Using Crowdsourcing Marketplaces for Network Measurements: The Case of Spoofer

Qasim Lone*, Matthew Luckie‡, Maciej Korczyński¶, Hadi Asghari*, Mobin Javed† and Michel van Eeten*
Delft University of Technology*, University of Waikato‡, Lahore University of Management Sciences†,
Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG¶
Email: q.b.lone@tudelft.nl

***Abstract*—Internet measurement tools are used to make inferences about network policies and practices across the Internet, such as censorship, traffic manipulation, bandwidth, and security measures. Some tools must be run from vantage points within individual networks, so are dependent on volunteer recruitment. A small pool of volunteers limits the impact of these tools. Crowdsourcing marketplaces can potentially recruit workers to run tools from networks not covered by the volunteer pool.**

We design an infrastructure to collect and synchronize measurements from five crowdsourcing platforms, and use that infrastructure to collect data on network source address validation policies for CAIDA's Spoofer project. In six weeks we increased the coverage of Spoofer measurements by recruiting 1519 workers from within 91 countries and 784 unique ASes for 2,000 Euro; 342 of these ASes were not previously covered, and represent a 15% increase in ASes over the prior 12 months. We describe lessons learned in recruiting and renumerating workers; in particular, strategies to address worker behavior when workers are screened because of overlap in the volunteer pool.

I. Introduction

A wealth of tools have been developed to collect data on network policies and practices across the Internet – e.g., for quality, security, and transparency purposes. Many measurements rely on a distributed set of vantage points to capture representative data. This is even more critical for tools that need to be run from *within* a network to enable correct inferences, such as censorship measurement [27], network performance debugging [10], [28], [35], [43], or detecting security policies such as deployment of Source Address Validation (SAV) [19].

A common challenge is acquiring an adequate set of vantage points. A conventional approach is recruiting volunteers via conferences, mailing lists, and other channels to deploy a tool or hardware probe [14]. Another approach is to use established distributed measurement platforms with a substantial number of vantage points, such as RIPE Atlas [14] and SamKnows [15]. As of October 2017, RIPE Atlas has 10,113 connected vantage points within 3,596 ASes routing IPv4 prefixes. However, these platforms only allow a limited set of measurement tools under their user agreements. For example, while there is demand among RIPE Atlas probe hosts for SAV testing (the case we examine in this paper), and operators of 117 probes within 83 IPv4 ASes have voluntarily tagged their probes asking for this testing, SAV testing on Atlas is unlikely to be supported, at least in the near future [32]. Other platforms, like PlanetLab [11], have most of their vantage points in educational networks, or have few vantage points to begin with. Project BISmark [42], for example, has only 57 active vantage points. Such limitations greatly reduce the types of networks that can be included in a study, especially for measurements that need to be conducted from within networks.

Crowdsourcing marketplaces offer an attractive complementary option for recruiting vantage points, as payment makes studies less reliant on volunteer recruitment. These platforms offer workers small monetary benefits for carrying out micro jobs that usually do not require extensive knowledge and can be completed within few minutes, and attract workers with diverse backgrounds and geographical locations.

In this paper, we explore how effective crowdsourcing marketplaces are in extending, within a limited budget, the coverage of vantage points for network measurements, compared to the volunteer-based approaches commonly used in network measurements. We design and test a system to conduct parallel measurements across five marketplaces, each with a different geographical reach, and assess the improvement in network coverage. We collect data for CAIDA's Spoofer project [19]. The client tests whether the network in which the vantage point is located filters packets with spoofed source IP addresses, a best practice known as SAV [26]. More comprehensive visibility into SAV compliance is important to incentivize network operators combat IP spoofing and mitigate the associated threats, most notably large-scale distributed denial of service attacks [23], [38].

Spoofer provides a very informative case study, as it is dependent on the coverage of vantage points inside networks. It is well known and has been recruiting volunteers for over a decade. To extend its reach, it cannot turn to platforms like RIPE Atlas, which currently does not allow spoofing measurements [32]. These factors make marketplaces valuable, but the tool also poses hurdles, as workers must be willing to install and run an executable, and such a task must be permitted within the Terms of Service of the platform.

To summarize, our main contributions are as follows:

1) We design an infrastructure to collect and synchronize parallel measurements via multiple marketplaces. Our infrastructure prevents invalid submissions, and can be extended to any measurement tool which reports a proof of completion.
2) We present experiences of how this design interacts with the marketplace platforms during measurement studies.

3) We assess the geographical diversity of the workers willing and able to complete the test, both between and within the platforms. We measure the effect of price elasticity (higher compensation) on the recruitment of additional workers. In total, we acquired vantage points from 91 countries and 784 unique Autonomous Systems (AS) and 1519 IP addresses at a price of approximately 2,000 Euro on platform fees and worker compensation.
4) We show that in six weeks, we increased the coverage of Spoofer by 342 unique ASes and 1470 /24's, a 15% increase of ASes over the prior 12 months.
5) We make our code available to the community [6].

II. Related Work

Numerous papers used crowdsourcing platforms from diverse fields such as behavioral sciences, automation [34], [39], and computer vision [22], [41]. Researchers have also explored the dynamics of microjob platforms, and estimated the worker demographics and geographical dispersion [40]. Furthermore, studies have looked at increasing experiment efficiency in terms of price or new users [25], [29], [33].

Closer to our work, there is a handful of studies in the area of information security. Christin *et al.* were able to hire 965 workers to execute their program for an hour [24]. The program collected the Windows version, the list of active processes, and detected whether the application was running in a virtual machine. The goal was to test if raising the price has an impact on participants willingness to execute potentially malicious applications. They observed that significantly more people downloaded the program when the price was raised to $0.50 and then $1.00. In another study, researchers were able to identify 85% of browsers running plug-ins with known vulnerabilities using JavaScript [31]. They concluded that for a mere $52, 1,000 machines could be compromised.

Huz *et al.* conducted two Internet measurements on the MTurk platform, acquiring additional vantage points for broadband speed tests and the state of IPv6 adoption [30]. They found that participants from the US and India constituted 89% of completed tasks. The campaigns were shorter than ours and only on MTurk. Their exploration of pricing effects had inconclusive results. They were also unable to conduct tests using an executable, as this was against the terms of service at the time. Similarly, Varvello *et al.* studied page load times recruiting 1000 paid participants [45]. This study accepted all workers and did not control for, nor optimize, the distribution of vantage points over networks.

Some experiments require workers to conduct subjective assessments, relying on the worker actively participating in the experiment. Mok *et al.* proposed a method to detect low-quality workers that reduce experiment quality in a Quality of Experience context [36]. We do not face the same challenges in this work; the spoofer system automatically evaluates the reliability of the host for conducting SAV measurements.

We build on prior work, most notably [30], by designing an infrastructure to control and optimizing network coverage across platforms, by comparing platforms with different geographical coverage, by running measurements using an executable, and by more systematically observing the impact of job pricing.

III. Background on the Spoofer Project

Determining if a given network blocks packets with spoofed source addresses requires a system within that network try sending packets with spoofed source addresses. The Spoofer project began in March 2005 as an effort by Beverly *et al.* to understand the prevalence of SAV deployment in the Internet using crowd-sourced measurements. They built a client/server system that allows the client to test whether or not packets with spoofed source addresses are discarded before they reach the server. For their initial study [19], they solicited volunteers through the North American Network Operators Group (NANOG) and dhsield security mailing lists to install and run the client. They received 459 client reports from unique IP addresses within 302 different prefixes; the server received packets with spoofed source addresses from 24.2% of these prefixes [19].

Between 2005 and 2009, the client-server system was updated to include a simple GUI for MacOS, IPv6 probing for UNIX systems, multiple destination support and traceroute probing to provide for tomography on paths where SAV is not deployed [21], and tracefilter to find where SAV is deployed [20]. However, there were three key issues limiting volunteer adoption and use of the system: (1) the lack of a user interface to the client software, (2) the user had to manually run the client software, and (3) the results were not made public so ISPs were not incentivized to deploy filtering. Figure 1 summarizes the data collection and project results over time; the peak in May 2006 coincides with a post to Slashdot seeking volunteers to run measurements [16].

In May 2015, CAIDA took over stewardship of the spoofer project, and in May 2016 released a new system that included a GUI and feature parity across all supported platforms (MacOS, Windows, and UNIX). The client operates in the background, testing networks as the volunteer's computer is attached to them, and once a week thereafter. CAIDA built a public reporting engine providing an anonymized view of results, allowing affected IPv4 /24 and IPv6 /40 blocks to be identified, reported with the origin AS of the block and IP geolocation. Raw IPv4 and IPv6 addresses of the tester are kept in a database, and are only disclosed to the affected network if the user consents to the raw IP addresses being shared for remediation, and the operator requires them to remediate. The client software deliberately does not include any tracking capability that would allow CAIDA to determine if tests conducted in different networks are from the same volunteer.

The crowdsourcing measurements we report in this paper contributed to the current peak volume of measurements received by the spoofer project in a single month (middle panel of figure 1). The measurements are, in spoofability, qualitatively similar to other measurements collected between November 2016 and December 2017, i.e. these measurements

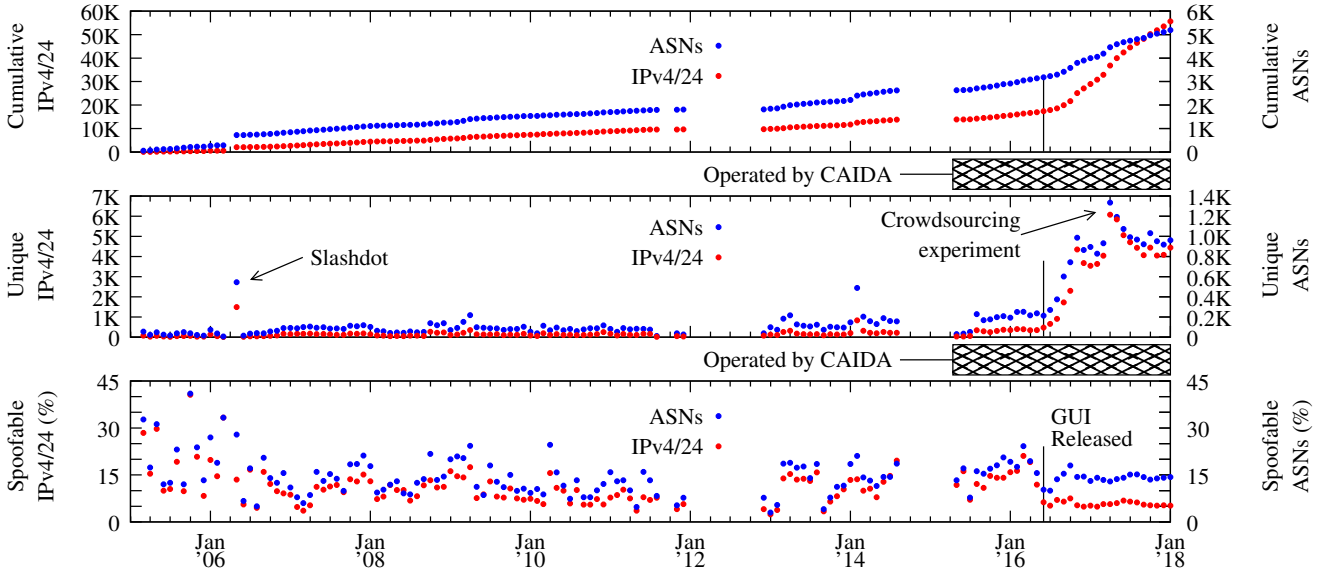

Fig. 1: Overview of Spoofer project data collection over time, aggregated per month. The gaps are due to hardware failures. Between November 2016 and December 2017, the range of spoofable IPv4 prefixes is 4.9% – 6.8%, and the range of spoofable ASNs is 13.1% – 14.5%. The two data collection peaks in April and May 2017 are due to the crowdsourcing experiments in this paper, and those results are qualitatively similar to those collected between November 2016 and December 2017.

Platform	Claimed Coverage	Claimed Population	Min Amount	Payment
MTurk	US, IN	500,000	No min	Credit Card
ProA	GB, US, EU	56,556	$7.50 USD/hr	Paypal
RW	IN, BD, US	N/A	$0.01 USD	Skrill, Paypal
Jobboy	US, BD	152,000	$0.01 USD	Paypal, Payza
Minijobz	BD, IN	N/A	$0.01 USD	Paypal, Payza

TABLE I: Crowdsourcing marketplaces we used in this study. The source of these demographics is various blog posts and platform websites, discussed in section IV. US: United States, IN: India,EU: Europe, GB: Great Britain, BD: Bangladesh.

are no more biased in that dimension than other measurements collected during this period (bottom panel of figure 1).

IV. Crowdsourcing Platforms

We compiled a list of 15 crowdsourcing platforms from prior research and blog posts [1], [37], [44]. First, we selected platforms that allowed tasks which require workers to install and run an executable on their machine, ruling out platforms like `CrowdFlower` [5]. We also excluded platforms where language barriers prevented us from determining whether running executables were allowed (e.g., `zbj.com` and `crowdworks.jp`). Second, the marketplace should support micro jobs. Platforms like `CloudFactory` [4] and `Upwork` [17] only support more complex jobs and impose higher minimum compensation levels.

Based on these requirements, we selected the following five platforms: Amazon Mechanical Turk (MTurk) [2], Prolific Academic (ProA) [12], RapidWorkers (RW) [13], Jobboy (JB) [8], and Minijobz (MJ) [9]. Table I lists features of the selected platforms. They provide diversity of coverage across Europe, the United States, and South Asia (India and Bangladesh), are flexible in setting compensation levels, and offer secure payment methods.

V. Infrastructure Design

Using marketplaces for network measurements is not trivial, as these platforms were not envisioned to support this use case. Screening of workers is based on worker demographics rather than properties of the network or client machines. Furthermore, tasks are generally integrated into the platform. Support for tracking completion of external tasks (e.g., running tools) is not directly available. In this section, we discuss how we tackle these challenges and design a measurement infrastructure to collect network measurement data.

A. Measurement Goal

We articulate our measurement goal as follows: given a limited budget of 2,000 euro, maximize the coverage of vantage points (workers) over networks. After estimating worker payouts, platform overhead, and unforeseen costs at 2 euro per worker, we estimated we could acquire data from 1,000 vantage points (VPs). In total, we obtained data from 1,519 VPs, which we discuss in §VII and §VIII.

Next, we consider how to distribute these points across the IP address space to optimize diversity across networks. One starting point is to seek one data point per Autonomous System. This might be too restrictive for very large ASes, which may have substantial internal heterogeneity. For large ASes, we allow one measurement per each /11 subnet. We chose the granularity of /11 based on two observations: (1) we expect most workers on the platforms to be located in

Platform	Job Posting	Worker proof our website	Worker proof Job website	View Submission	Payment
MTurk	iframe	-	URL	API	API
ProA	iframe	URL + ID	-	CSV	CSV
RW	link	URL	Validation code	Web UI	Manual
MJ	link	URL	Validation code	Web UI	Manual
JB	link	URL	Validation code	Web UI	Manual

TABLE II: Interactions between the microjob platforms and our infrastructure.

broadband networks, and (2) we know these networks collectively represent around 2.4 billion addresses [18]). When distributing 1,000 vantage points across this space, the closest block aggregation is /11. Note that this granularity can be changed based on a study's budget and objectives.

B. Measurement Infrastructure

Researchers may need to screen out workers from network blocks where they already have a vantage point. We therefore determined the eligibility of workers interested in our task and selected them accordingly. We discuss our measurement infrastructure and how we integrate this design consideration.

(i) Job posting: All platforms allow linking to an external website in the job posting. For MTurk and ProA, our website was rendered as an `iframe` inside the platform site. We redirected workers for the other platforms to our website with platform name in the URL arguments to record which platform they participated from.

(ii) Screening: When a potential worker visits our website, we check whether we already have a test result for the corresponding network block they connected from. If so, the potential worker is told that they are ineligible. Otherwise, they are presented with instructions and a form to submit the result from running the Spoofer tool.

(iii) Proof of completion and payment: Upon completion, the Spoofer tool generates a URL with a unique session ID. We ask the workers to submit this URL as a proof of completion. For Mturk, the completion URL must be submitted to Mturk instead of our website, because the terms of service require that all worker-submitted data be stored on Amazon servers first. We set up a cron job to download these URLs and the corresponding Mturker IDs to our centralized database. This allowed us to automate payments on Mturk using the provided payment API. For ProA, we requested workers to submit the worker ID and completion URL to our website. For bulk payments, we uploaded the CSV with worker IDs to the platform. For RapidWorkers, Jobboy, and Minijobz, we asked workers to submit the completion URL to the platform, as there is no easy way to extract a worker ID from these platforms, which is necessary for payments. Further, these platforms do not provide an automatic payment method, and we had to manually approve payment for each successful submission.

(iv) Centralized data collection: A centralized database is required to synchronize the results collected from different crowdsourcing platforms in order to screen workers. Because MTurk required us to store data on Amazon servers, and there is a delay before we subsequently copied the data to our centralized database, we might be too late to screen out subsequent submissions from the same worker on MTurk. To avoid this, we used MTurk's qualification criteria: when a given worker accepts our task, we set a qualification criteria on the worker ID that disqualifies them for accepting it again. We reset this flag in new campaigns, so that workers can participate from a different network block, if eligible.

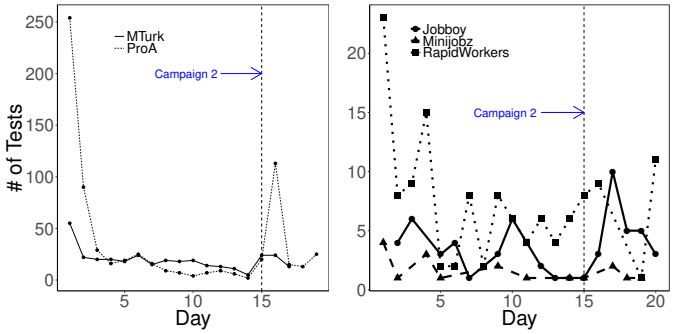

Fig. 2: Job completion for bigger (left) and smaller (right) platforms. When we increased compensation (campaign 2) we attracted additional workers on all platforms.

C. Measurement Campaigns

We ran three subsequent campaigns to evaluate the effectiveness in recruiting vantage points across different networks and to measure price elasticity.

Campaign 1: 50 cents per test. The first campaign lasted two weeks on all platforms. On average, it takes around 4 minutes to download, install, and run the client and to report the completion code to the platform. Offering 50 cents for this time is roughly equivalent to the minimum wage in the Netherlands [3]. Further, Christin *et al.* found that when workers need to install software, raising the compensation to 50 cents caused a dramatic increase in workers [24].

The goals of this experiment were to test our setup and exhaust the pool of workers willing to do the job for 50 cents. We ran this campaign for two weeks, and the completion rate from each platform decreased per day. The last five days brought in only 10% of the results. In total, we received completed submissions from 1,155 workers in 85 countries.

Campaign 2: $1 per test. When few new workers were selecting the job, we increased compensation to $1 to assess price elasticity – i.e., whether higher payment attracted additional workers. The higher compensation was set at the start of day 15. Figure 2 shows that all platforms had an increase in potential workers and completed tasks. RapidWorkers had an outage after we raised the price, so the increase occurred on day 19, when the platform was back online.

We were able to get 364 new submissions from 63 countries after the price increase. Some of these workers will have seen, but not taken up, the task during campaign 1. Of the 364 new submissions, 63 were from IP addresses from which we saw workers viewing, but not selecting, the task during campaign 1. This undercounts the fraction of users who responded

directly to the price increase. Workers can see the title and the compensation level on the task list of the platform, without visiting our page. In other words, a portion of the workers from new IP addresses have also seen the task during campaign 1 and are now responding to the higher price, though we cannot estimate what portion. Combined with the fact that the higher price also brought in more new users than during the last period of campaign 1, we can safely conclude that the price level makes a significant difference in recruiting additional vantage points.

Campaign 3: 10 cent job plus 90 cent bonus. In the final phase, we changed the compensation structure. We ran this campaign as a proof of concept and to resolve the problem of ProA and MTurk worker complaints about compensations (more in section VI). We offered 10 cents to workers for just reading our task. We offered an extra "bonus" to workers who were eligible, to be paid after completing the test. The campaign ran for two days on ProA. 1243 workers participated from which 43 received bonuses. On MTurk, we ran the campaign for a week, 12 workers from a total of 211 participants received bonuses. The low ratio of eligible workers (4-6% compared to 38% for campaigns 1 and 2) combined reflects that eligibility rate goes down over time as more address blocks are already covered. That also makes this pricing structure less efficient, since an increasing fraction of spending will be on workers testing their eligibility rather than actual tests. In our analysis we did not use results from this campaign (§VII,VIII) because it was limited to two platforms (ProA, Amazon) and lasted only for 2 and 7 days respectively.

D. Ethical considerations

Ethical considerations informed the design of our study. The first was fair compensation. One could argue that since microjob platforms are markets, workers can refuse low payouts. Still, due to personal convictions, we did not want to go below the approximate equivalent of the Dutch minimum wage, the location of the majority of this paper's authors. The second consideration was that the measurement tool should not harm worker machines. The Spoofer tool is from a trusted source, does not slow down the machine or the network, is open-source, and can be easily un-installed. Third, we needed to work within the terms of service of the platforms. We only ran our measurements on platforms which allowed software to be downloaded and executed on user machines. Note that previously, Huz *et al.* was not allowed to run the Spoofer tool on MTurk [30], but the terms have since been relaxed to only prohibit software that can be harmful to users. Finally, for privacy considerations we did not ask workers for personal information, which is also forbidden on many platforms. We saved the minimum data necessary to ensure measurement validity: the worker's IP address and user-agent. We saved the worker's IP address to ensure the IP address recorded by the Spoofer project (§ III) corresponded to the IP address used by the worker when selecting our task. Last, to ensure informed consent, we provided clear information about the study.

Classification	MTurk	ProA	Total
Screening	32	118	150
Unclear instruction	5	22	27
Application error	0	32	32
Platform error	4	68	72
Request early payment	9	20	29
Total	50	260	310

TABLE III: Interaction with workers from the ProA and MTurk platforms. Despite having similar numbers of potential workers (Table IV) we had much more interaction with ProA workers, particularly on screening.

E. Interaction with workers

ProA and MTurk provide an option to allow communication between workers and job posters; we resolved all worker questions and complaints. There were only two questions regarding the legitimacy of the Spoofer tool and data being collected. We sent them the prior paper on Spoofer and an example of the data being collected. One user proceeded to test the tool, while the other one did not respond. The breakdown of the rest of the messages received is summarized in Table III.

The majority of comments were about the screening process. Potential workers wanted to know if they would be allowed to run the test in the future and also showed their interest in conducting our study. A few workers demanded to be paid for reading the ineligibility message.

Some workers requested additional help for installing the software. We improved the description of our task based on the feedback we received. A few workers were still unable to run the application, which was mostly due to an incompatible operating system, old hardware, or firewall preventing the installation. We compensated them for their time and effort.

We also received a few messages where workers, after successfully running Spoofer, were not able to upload the results due to some temporary failure of crowdsourcing platform or our server. After verifying their test, we manually entered the result in our database and paid them for the task.

Finally, there were some workers who requested early acceptance of their submissions. We changed our payment process from one time per week to every three days for successful submissions.

F. Follow up tests

The task description included instructions on how to un-install the Spoofer tool after submitting the unique result identifier. Still, the Spoofer project received at least one or more follow-up tests from 433 of the 1519 (28.5%) IP addresses that the workers tested.

VI. Evaluation of design

Our infrastructure met the requirements outlined in section V-B. However, we did encounter several complications along the way, all related to worker behavior.

First, crowdsourcing platforms are designed for screening human subjects, not vantage points. In other words, the platforms offer screening in terms of subject demographics. We

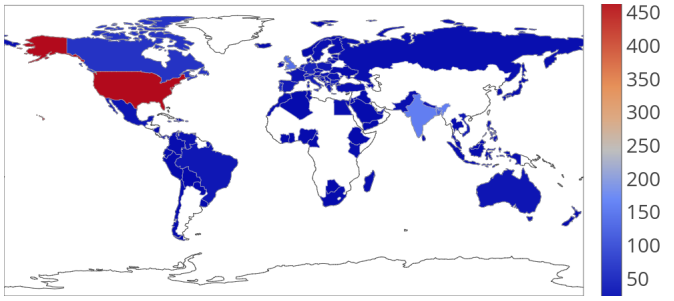

Fig. 3: Location of workers which completed the task. The majority were located in the US and India.

had to implement our own automated screening mechanisms, the result of which then had to be returned to the platform in a platform-specific way for handling task selection and completion. This limitation caused problems on ProA in particular, as the platform allowed participants to mark the task as complete, even if we screened them as not eligible. We ended up with a large number of users who submitted invalid completed tasks. We could have rejected their entries, but cancelling would result in negative scores for these workers. We discussed this issue with ProA staff and ProA cancelled these submissions.

A second issue is that some participants behaved strategically. Some workers on ProA ignored instructions, seemingly consciously, and reported a task as complete, perhaps to see if they would get paid anyway. Due to our requirement that the worker submits a URL with a unique session key that only they can know upon completion of the test, these workers were easily detected. Some workers who were not eligible also sent complaints, arguing that they should be compensated for reading the message that they were not eligible. Interestingly, complaints increased with the higher price of campaign 2. Some workers also complained directly to the platform operator, which led MTurk to suspend one campaign. The automated message cited a violation of their terms of use by collecting Personally Identifiable Information (PII). We think this is because the complaint form on MTurk only offered two options: report a broken task or a privacy violation. In response, campaign 3 tested the compensation model of small payout plus larger bonus, which prevented further complaints.

The final complication is that we could not clearly identify why certain eligible participants did not complete the test. There could be a number of reasons. First, the price of the job might be too low for some workers. Second, they might not like running executables. Third, some workers were using mobile devices for which there is no Spoofer client. Fourth, language barriers may have discouraged some workers. Further research is required on how to improve task uptake.

VII. Analysis of platforms

Coverage of platforms. Our website was visited from 1,978 unique ASes in 142 countries. Even though there is a diversity of networks and countries, we observed that 10 ASes of all potential workers account for 90% of the unique IP addresses. This highlights the need for screening of workers to obtain an effective distribution of vantage points across networks.

Table IV shows the distribution of potential workers for the largest five countries per platform, by unique IP address. The majority of potential workers for MTurk were based in US (49.7%) and India (34.4%), whereas 47% and 29.1% of potential workers in ProA were from UK and US, respectively. RapidWorkers, Jobboy, and Minijobz were more dominant in Bangladesh, India and US.

In terms of the added value of each platform, 29 countries were unique to MTurk, five to only ProA, another five to RapidWorkers, two to Minijobz, and one to Jobboy.

Furthermore, the overlap of ASes between platforms from which workers were interested in completing the task was significant. In the case of smaller platforms (Jobboy, Rapidworkers and Minijobz) the overlap was 75%, 77% and 85%, respectively, when compared to all ASes from which workers visited our website. It was 42% for MTurk and 46% for ProA. However, the overlap in terms of unique /24 networks is relatively small, indicating the significance of choosing prefixes that can be tested by adding multiple platforms. Table VII illustrates pairwise crowdsourcing platform intersections as a matrix, with unique /24 networks from which workers were interested in completing the task. The rightmost column indicates the percentage and absolute number of /24 networks that the platform has in common with all other platforms combined. In the case of ProA we find only 3% of such /24 networks, while it was only 6% for MTurk when compared to all other platforms.

Fluctuations over time. Figure 2 shows the number of new potential workers per day for MTurk stabilized after the initial peak. The number of potential workers from RapidWorkers fluctuates over different days of the week. The number of potential workers increases for Jobboy over time while the pool decreases for ProA.

Completion per platform. Since our design accepts one observation per address block, only 38% of potential workers were eligible to complete the task. Figure 3 shows the countries of the workers who completed the task and ran the Spoofer tool; while we have submissions from 91 countries, the majority of submissions are from US and India.

Table V shows the tests contributed by each platform from respective ASes and countries. The three smaller platforms (RapidWorkers, Minijobz, and Jobboy) added results from 7 countries which were absent from the results from MTurk and ProA. MTurk and ProA added measurements from 12 and 14 countries absent from other platforms, respectively.

Table VI shows the OS distribution of participants of the study along with overall users of spoofer tool. Crowdsourcing platform users seem to be closer to global OS market share [7] when compared to volunteer spoofer users.

VIII. Contributions to Spoofer

What is the added value of the crowdsourcing marketplaces compared to the volunteer pool of Spoofer? Within the study

MTurk			ProA			RW			JB			MJ		
Country	Number		Country	Number		Country	Number		Country	Number		Country	Number	
US	4226	49.7%	GB	3615	47.0%	IN	719	40.6%	BD	634	65.1%	BD	67	20.4%
IN	2925	34.4%	US	2238	29.1%	BD	495	28.0%	US	80	8.2%	IN	53	16.2%
VE	110	1.3%	PT	231	3.0%	US	133	7.5%	IN	40	4.1%	MA	37	11.3%
CA	102	1.2%	CA	194	2.5%	NP	86	4.9%	NP	26	2.7%	US	29	8.9%
GB	78	0.9%	IT	177	2.3%	LK	29	1.6%	EG	12	1.2%	DZ	14	4.3%
Other	1161	13.7%	Other	1231	16.01%	Other	307	17.35	Other	182	18.7%	Other	127	38.9%
Total	8500	100%	Total	7686	100%	Total	1769	100%	Total	974	100%	Total	327	100%

TABLE IV: Number of potential workers interested in performing the study by country code. We report the top 5. US: United States, IN: India, VE: Venezuela, CA: Canada, GB: Great Britain, PT: Portugal, IT: Italy, BD: Bangladesh, NP: Nepal, LK: Sri Lanka, EG: Egypt, MA: Morocco, DZ: Algeria.

Platform	Tests	ASes	Countries
MTurk	424 (27.9%)	255	51
ProA	806 (53.1%)	423	69
RW	165 (10.9%)	134	36
JB	92 (6.1%)	85	24
MJ	32 (2.1%)	24	18
Total(Unique)	1519	784	91

TABLE V: Distribution of workers which completed the Spoofer task. The majority of tasks were completed on the ProA platform.

OS	Crowdsourced	Volunteer
Linux	1.2 % (0.14)	8.1%
MacOS	10.6% (0.52)	20.4%
Windows	88.2% (1.24)	71.1%

TABLE VI: Spoofer client OSes of crowdsourced workers compared to volunteers. The portion of MacOS and Linux users in the crowdsourced population is much less than the volunteer (0.52 and 0.14) population.

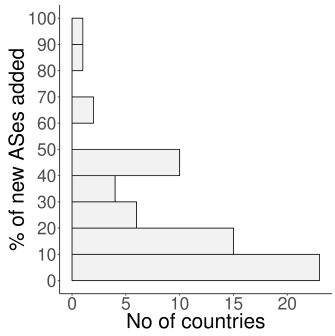

Fig. 4: Percentage of new ASes per country added in Spoofer.

period, we collected data from 1,519 vantage points. While we only allowed eligible workers to complete the task, we did not screen out workers from running the tool from networks that were already present in the volunteer-based Spoofer dataset. The reason is that we wanted to assess the general distribution of workers across networks and how they compared to the volunteer pool that the project has recruited over many years. During our campaigns, on average 12% of daily Spoofer tests came from crowdsourcing platforms. We found 6% overlap in /24 subnets between crowdsourced and volunteer tests.

	MJ	MTurk	RW	ProA	JB	TOT
MJ		12%,38	24%,74	1%,6	16%,52	34%,105
MTurk	0%,38		3%,275	3%,228	0%,43	6%,518
RW	5%,74	20%,275		3%,46	14%,198	36%,489
ProA	0%,6	3%,228	0%,46		0%,15	3%,259
JB	8%,52	7%,43	33%198	2%,15		39%,231

TABLE VII: Pairwise overlap of /24 networks of potential workers of crowdsourcing platforms.

	Mar (2016,-2017) Spoofer Tool		(25 Mar-4 May 2017) CS Platform		Unique CS Platform	
	Total	Spoofable	Total	Spoofable	Total	Spoofable
Countries	143	-	91	-	1	-
ASes	2,237	294	784	66	342	48
/24 blocks	13,081	583	1519	69	1470	69

TABLE VIII: Comparison of spoofer vantage points with crowdsourcing platforms.

Table VIII compares one year of Spoofer volunteer measurements with our 6 weeks of data collection using crowdsourcing platforms. The crowdsourced tests added one country that was missing from a year of Spoofer data: Ivory Coast. One network was found to allow spoofing. For all other countries, the crowdsourced tests increases the coverage of ASes. Figure 4 shows the percentage of additional vantage points we gathered. For instance, in the US, one year of Spoofer measurements collected data from 778 unique ASes, while our much shorter study added tests from 97 additional ASes, 48 of which allowed IP spoofing. Importantly, crowdsourced tests had minimal overlap with the volunteer tests at the level of /24 blocks: only 49 out of 1519 /24 overlapped.

CAIDA notifies operators of networks that do not filter packets with spoofed source addresses. One of the 69 affected networks discovered by our crowdsourcing platform has remediated.

IX. Conclusion

We have presented the first systematic study to deploy multiple crowdsourcing marketplaces to acquire vantage points for Internet measurements. We designed and tested an infrastructure that was able to control the distribution of vantage points. We provide the code of our infrastructure [6].

Using CAIDA's Spoofer tool as a case study, we found that with a limited budget of 2,000 euro, we were able to acquire vantage points in 91 countries and 784 ASNs, 342 of which did

not have a vantage point in the 12 months before our study. The measurements are, in spoofability, qualitatively similar to volunteer-based measurements and they do not introduce additional bias. We find evidence that measurement tasks are quite price sensitive and that higher compensation is likely to recruit even more vantage points.

Crowdsourcing marketplaces provide a realistic and valuable option for recruiting vantage points for Internet measurements. Whether it is the right option for a specific project, depends on several considerations. First, commercially crowdsourced vantage points are relatively costly, especially for longer-term studies. Prolific and Amazon do allow giving bonuses based on worker IDs. If longitudinal measurements are required, workers can be compensated with smaller bonuses per week or month to keep the tool running. Second, if a study seeks a specific set of vantage points outside of its current coverage, then accurately screening workers can make crowdsourcing quite cost effective – almost offering a 'no cure, no pay' approach. Third, one could also see crowdsourcing as a way to to acquire ground truth data for researchers to validate conclusions based on other, cheaper network measurements. Fourth, and final, there seems to be a potential to retain some of the workers as volunteers. Within our study, we found that over one in four workers kept the tool running and submitted unpaid follow-up tests. A project can motivate workers to contribute. For example, the GalaxyZoo project had great success, where 150,000 people participated in a year because they enjoyed the task and they wanted to help advance astronomy.

While crowdsourcing vantage points costs money, important policy efforts, such as the adoption of SAV, should not be wholly dependent on volunteers. Being able to compensate participants in an easy and scalable way is a valuable option to improve our visibility into issues in security, privacy, censorship, and other areas, and designers of measurement systems should consider including built-in payment mechanisms.

Acknowledgments: We are thankful to ProA staff for informative feedback and their help with invalid submissions. This work was partially funded by NCSC NL and by the Department of Homeland Security Science and Technology Directorate, Homeland Security Advanced Research Projects Agency, Cyber Security Division BAA HSHQDC-14-R-B0005, and the Government of United Kingdom of Great Britain and Northern Ireland via contract number D15PC00188.

REFERENCES

[1] www.behind-the-enemy-lines.com/2010/10/explosion-of-micro-crowdsourcing.html.
[2] Amazon Mechanical Turk. https://www.mturk.com/.
[3] Amount of the minimum wage. www.government.nl/topics/minimum-wage/contents/amount-of-the-minimum-wage.
[4] Cloudfactory. https://www.cloudfactory.com/.
[5] Crowd Flower. http://www.crowdflower.com/.
[6] Crowdsourcing Tools. https://github.com/qblone/crowdsourcingTools.
[7] Desktop os market share worldwide. http://gs.statcounter.com/os-market-share/desktop/worldwide/#monthly-201712-201712-bar.
[8] JobBoy. http://www.jobboy.com/.
[9] Minijobz. https://minijobz.com/.
[10] NDT. www.measurementlab.net/tools/ndt/.
[11] Planet Lab. https://www.planet-lab.org/.
[12] Prolific. https://www.prolific.ac/.
[13] Rapidworkers. http://rapidworkers.com/.
[14] RIPE Atlas. https://atlas.ripe.net/.
[15] Samknows. https://www.samknows.com/.
[16] Slashdot: Can you spoof IP packets? https://slashdot.org/story/06/05/02/1729257/can-you-spoof-ip-packets.
[17] upwork. https://www.upwork.com/.
[18] H. Asghari. Cybersecurity via intermediaries: Analyzing security measurements to understand intermediary incentives and inform public policy. chapter 3. 2016.
[19] R. Beverly and S. Bauer. The Spoofer project: Inferring the extent of source address filtering on the Internet. In *Usenix Sruti*, 2005.
[20] R. Beverly and S. Bauer. Tracefilter: A tool for locating network source address validation filters. In *USENIX Security Poster*, 2007.
[21] R. Beverly, A. Berger, Y. Hyun, and k. claffy. Understanding the efficacy of deployed Internet source address validation filtering. IMC '09.
[22] S. Branson, C. Wah, F. Schroff, B. Babenko, P. Welinder, P. Perona, and S. Belongie. Visual recognition with humans in the loop. *ECCV*, 2010.
[23] P. Bright. Spamhaus DDoS grows to Internet-threatening size, Mar. 13.
[24] N. Christin, S. Egelman, T. Vidas, and J. Grossklags. It's all about the benjamins: An empirical study on incentivizing users to ignore security advice. In *FC11*.
[25] P. Dai, Mausam, and D. S. Weld. Decision-theoretic control of crowdsourced workflows. In *AAAI-10*, pages 1168–1174.
[26] P. Ferguson and D. Senie. Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing. RFC 2827.
[27] A. Filasto and J. Appelbaum. Ooni: Open observatory of network interference. In *FOCI*, 2012.
[28] R. Hiran, N. Carlsson, and N. Shahmehri. Crowd-based detection of routing anomalies on the internet. In *Communications and Network Security (CNS), 2015 IEEE Conference on*, pages 388–396. IEEE, 2015.
[29] J. J. Horton and L. B. Chilton. The labor economics of paid crowdsourcing. In *ACM EC*. ACM, 2010.
[30] G. Huz, S. Bauer, R. Beverly, et al. Experience in using mturk for network measurement. In *SIGCOMM C2B(I)D Workshop*, 2015.
[31] C. Kanich, S. Checkoway, and K. Mowery. Putting out a hit: Crowdsourcing malware installs. In *WOOT*, 2011.
[32] D. Karrenberg. [atlas] "Spoofing" tests. https://www.ripe.net/ripe/mail/archives/ripe-atlas/2013-September/001026.html.
[33] A. Kittur, E. H. Chi, and B. Suh. Crowdsourcing user studies with mechanical turk. In *ACM SIGCHI*, 2008.
[34] W. Mason and S. Suri. Conducting behavioral research on amazon's mechanical turk. *Behavior research methods*, 44(1), 2012.
[35] W. Matthews and L. Cottrell. The pinger project: active internet performance monitoring for the henp community. *IEEE Comm. Magazine*, 38(5), 2000.
[36] R. K. P. Mok, R. K. C. Chang, and W. Li. Detecting low-quality workers in QoE crowdtesting: A worker behavior-based approach. *IEEE Transactions on Multimedia*, 19.
[37] E. Peer, S. Samat, L. Brandimarte, and A. Acquisti. Beyond the turk: An empirical comparison of alternative platforms for crowdsourcing online behavioral research.
[38] M. Prince. Technical details behind a 400Gbps NTP amplification DDoS attack, Feb. 2014. http://blog.cloudflare.com/.
[39] M. Restivo and A. van de Rijt. No praise without effort: experimental evidence on how rewards affect wikipedia's contributor community. *ICS*, 17(4), 2014.
[40] J. Ross, L. Irani, M. Silberman, A. Zaldivar, and B. Tomlinson. Who are the crowdworkers?: shifting demographics in mechanical turk. In *CHI'10*.
[41] A. Sorokin and D. Forsyth. Utility data annotation with amazon mechanical turk. In *CVPRW'08*. IEEE, 2008.
[42] S. Sundaresan, S. Burnett, N. Feamster, and W. De Donato. Bismark: A testbed for deploying measurements and applications in broadband access networks. In *USENIX ATC*, 2014.
[43] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, and K. Gibbs. Iperf: The tcp/udp bandwidth measurement tool. 2005.
[44] D. Vakharia and M. Lease. Beyond mechanical turk: An analysis of paid crowd work platforms. In *iConference*, 2015.
[45] M. Varvello, J. Blackburn, D. Naylor, and K. Papagiannaki. Eyeorg: A platform for crowdsourcing web quality of experience measurements. In *CoNEXT'16*.