Helping TicketMaster: Changing the Economics of Ticket Robots with Geographic Proof-of-Work

Edward Kaiser
Portland State University
edkaiser@cs.pdx.edu

Wu-chang Feng
Portland State University
wuchang@cs.pdx.edu

***Abstract*—When tickets for popular events such as Hannah Montana concerts go on sale online, they sell out almost instantly. Unfortunately, a significant number of them are purchased by world-wide networks of ticket purchasing robots run by scalpers looking to turn a quick profit. Ticket outlets currently employ CAPTCHAs to slow down fully automated purchasing robots. Since the profit associated with scalping tickets is several orders of magnitude larger than the cost associated with paying humans to solve the CAPTCHAs, this approach has been ineffective.**

CAPTCHAs have a fundamental flaw when used to protect online tickets: the cost to solve them using humans is fixed and small. To address this problem, this paper explores a novel alternative based on geographically-driven proof-of-work. The crux of the approach exploits the observation that most legitimate clients are located geographically close to the event. By requiring every client to solve a cryptographic puzzle whose difficulty is based on their distance to the event, ticket purchasing robots must be placed close to each event in order to monopolize the tickets. This requirement significantly increases the cost of operating such networks. Using emulation and simulation, we demonstrate the utility of our approach in tackling the online ticketing problem.

Keywords: Online Ticketing, IP Geolocation, Proof-of-Work

I. Introduction

Event tickets are a $30 billion market with a majority of the revenue coming from online purchases [17]. For a number of reasons, tickets are sold as commodities with fixed prices [12]. One of the biggest problems in selling tickets online is the ability for scalpers to instantly snap up all available tickets so that they can resell them at substantially higher prices [19], [20]. To deter automated ticket purchasing robots, vendors like TicketMaster employ CAPTCHAs [22] like the one shown in Figure 1. Unfortunately, outsourcing CAPTCHAs costs less than a penny per solution while the profit from reselling a ticket is much larger [8]. Robotic networks employ humans to solve CAPTCHAs and routinely purchase the majority of popular event tickets [18].

The key disadvantage of CAPTCHAs in addressing this problem is their inability to adapt the cost for adversaries. Proof-of-work puzzles are an alternative solution that forces clients to commit resources before being allowed access to the server. Since their conception [5], numerous proof-of-work protocols have been proposed [1], [3], [6], [16], [23]. The difficulty of each puzzle can be individually set. Managing the puzzle difficulty is critical to their effectiveness as studies have shown that uniformly applied proof-of-work is inadequate against adversaries with significant resources [13]. In such cases, legitimate clients are penalized at an unacceptable level rendering the resource useless. Differential proof-of-work schemes adaptively issue more difficult puzzles to potential adversaries. While other studies have shown this approach can work [14], developing algorithms for setting puzzle difficulty in a differential fashion is an open challenge.

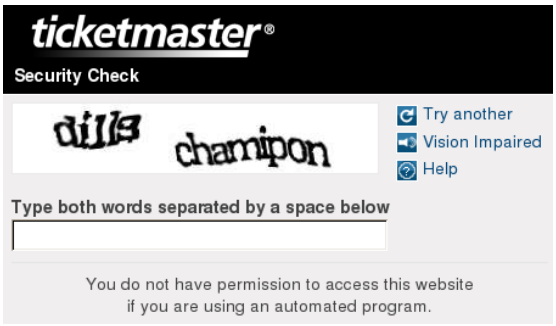

Fig. 1. A TicketMaster CAPTCHA (7/17/09).

Contributions. To tackle the problem of online ticket robots and change the economics for scalpers employing them, this paper explores a web-based proof-of-work mechanism that issues client-specific challenges with difficulty determined as a function of the client's geographic distance from the event. The key observation is that *most legitimate purchases come from clients located in close geographic proximity to the event*. The approach leverages modern IP geolocation databases which are 90% accurate in resolving the geographic location of each client to within 25 miles [7], [15] and adaptively issues distant clients more difficult puzzles. In doing so, operators of ticket purchasing networks are forced to acquire resources in close proximity to each event in order to monopolize event tickets. Unlike previous proof-of-work systems that require changes to end-hosts, protocols, and routers, the approach presented in this paper does not require changes to the software running on either the client or server and can be readily deployed on current online ticketing applications.

While this paper focuses on the online ticketing problem, one fundamental contribution is the notion that *geographic distance may be used as a heuristic of client legitimacy* and could even be applied to other network security problems. For example, online comment spam that prevalently affects articles published by regional news outlets could similarly be mitigated using geographically driven proof-of-work. Additionally, web services with localized content could primarily throttle distant clients when encountering resource consumption attacks.

II. Adversary Model

ADVERSARY GOAL: We assume that legitimate demand for event tickets is sufficient so that all tickets would normally be sold. As a result, the adversary's goal is to *simply acquire as many tickets as possible when they become available for sale*. To simplify the adversary model, we further assume that all the tickets to the event are desirable for resale so the adversary will purchase any and all tickets given the opportunity. As a result, an adversary will always purchase the maximum number of tickets allowed per transaction (usually between 4 and 8 tickets) so from hereon we will use the term "ticket" to really mean the number of tickets allowed per transaction.

GENERAL STRATEGY: Long before tickets go on sale, the adversary establishes control of a botnet. This typically involves stealthily compromising a large number of computers attached to the Internet, or possibly leasing an existing botnet from herders [9]. In terms of network and computation resources, these compromised botnet computers are individually roughly equivalent to the computers used by legitimate clients. In fact, some legitimate client computers may be compromised and unknowingly running botnet software targeting the very same event that the computer's owner is interested in.

Timed to coincide with the start of the ticket sale (i.e., time $t = 0$), the adversary directs the botnet to execute as many ticket purchasing transactions as possible. Since the adversary intends to use the botnet to buyout multiple events or launch other network attacks, the adversary is careful to operate the botnet in a fashion that neither alerts the online ticket vendor of the illegitimate purchase requests nor alerts the true owners of the physical machines as to their misuse.

For any popular event, there is a population of legitimate clients (i.e., dedicated die-hard fans) who also attempt to purchase tickets at the moment they go on sale. To simplify the evaluation of our approach, we assume that these legitimate clients represent equal the number of tickets on sale (i.e., TICKETS $= |C|$) so that the event would sell-out shortly even without the presence of ticket purchasing robots. This allows us to reason that any ticket purchased by an adversary is one that would have otherwise been sold to a legitimate client. In practice, this assumption does not overly weaken the adversary model since adversaries target extremely popular events to minimize the risk of purchasing tickets which they cannot easily resell later at a markup.

EXISTING DEFENSES: Online ticket vendors currently track the network addresses of successful ticket purchasers and restrict each address to one purchase per event. As a result, hosts that are behind network address translating proxies are denied by ticket vendors. This means that any adversary who generates a large number of ticket purchase transactions must have an equivalent number of unique network addresses to successfully complete them. Consequently, this restricts any traffic forwarding and tunneling that an adversary may perform as they must similarly control an equivalent number of forwarders with unique network addresses.

III. Architecture

There are two fundamental components to our approach: the *proof-of-work mechanism* and the *geographic policy* that configures the proof-of-work mechanism.

A. Proof-of-Work Mechanism

Proof-of-work mechanisms consist of three subcomponents: a server-side *issuer* that creates and delivers a puzzle to the client, a client-side *solver* that generates and returns a puzzle solution to the server, and a server-side *verifier* that denies or accepts solutions based on their validity. An obstacle to the deployment of proof-of-work systems is that they require modifications to end hosts, network protocols, or routers. One proof-of-work system that requires few changes is `mod_kaPoW` [11] which is deployed by simply loading an Apache module. The module transparently attaches challenges to URLs within served HTML documents and supplies clients with a JavaScript solver. The module verifies that correct answers accompany all subsequent client requests.

The proof-of-work mechanism in this approach is similar but rather than use an Apache module, the issuer and the verifier are implemented in PHP, a ubiquitous web scripting language. This requires no changes to the web server so it may even be used by websites that cannot load Apache modules. The approach continues to leverage the targeted hash reversal puzzle construction and a periodically updated server secret K to generate client nonces via the block cipher encryption of the client IP address: $E_K(IP_c)$. The server protects the URL to purchase a ticket by specifying the client-specific difficulty D_c so the JavaScript solver must find a solution S such that

$$H(E_K(IP_c) \;||\; URL \;||\; S) \bmod D_c = 0 \qquad (1)$$

where H is a pre-image resistant cryptographic hash function. The solver must perform a brute-force search to find a value for S satisfying the equation. Using a hash function which uniformly distributes its output [2], the probability that any given S satisfies the equation is $\frac{1}{D_c}$, and the number of attempts required to find a valid solution are geometrically distributed with a mean of D_c.

B. Geographic Proof-of-Work Policy

The goal of any proof-of-work mechanism is to maximize the amount of work that adversaries must perform while simultaneously minimizing the work imposed upon legitimate clients. The key observation behind our approach is that most legitimate purchasers of event tickets will do so in close geographic proximity to where the event takes place. Given that commercial geolocation databases which map IP addresses to their geographic location have become very accurate, our hypothesis is that a proof-of-work system whose difficulties are driven by geographic distance can limit scalping by forcing potential purchasers to perform work commensurate to the distance they are away from the actual event. Adversaries must then physically own significant resources near event centers in order to monopolize ticket purchases, thereby making scalping much more costly than simple CAPTCHA outsourcing.

Function	Requests Serviced per Minute
Serve blank PHP page	36,583
Lookup client geolocation	12,462
Lookup client geolocation and issue puzzle	12,444
Lookup client geolocation and verify puzzle	12,412

TABLE I
PROTOTYPE TICKET SERVER THROUGHPUT ACROSS A RANGE OF TASKS.

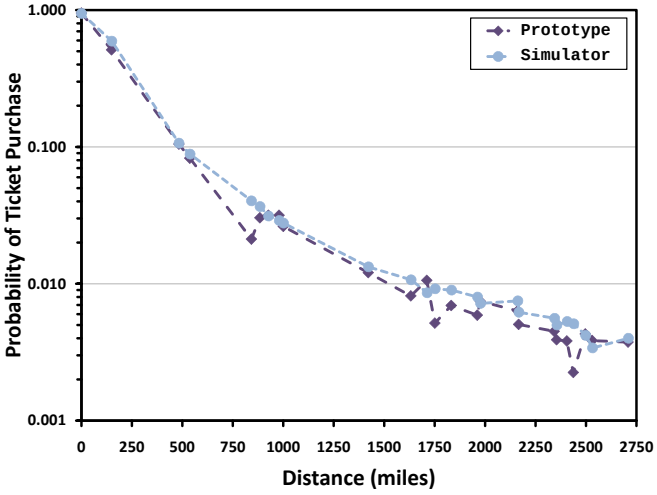

Fig. 2. The probability that prototype and simulator clients may purchase a ticket vs. their distance from the event.

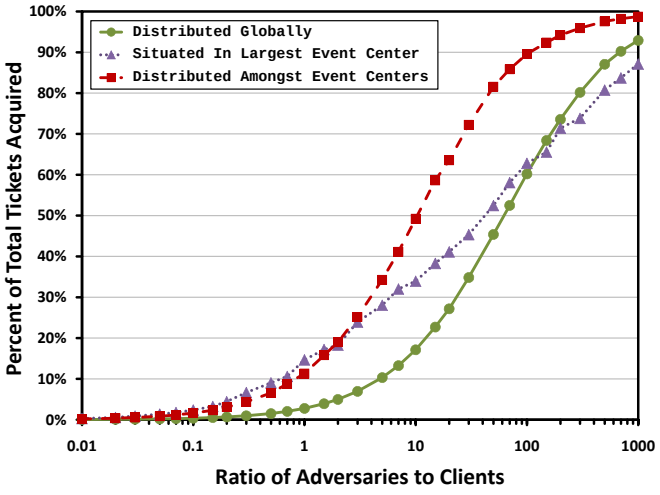

Fig. 3. The percentage of total tickets acquired by adversaries vs. their ratio to clients, using various geographic distributions.

IV. Evaluation

To evaluate the above hypothesis, we leverage accurate commercial geolocation databases [7], [15] to ascertain d_c, the distance of a given client from the event. This distance is then used to set the difficulty D_c of the puzzle that must be solved by that client before being able to purchase a ticket. Since it is unclear how to best set the difficulty, we explore a number of policies and evaluate the ability to thwart a large number of adversaries. Specifically, we aim to maximize the tickets purchased by the legitimate clients C who intend to attend the event and minimize the tickets purchased by the adversaries A attempting to purchase tickets for resale.

A. Prototype

We implemented a prototype that leverages MaxMind's `mod_geoip` [15]. The prototype is publicly accessible [10] and consists of a single PHP script that attaches a challenge to the link for the ticket-purchasing page, validates subsequent solutions, and only allows clients with valid solutions to access the ticket-purchasing page. Table I shows the baseline performance of the prototype on an Intel Core 2 Quad system (Q6600/2.4GHz) running Apache 2.2.9 on Fedora Linux. As the table shows, the server processes over 36,000 blank PHP pages a minute. When IP address resolution is added, the throughput of the system drops by two-thirds due to the overhead of looking up the IP address in the geolocation database. The cost of issuing and validating proof-of-work challenges is negligible compared to that of geolocation resolution. In each case, the performance is more than adequate to support the ticketing application as the capacity of most venues is below the amount of requests the server can process in a minute.

B. Simulator

The prototype above shows how geographic proof-of-work can be easily added to the online ticketing application. To show that it can mitigate realistic networks of ticket-purchasing robots, however, large-scale experimentation using thousands of robots must be performed. Since such experimentation is impractical, we have instead developed a simulator that closely models the behavior of the prototype server and its clients. To validate that the simulator accurately represents the implementation, we compare the results of the following small-scale experiment on the prototype with the identical experiment in the simulator.

The experiment consists of an event in a city on the west coast of the USA for which 100 legitimate clients and 100 adversaries attempt to purchase the 100 available tickets. While the legitimate clients are all located near the city, adversaries are randomly distributed across the 25 largest metropolitan areas in the United States in proportion to the size of each area [21]. As described in Section IV-C1, this distribution maximizes the adversaries' ability to acquire tickets across all events held across the country. Driving the proof-of-work mechanism, the puzzle difficulty is set as $D_c = 100{d_c}^2 + 10^6$. Alternatives are explored in Section IV-C3.

The experiment was performed 10,000 times, both on the prototype and in simulation. Figure 2 shows the probability that clients and adversaries successfully purchase tickets to an event as a function of their distances from the event. As the figure shows, the results from the simulator closely match those from the actual prototype with local clients having an exponentially higher probability of purchasing a ticket than their distant peers.

REGION	POPULATION	EVENTS
New York City, NY	17,799,861	1,756
Los Angeles, CA	11,789,487	1,163
Chicago, IL	8,307,904	819
Philadelphia, PA	5,149,079	508
Miami, FL	4,919,036	487
Dallas, TX	4,145,659	412
Boston, MA	4,032,484	397
Washington, DC	3,933,920	388
Detroit, MI	3,903,377	385
Houston, TX	3,822,509	377
Atlanta, GA	3,499,840	345
San Francisco, CA	2,995,769	295
Phoenix, AZ	2,907,049	286
Seattle, WA	2,712,205	267
San Diego, CA	2,674,436	263
Minneapolis, MN	2,388,593	235
St. Louis, IL	2,077,662	204
Baltimore, MD	2,076,354	201
Tampa, FL	2,062,339	203
Denver, CO	1,984,887	197
Cleveland, OH	1,786,647	173
Pittsburgh, PA	1,753,136	173
Portland, OR	1,583,138	156
San Jose, CA	1,538,312	157
Riverside, CA	1,506,816	154
TOTAL	101,350,499	10,000

TABLE II
THE POPULATION OF THE 25 LARGEST U.S. METROPOLITAN AREAS AND HOW MANY SIMULATED EVENTS OCCUR IN EACH.

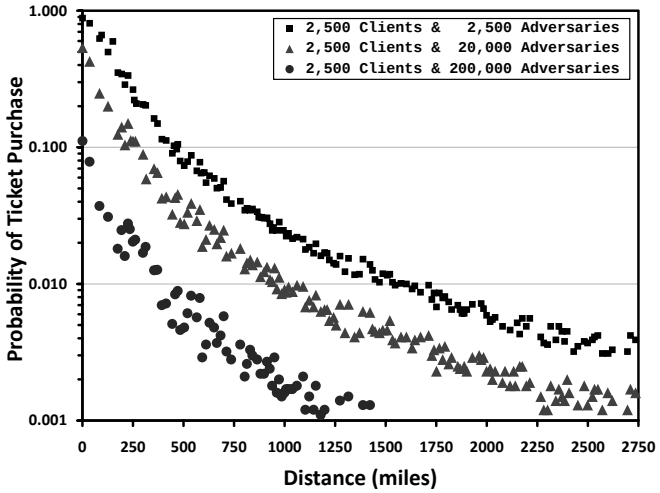

Fig. 4. The probability a client may purchase a ticket vs. their distance from the event, using large legitimate client and adversary populations.

ADVERSARIES	TICKETS ACQUIRED BY		
	C	A_{local}	A_{far}
2,500	88.7%	4.9%	6.4%
20,000	56.2%	23.0%	20.8%
200,000	12.9%	51.0%	36.1%

TABLE III
THE PERCENTAGE OF TOTAL TICKETS ACQUIRED BY THE POPULATIONS EVALUATED IN FIGURE 4. THE CLIENT POPULATION (AND THUS TICKETS) EQUAL 2,500.

C. Adversary Experiments

Similar to real-world ticket outlets, the simulated server sells tickets to events throughout the 25 largest metropolitan areas in the United States with events occurring in proportion to the population of each area. The remainder of this evaluation investigates the ability of an adversary network to purchase tickets to the 10,000 events shown in Table II.

1) Best Adversary Distribution: We first explore geographic distribution strategies that the adversary network might take to maximize its success. In each experiment, an event location is selected and 2,500 local clients attempt to purchase the 2,500 tickets. The adversary population is exponentially increased to see what percent of the total tickets they can purchase. Once again, the difficulty algorithm is $D_c = 100{d_c}^2 + 10^6$.

Figure 3 shows the success of three strategies for distributing adversaries. The first approach assembles adversaries all around the globe like a naïve botnet might. Adversary IP addresses were obtained from the 10,000 worst daily offenders reported by DShield [4]. Not surprisingly, this approach requires orders of magnitude more adversaries than other approaches because many of the bots are far away (i.e., not in North America) from where events are held.

In the second approach, all adversaries are situated in the largest event center: New York City. Acquiring tickets to events in that area is easy, however, acquiring tickets to events held in other areas remains challenging – they must get "lucky" when solving their puzzles to have a chance to purchase tickets before local legitimate clients do.

The third approach distributes adversaries throughout the 25 largest areas in the United States in proportion to their population. This simulates the repeated or long-term leasing (from a botnet controller) of only those zombie machines that are geographically desirable to at least one event location. In this approach, each adversary is local to at least some events and on average 5.96% of the adversaries are local to a randomly selected event. Of the three adversary approaches, this one performs the best, particularly in purchasing the last (i.e., highest) percentile of tickets, and is selected for subsequent experiments.

2) Large Adversary Populations: The previous experiments qualitatively demonstrate the ability for geographic proof-of-work to slow down an adversary. To quantify the extent at which this is the case, we simulate the performance of the system as the number of adversaries is steadily increased. In these experiments, adversaries are distributed across the 25 largest metropolitan areas as before and the difficulty algorithm is again calculated as $D_c = 100{d_c}^2 + 10^6$. Figure 4 shows the ability of individuals to purchase tickets with respect to their distance from the event as the population size of adversaries is changed. As expected, an individual's purchasing ability decreases the further away they are from the event location so local clients stand a much better chance of acquiring tickets. In addition, as the number of total clients is increased, the probability of successfully purchasing a ticket drops across all distances simply because there are more individuals competing for the same finite number of tickets.

As the adversary population is increased significantly versus the legitimate client population, larger numbers of local adversaries A_{local} begin to compete with the legitimate clients. This decreases the percentage of tickets that go to legitimate clients as an increasing percentage of tickets are acquired by adversaries, as shown in Table III. While the adversary network as a whole acquires more tickets across all events, for any specific event, non-local adversaries A_{far} are largely unsuccessful. With increased distance, adversary effectiveness quickly drops off. This is particularly evident in Figure 4 when the 200,000 adversaries outnumber the 2,500 clients (and thus tickets) by a ratio of 80 to 1; adversaries beyond 1,500 miles have less than a 1% chance to acquire tickets. As the adversary population increases, individual local adversaries also have a diminished ability to purchase tickets because they are competing amongst themselves (not just legitimate clients) for the limited tickets.

Throughout the 10,000 events on average 11,872 of the 200,000 adversaries were local to any given event. The local adversaries roughly represent 5.96% of the total adversary population yet account for 58.6% of tickets acquired by the entire adversary population (51.0% of all tickets sold). On average 94.04% (118,128) of adversaries are non-local and manage to purchase only 36.1% of total tickets. The adversary network's success comes at a great cost as 98.9% of the individual adversaries have nothing to show for their arduous proof-of-work computation.

3) Difficulty Algorithms: The prior experiments have used a single difficulty algorithm for determining the amount of work a client must perform as a function of its geographic distance from the server. To examine how sensitive our approach is to this algorithm, we examine a number of alternatives. In comparing algorithms, it is helpful to derive the worst-case and best-case scenarios. The worst case scenario is when the server operates without proof-of-work challenges. Assuming that clients and adversaries arrive at roughly the same time, the percentage of total tickets that the adversaries will be expected to acquire is:

$$without \approx \frac{|A|}{|A| + |C|} \tag{2}$$

Conversely the theoretical best that the system can do using geographically-driven proof-of-work is deny all non-local adversaries so that only local adversaries A_{local} compete with legitimate clients for the tickets. The percentage of tickets they acquire is similarly governed by:

$$theoretical\ best \approx \frac{|A_{local}|}{|A_{local}| + |C|} \tag{3}$$

Figure 5 demonstrates the effectiveness of three different difficulty algorithms on impeding adversaries with respect to the theoretical bounds described above. The algorithms shown are: linear ($D_c = 3000d_c + 10^6$), degree 2 polynomial ($D_c = 100{d_c}^2 + 10^6$), and exponential ($D_c = 1.224^{d_c} + 10^6$). The above theoretical bounds were experimentally tested and are shown in the figure as well.

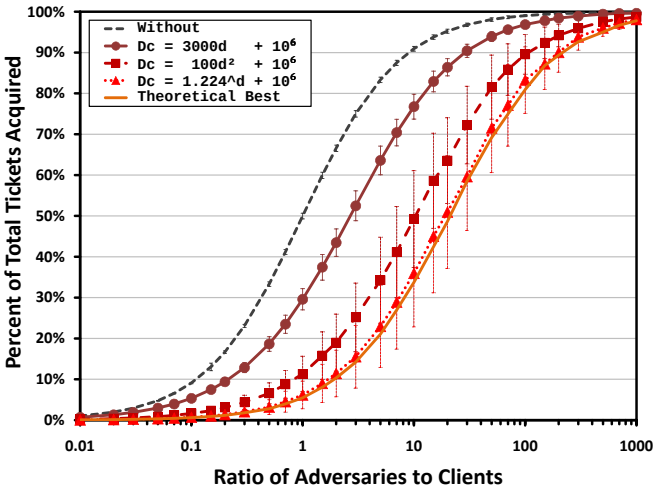

Fig. 5. The percentage of total tickets acquired by adversaries vs. the ratio of adversaries to clients, using various difficulty functions.

The average client delay (in seconds) for these functions closely follows the difficulty divided by the number of hashes computable in one second (i.e., $\frac{D_c}{1{,}000{,}000}$). Thus, for these functions the delay is roughly 1 second for legitimate clients (due to the 10^6 constant) and quickly grows to minutes for distant adversaries. As the figure shows, minimal geographic differentiation is needed to give clients noticeable advantage, yet with slightly more aggressive differentiation the system quickly nears the theoretical best curve. Using the linear difficulty algorithm, remote adversaries are delayed on the order of tens of seconds. In contrast, the polynomial algorithm ramps up the difficulty so that distant adversaries across the country (3,000 miles away) are delayed several minutes. The exponential algorithm is much more severe and delays adversaries further than 100 miles away several minutes. The three algorithms impede adversaries such that the adversaries must multiply their population size by a factor of 2.72, 10.4, and 19.2 (for the respective linear, polynomial, and exponential algorithms) to acquire the same percentage of tickets as a server operating without geographic proof-of-work protection.

The probabilistic nature of puzzle solving means that in some experiments adversaries get "unlucky" and do worse than the theoretical best equation dictates (as evidenced by the error-bars reaching below the theoretical best curve). Conversely, sometimes adversaries get "lucky" when solving their puzzles and thus get more tickets than expected.

V. Discussion

A. Why Geographic Databases?

While geographic proof-of-work increases the monetary cost to adversaries by forcing them to have a presence near each event, there are two problems with using IP-based geolocation databases. The first problem is that non-local and erroneously geolocated legitimate clients will be unfairly penalized. The second problem is that for small events in large event centers, the cost of obtaining sufficient unique local machines to monopolize the event tickets may not be high enough to deter automated ticket purchasing.

It is important that the policy itself adapts to the countermeasures employed by the adversary. A simple modification to the policy would be to use the credit card's geographic billing address when determining the difficulty of the proof-of-work challenge. Clients must already provide authentic credit card information including the billing address in order to purchase tickets. Using that information, the system would have another method for determining where clients are geographically purchasing event tickets from, one which is possibly harder to spoof. This would increase adversary operating costs by forcing them to obtain and maintain a large number of unique local credit cards for every event center targeted.

B. Why Proof-of-Work?

Proof-of-work forces clients to commit their computational resources before they may proceed with the ticket purchasing transaction. One might consider using geographic locations alone without proof-of-work to avoid the client's resource commitment. For example, ticket vendors could alternatively sell tickets probabilistically at different times based on the client's geographic distance to the event. However, those methods lack two of benefits of using proof-of-work.

First, *proof-of-work deters an adversary from using a single machine to launch multiple requests*. If tickets were sold probabilistically based on client distance, an adversary would simply flood the vendor with requests until successful. With proof-of-work, the adversary gains little benefit from flooding requests since the challenge must still be solved before a request is granted. Additionally, proof-of-work prevents an adversary from using a single machine to participate in concurrent ticket purchasing campaigns (or attack other network protocols protected by proof-of-work) since solving simultaneous proof-of-work challenges simply slows down the solution of each rather than provide an advantage.

Second, *proof-of-work increases the likelihood that any individual botnet machine will be discovered and repaired*. Aggressive adversaries using distant machines to purchase tickets will incur steep computational penalties which may make individual machines unresponsive to their real owners. This increases the chance that the owner of the machine will investigate the system degradation and fix it (i.e., remove the zombie software). The risk of detection and removal will thus deter adversaries from targeting ticket vendors protected by proof-of-work. Likewise, adversaries using local zombie machines also increase the risk of being discovered when conflicting with the legitimate owners also attempting to purchase tickets to the event. Since the ticket vendor allows only one transaction per network address, two outcomes are possible. If the legitimate owner completes their transaction first the adversary cannot complete a transaction with that machine. On the other hand, if the zombie completes their transaction first the legitimate owner will get an error message claiming that they have already purchased a ticket to the event increasing the chance that the owner of the machine will discover the zombie software and remove it.

VI. Conclusion

Online ticket outlets currently employ CAPTCHAs to slow down fully automated ticket-purchasing scalper networks. Unfortunately, intelligent adversaries sidestep CAPTCHAs by outsourcing them to humans for less than a penny per solution. This highlights their weakness in protecting the ticketing application: the cost for solving them using humans is small and fixed. This paper presented a novel alternative based on geographically-driven proof-of-work. The approach relies on the observation that most legitimate clients are located in close geographic proximity to an event. Leveraging accurate IP geolocation databases, the system assigns client-specific challenges that are more difficult the further away a client is from the event. A prototype of the system has been implemented in PHP and shown to efficiently and differentially service clients and adversaries. Using an accurate simulator, experiments indicate that an adversary must use up to 19.2 times as many machines to acquire the same percentage of tickets that they would otherwise acquire if the server was unprotected.

References

[1] T. Aura, P. Nikander, and J. Leiwo. DoS-Resistant Authentication with Client Puzzles. In *Workshop on Security Protocols*, April 2000.

[2] M. Bellare and T. Kohno. Hash Function Balance and its Impact on Birthday Attacks. In *EUROCRYPT*, May 2004.

[3] D. Dean and A. Stubblefield. Using Client Puzzles to Protect TLS. In *USENIX Security Symposium*, August 2001.

[4] DShield. Distributed intrusion detection system. http://www.dshield.org.

[5] C. Dwork and M. Naor. Pricing via Processing or Combatting Junk Mail. In *CRYPTO*, August 1992.

[6] W. Feng and E. Kaiser. The Case for Public Work. In *IEEE Global Internet Symposium*, May 2007.

[7] Geobytes, Inc. GeoNetMap. http://www.geobytes.com.

[8] GetAFreelancer.com. Captcha Entry Projects. http://www.getafreelancer.com/projects/by-tag/captcha-entry.html.

[9] N. Ianelli and A. Hackworth. Botnets as a Vehicle for Online Crime, December 2005. CERT RFC 1700.

[10] E. Kaiser and W. Feng. kaPoW Online Ticketing Application. http://kapow.cs.pdx.edu/geotickets.

[11] E. Kaiser and W. Feng. mod_kaPoW: Protecting the Web with Transparent Proof-of-Work. In *IEEE Global Internet Symposium*, April 2008.

[12] P. Krugman. Thinking Outside the Box Office. *Slate*, May 1999. http://slate.msn.com/id/28017/.

[13] B. Laurie and R. Clayton. 'Proof-of-Work' Proves Not to Work'. In *Workshop on Economics and Information Security*, May 2004.

[14] D. Liu and L. Camp. Proof of Work Can Work. In *Workshop on Economics of Information Security*, June 2006.

[15] MaxMind, Inc. Geolocation and Online Fraud Prevention from MaxMind. http://www.maxmind.com.

[16] B. Parno, D. Wendlandt, E. Shi, A. Perrig, B. Maggs, and Y. Hu. Portcullis: Protecting Connection Setup from Denial-of-Capability Attacks. In *ACM SIGCOMM*, August 2007.

[17] B. Siwiki. Big Ticket Items, January 2007. http://www.internetretailer.com/.

[18] R. Stross. Hannah Montana Tickets on Sale! Oops, They're Gone. *New York Times*, December 2007.

[19] StubHub, Inc. Tickets at StubHub! http://www.stubhub.com/.

[20] TNOW Entertainment Group. Tickets at TicketsNow. http://www.ticketsnow.com/.

[21] US Census Bureau. List of Populations of Urbanized Areas, 2000. http://www.census.gov/geo/www/ua/ua2k.txt.

[22] L. von Ahn, M. Blum, N. Hopper, and J. Langford. CAPTCHA: Using Hard AI Problems for Security. In *CRYPTO*, August 2003.

[23] X. Wang and M. Reiter. Mitigating Bandwidth-Exhaustion Attacks Using Congestion Puzzles. In *ACM CCS*, October 2004.