu.com/.

[14] KATZ-BASSET, E., JOHN, J., KRISHNAMURTHY, A., WETHERALL, D., ANDERSON, T., AND CHAWATHE, Y. Towards IP geolocation using delay and topology mesurements. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference* (October 2006).

[15] KUROSE, J., AND ROSS, K. *Computer Networking: A top-down approach featuring the Internet*. Addison-Wesley, 2005.

[16] Maxmind - geolocation and online fraud prevention, 2010. http://www.maxmind.com.

[17] M.CASADO, AND FREEDMAN, M. Peering through the shroud: The effect of edge opacity on IP-based client identification. In *Proceedings of the 4th Symposium on Networked Systems Design and Implementation (NSDI)* (Cambridge, MA, April 2007).

[18] MUIR, J., AND VAN OORSCHOT, P. Internet geolocation: Evasion and counterevasion. *ACM Computing Surveys 42*, 1 (December 2009).

[19] PADMANABHAN, V., AND SUBRAMANIAN, L. An investigation of geographic mapping techniques for Internet hosts. In *Proceedings of ACM SIGCOMM* (August 2001).

[20] Pandora Internet radio, 2010. http://www.pandora.com.

[21] Planetlab, 2010. http://www.planet-lab.org.

[22] Quova – IP geolocation experts, 2010. http://www.quova.com.

[23] Reseaux IP Europeens (RIPE), 2010. http://www.ripe.net.

[24] RISTENPART, T., TROMER, E., SHACHAM, H., AND SAVAGE, S. Hey, you, get off my cloud! exploring information leakage in third-party compute clouds. In *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS 2009)* (November 2009).

[25] SANTOS, N., GUMMADI, K. P., AND RODRIGUES, R. Towards trusted cloud computing. In *Proceedings of the 1st Workshop in Hot Topics in Cloud Computing (HotCloud)* (June 2009).

[26] SIWPERSAD, S., GUEYE, B., AND UHLIG, S. Assessing the geographic resolution of exhaustive tabulation. In *Proceedings of the Passive and Active Measurement Workshop* (April 2008).

[27] SPRING, N., MAHAJAN, R., AND WETHERALL, D. Measuring ISP topologies with Rocketfuel. In *Proceedings of ACM SIGCOMM* (August 2002).

[28] THE TOR PROJECT. Tor: Overview, 2010. http://www.torproject.org/overview.html.en.

[29] TRANCREDI, P., AND MCCLUNG, K. Use case: Restrict access to online bettors, August 2009. `http://www.quova.com/Uses/UseCaseDetail/09-08-31/Restrict_Access_to_Online_Bettors.aspx`.

[30] WONG, B., STOYANOV, I., AND SIRER, E. G. Octant: A comprehensive framework for the geolocalization of Internet hosts. In *Proceedings of the 4th Symposium on Networked Systems Design and Implementation (NSDI)* (Cambridge, MA, April 2007).

[31] YOUNG, I., MARK, B., AND RICHARDS, D. Statistical geolocation of Internet hosts. In *Proceedings of the 18th International Conference on Computer Communications and Networks* (August 2009).

Notes

[1] In reality, the consumer of geolocation information will likely contract out geolocation services from a third party geolocation provider that will maintain landmarks. Given the common goals of these two entities we model them as a single party.

[2] The adversary can assume that the gateway routers are geolocated to their true locations.

[3] We make similar observations for the *undns* attack extension.