Beyond Bot Detection: Combating Fraudulent Online Survey Takers*

Ziyi Zhang
University of Wisconsin-Madison
USA

Shuofei Zhu
The Pennsylvania State University
USA

Jaron Mink
University of Illinois at Urbana-Champaign
USA

Aiping Xiong
The Pennsylvania State University
USA

Linhai Song
The Pennsylvania State University
USA

Gang Wang
University of Illinois at Urbana-Champaign
USA

ABSTRACT

Different techniques have been recommended to detect fraudulent responses in online surveys, but little research has been taken to systematically test the extent to which they actually work in practice. In this paper, we conduct an empirical evaluation of 22 anti-fraud tests in two complementary online surveys. The first survey recruits Rust programmers on public online forums and social media networks. We find that fraudulent respondents involve both bot and human characteristics. Among different anti-fraud tests, those designed based on domain knowledge are the most effective. By combining individual tests, we can achieve a detection performance as good as commercial techniques while making the results more explainable. To explore these tests under a broader context, we ran a different survey on Amazon Mechanical Turk (MTurk). The results show that for a generic survey without requiring users to have any domain knowledge, it is more difficult to distinguish fraudulent responses. However, a subset of tests still remain effective.

CCS CONCEPTS

• **Security and privacy → Intrusion/anomaly detection and malware mitigation**.

KEYWORDS

Online Survey; Fraud Detection

ACM Reference Format:
Ziyi Zhang, Shuofei Zhu, Jaron Mink, Aiping Xiong, Linhai Song, and Gang Wang. 2022. Beyond Bot Detection: Combating Fraudulent Online Survey Takers. In *Proceedings of the ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Lyon, France.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3485447.3512230

*This work was supported in part by NSF grants CNS–1955965, CNS–2030521, and CCF–2145394; the NSF Graduate Research Fellowship Program under Grant No. DGE–1746047; and an IST seed grant from Pennsylvania State University.

WWW '22, April 25–29, 2022, Lyon, France.

ACM ISBN 978-1-4503-9096-5/22/04…$15.00
https://doi.org/10.1145/3485447.3512230

1 INTRODUCTION

Surveys, as a form of user study, are widely used by psychologists, sociologists, and also computer scientists to understand humans' attitudes and analyze their behaviors [42]. Today, many surveys are conducted on the Internet (i.e., online surveys) for the purposes of quickly accessing a large number of qualified participants and significantly reducing the overhead of data collection [9, 16].

Despite such benefits, online surveys are conducted over the open Internet where enforcing controls is difficult. Inevitably, online surveys can be easily overwhelmed by fraudulent responses, including responses submitted from ineligible or inattentive participants [8, 54], and those automatically generated by bots [25, 43]. Fraudulent responses not only waste surveys' compensation but also pollute the survey data and can even distort the surveyors' conclusions [14, 15]. As such, systematic methods are needed to detect fraudulent responses to ensure online surveys' quality.

Existing techniques against fraudulent responses can be roughly categorized into up-front tests and post-hoc tests. Up-front tests (e.g., CAPTCHAs [50, 51]) aim to differentiate automated bots from human beings and prevent bots from submitting responses. Post-hoc tests inspect collected responses and filter out redundant responses submitted from the same participants [25] and low-quality responses from inattentive or ineligible participants [29, 47, 48].

Unfortunately, today's bots are increasingly sophisticated. For example, using deep learning models, there is a possibility for bots to solve difficult CAPTCHAs [44], and even perform Natural Language Processing (NLP) to parse and answer questions [11, 40]. In the meantime, human behaviors on the Internet are also dynamically changing. For example, with the increasing awareness of privacy, web users may use privacy-preserving tools (e.g., private browsing [22], Tor [39], VPNs [12]), which can make it challenging to verify users' identity/authenticity during online surveys. To these ends, it is unclear whether existing techniques are still sufficient to ensure survey data's quality, and there is an urgent need to perform an empirical assessment.

In this paper, we systematically evaluate 22 individual tests for fraudulent responses. Some of the tests are provided by commercial tools [18, 31, 32, 36, 49], while others are designed by us through referring to previous literature [4, 33, 35]. We perform two surveys on Qualtrics to examine those tests. The first survey recruits participants of particular expertise (i.e., Rust programming) on public

forums (the *Rust survey*), while the second survey recruits participants on Amazon Mechanical Turks (MTurk) [37] (the *MTurk survey*) and does not pose any particular requirements on participants. The two surveys complement each other well by covering the two most popular recruiting methods of online surveys and two types of background requirements on participants. In the end, we collect 289 and 162 complete responses for the two surveys respectively. We differentiate valid responses that either provide meaningful information or genuinely attempt to do so from invalid ones. We solely rely on answers to open-ended questions to make our labeling independent from the evaluated tests.

For the Rust survey, we use three steps to inspect the 50 valid responses and 239 invalid responses. We first compute the precision and recall for the 22 tests to understand their effectiveness. We then examine our collected behavioral data (e.g., response time) to figure out the nature of the respondents' activities (e.g., automated bot behaviors, human interventions). In the end, we combine multiple individual tests to construct ensemble tests and push the limits of fraudulent response detection. We find that 1) tests based on domain knowledge achieve the best effectiveness; 2) both attention checks and consistency checks need to be better designed to avoid potential false alarms; 3) combining simple technical tests can achieve a detection performance as good as commercial techniques and also generate more explainable results; and 4) the activities of fraudulent respondents involve both bot and human characteristics.

The MTurk survey targets general Internet users (without requiring domain knowledge) and has a different recruiting method. It is designed to validate the findings of the Rust survey. While MTurk's built-in mechanisms (e.g., account registration, worker ID system) help to reduce the participation of bots, we find it is more difficult to distinguish fraudulent human users from automated bots without domain-knowledge based questions. Nevertheless, a subset of tests remain effective, such as browser fingerprints (to detect multiple MTurk accounts controlled by the same person), VPN analysis (to detect people who want bypass location restrictions), and attention and consistency checks.

Overall, our systematic evaluation reveals how fraudulent respondents behave and the effectiveness of different fraud detection tests, all of which can benefit survey designers, data analyzers, fraud detection researchers, and survey platform providers.

In summary, this paper makes the following key contributions:

- We perform a systematic evaluation on a large collection of anti-fraud tests for online surveys and reveal their strengths and weaknesses.
- We conduct head-to-head comparisons on respondents hired through two recruiting methods and demonstrate their characteristics.
- We present ten findings that can benefit future survey designers and survey data analyzers.

2 BACKGROUND AND RELATED WORK

Fraudulent Responses of Online Surveys. Due to the uncontrolled environment of the Internet, online surveys [9] are subject to invalid responses from bots and ineligible and inattentive participants. Bots are computer programs that complete online forms automatically [54]. Ineligible participants are 1) people who do not meet the study requirements but wish to obtain the study's compensation; or 2) eligible people who submit responses multiple times. Inattentive participants may be eligible for the study, but they complete the study quickly without attending to the questions or responding thoughtfully.

Anti-Fraud Techniques. Existing anti-fraud techniques can be applied to detect fraudulent responses for online surveys. These techniques can be classified into *up-front* and *post-hoc* methods.

Up-front methods such as CAPTCHAs [50, 51] can be used to prevent bots from accessing online surveys [54]. However, CAPTCHAs cannot stop ineligible and inattentive participants. Also, recent studies show that deep learning based models can be used to solve (some) CAPTCHAs [2, 44]. In addition, bots can be detected by analyzing browser properties (fingerprinting) [4, 33, 35] or using interactive challenges [55]. To prevent double submissions, existing techniques need to use IP/cookies to uniquely identify a respondent, but fraudulent users/bots can bypass these techniques using VPN services [17]. Finally, to identify eligible participants, screening questions are often used before the survey [1, 5, 7, 14, 15].

Post-hoc measures are applied to the obtained results to filter out invalid responses. In general, those measures can be grouped into three categories: 1) specially crafted survey questions, including attention-check questions [29], consistency-check questions [48], and open-ended questions [47]; 2) analysis of participants' behaviors, such as survey response time [10, 53] and mouse movement [46]; and 3) analysis of computer information, such as IP addresses [6] and browser fingerprints [4, 35].

To our best knowledge, no existing work has systematically evaluated these methods in a holistic setup to empirically compare their effectiveness.

3 METHODOLOGY AND DESIGN

We run an online survey by recruiting Rust programmers on public online forums and social media. Using the survey, we design and evaluate an extensive collection of measures to distinguish fraudulent responses. To examine these measures in a broader context, we then run a second survey on Amazon Mechanical Turk (MTurk) [37]. The two surveys were reviewed and approved by our local IRBs. In the following, we describe the high-level designs of the two surveys, the detailed tests implemented against fraudulent participants, and our methodology to label ground truth.

3.1 Two User Studies

Rust Survey. Our main survey is designed to understand Rust programmers' programming experiences and challenges. Given the need for recruiting participants with certain technical backgrounds (i.e., Rust programming experience), we distribute the survey advertisement on two Rust-related subreddits, two Rust forums, and one author's Twitter account. The survey contains three phrases. In Phase 1, we ask about participants' demographic information (Phase 1.1) and previous Rust programming experience (Phase 1.2). We ask an open-ended question about an important Rust concept at the end of this phase. In Phase 2, we aim to pinpoint the programming challenges of Rust. We ask two open-ended questions in this phase to understand the impact of the information provided by the Rust compiler. We show participants two versions of compiler error

Table 1: Evaluation of Anti-Fraud Tests. Test10 does not include responses without UIDs. Test15 considers both complete responses and incomplete responses that block at the reCAPTCHA test. Test16 analyzes both responses where participants are waiting for the pop-up button and responses where participants click the pop-up button.

Test Information			Rust (Section 4)						MTurk (Section 5)					
ID	Description	Category	TP	FP	TN	FN	Precision	Recall	TP	FP	TN	FN	Precision	Recall
Test1	UID Missing	Redirect Page	23	0	50	216	1.000	0.096	0	0	46	116	–	0.000
Test2	Unexpected Referrer	Redirect Page	83	1	49	156	0.988	0.347	3	4	42	113	0.429	0.026
Test3	Timezone Mismatch	VPN	63	12	38	176	0.840	0.264	65	18	28	51	0.783	0.560
Test4	WebRTC	VPN	189	10	40	50	0.950	0.791	112	38	8	4	0.747	0.966
Test5	VirusTotal (IP)	Activity History	8	2	48	231	0.800	0.033	2	0	46	114	1.000	0.017
Test6	MinFraud (IP)	Activity History	190	8	42	49	0.960	0.795	111	38	8	5	0.745	0.957
Test7	IPRegistry (IP)	Activity History	12	2	48	227	0.857	0.050	0	0	46	116	–	0.000
Test8	DNSBL (IP)	Activity History	180	35	15	59	0.837	0.753	61	21	25	55	0.744	0.526
Test9	RelevantID (Fraud)	Activity History	111	2	48	128	0.982	0.464	18	12	34	98	0.600	0.155
Test10	UID	Duplication	69	0	50	147	1.000	0.319	2	2	44	114	0.500	0.017
Test11	Cookie	Duplication	7	0	50	232	1.000	0.029	2	2	39	100	0.500	0.020
Test12	IPAddress	Duplication	34	0	50	205	1.000	0.142	2	2	44	114	0.500	0.017
Test13	Browser Fingerprint	Duplication	95	0	50	144	1.000	0.397	41	7	34	62	0.854	0.398
Test14	RelevantID (Duplicate)	Duplication	20	2	48	219	0.909	0.084	2	3	43	114	0.400	0.017
Test15	reCAPTCHA	Automation	8	2	50	239	0.800	0.032	0	0	46	116	–	0.000
Test16	Waiting for 5 Seconds	Automation	1	0	52	247	1.000	0.004	3	2	46	116	0.600	0.025
Test17	Resolution	Automation	82	0	50	157	1.000	0.343	1	1	45	115	0.500	0.009
Test18	Attention Check	Psychology	187	21	29	52	0.899	0.782	87	18	28	29	0.829	0.750
Test19	Consistency Check	Psychology	104	4	46	135	0.963	0.435	32	5	41	84	0.865	0.276
Test20	Basic Rust 1	Rust Knowledge	161	1	49	78	0.994	0.674	–	–	–	–	–	–
Test21	Basic Rust 2	Rust Knowledge	195	1	49	44	0.995	0.816	–	–	–	–	–	–
Test22	Rust Safety	Rust Knowledge	206	11	39	33	0.949	0.862	–	–	–	–	–	–
Test23	RelevantID	Ensemble	117	4	46	122	0.967	0.490	20	13	33	96	0.606	0.172
Test24	Test{1, 2, 10, 11, 12, 13, 17}	Ensemble	117	1	49	122	0.992	0.490	44	12	29	58	0.786	0.431
Test25	Test{18, 19}	Ensemble	202	23	27	37	0.898	0.845	94	21	25	22	0.817	0.810
Test26	Test{20, 21, 22}	Ensemble	233	11	39	6	0.955	0.975	–	–	–	–	–	–
Test27	Test{4, 6, 13, 18, 19}	Ensemble	235	30	20	4	0.887	0.983	116	41	5	0	0.739	1.000
Test28	Test{4, 13, 19}	Ensemble	217	13	37	22	0.943	0.908	115	40	6	1	0.742	0.991

messages for the same programming error, and ask participants to explain the root cause of the error in the two open-ended questions. In Phase 3, we ask several post-session questions (e.g., years of programming experience). We pay each participant $10 for their participation.

MTurk Survey. We use the second survey to understand how well our designed anti-fraud tests work under broader contexts. As such, we intentionally make the second survey different from the first one. Our second survey is to examine how users perceive fake social network profiles. In this survey, participants view three social media profiles. Under each profile, participants answer three questions that reflect their trust towards the profile, one question about whether they would accept a connection request from the profile, and one open-ended question to describe the reasons for their decision. After examining three profiles, the participants answer questions about their demographics and social media experience, and one last open-ended question about their strategy for assessing profiles. Since this study does not require participants to have any technical background, we use Amazon MTurk [37] to recruit participants. We recruit MTurk workers from the U.S. region (for English speakers) and do not have other restrictions so as to broadly attract participants. The compensation for each participant is $1.75.

3.2 Anti-Fraud Tests

This section describes our designed tests for distinguishing fraudulent responses from valid responses. Some tests depend on the information provided by Qualtrics, while others rely on our designed survey questions or embedded JavaScript scripts. Below, we use the Rust survey as an example to explain the design of these tests. Similar tests are adapted for the MTurk survey (with minor differences; see Section 5).

Redirect Page. We build a redirect page for the Rust study, which contains the URL to the survey on Qualtrics. We post the URL to the redirect page when recruiting participants. We expect that a valid respondent visits the redirect page first and then participates in the survey. We design the survey procedure this way for three reasons. First, the redirect page shows the paper authors' affiliation, and thus it helps participants confirm the survey advertisement is not a phishing attempt, which is particularly important when recruiting on public forums. Second, the redirect page assigns each respondent a unique identifier (UID), which helps detect responses submitted from the same respondent. Third, we anticipate fraudulent participants may copy the Qualtrics survey link to a bot to automatically generate invalid responses, and inspecting whether the referrer field of a response is the redirect page can help detect such cases. We design two tests (Test1 and Test2 in Table 1) based

on the redirect page by considering a response invalid if it does not have a UID or its referrer is not the redirect page.

VPN Detection. Fraudulent participants may use VPNs or proxies to hide their identities. Thus, we design two tests (Test3 and Test4 in Table 1) to pinpoint possible VPN or proxy usages. Given a response, Test3 compares the timezone extracted from the participant's browser with the timezone inferred by the IP address recorded by Qualtrics. If the two timezones are different, Test3 reports the response as invalid. Test4 compares the IPs leaked by WebRTC [52] with the IPs recorded by Qualtrics. If a response does not have a leaked WebRTC IP or its leaked WebRTC IP is different from its Qualtrics IP, Test4 reports it as invalid.

Activity History. We query four public IP databases [18, 32, 36, 49] to check whether the IPs recorded by Qualtrics were used to conduct malicious activities previously and design four tests accordingly (Test5–8 in Table 1). RelevantID is an online fraud detection service that can identify unique survey participants using IP addresses, geo-locations, and browser information [31]. RelevantID computes fraud scores for survey participants, based on their historical survey behaviors. We enable RelevantID on Qualtrics and build Test9 by considering a response invalid if its fraud score is larger than or equal to 30, a criterion recommended by Qualtrics [41].

Duplication Detection. We design five tests (Test10–14 in Table 1) to detect duplicated responses submitted by the same participant by checking UIDs assigned by the redirect page, IPs recorded by Qualtrics, allocated cookies, browser fingerprints [21] computed by embedded JavaScript scripts, and duplication scores reported by RelevantID, respectively. If one response is detected as a duplicate of another, these tests consider both of the responses as invalid.

Automation Detection. We construct three tests (Test15–17 in Table 1) to pinpoint (or stop) possible survey bots. Test15 and Test16 are based on survey questions. We set up a reCAPTCHA v2 test [23] at the end of the Rust survey. Moreover, we require each participant to wait for five seconds and then click a pop-up button to continue before the reCAPTCHA test. If an incomplete response terminates at the reCAPTCHA test or while waiting for the pop-up button, it is detected as invalid by Test15 or Test16 respectively. We record participants' screen resolutions using embedded JavaScript scripts. If a resolution contains an odd number (e.g., $1364 * 615$), Test17 detects the response as invalid, since the response is likely to be submitted from a bot-used virtual machine.

Attention Check. Attention checks are widely used to exclude inattentive respondents [29]. We design an attention check (Test18 in Table 1 and Figure 3a in the Appendix C) that first asks which industry a participant is in and then explicitly instructs the participant to ignore the question and choose the italicized option. If a response chooses a non-italicized option, we consider it invalid. To prevent the check from being automatically resolved by Natural Language Processing (NLP) techniques, we present the check and all its options in images. In addition, we present the check to each participant in one of the three survey phases with equal probability to examine whether the check's location impacts its effectiveness.

Consistency Check. Test19 in Table 1 is a consistency check. We first ask for participants' occupations in Phase 1 and then ask for their occupations again in Phase 3 with slightly different wording. If a response gives two different answers to the two questions, Test19 detects it as invalid. Figure 3b in Appendix C shows the details.

Rust Knowledge. Domain knowledge can be leveraged to examine whether respondents are qualified [14]. We design three tests based on Rust knowledge (Test20–22 in Table 1). We present a simple Rust program to participants and ask them what the first line and the second line of the output are for Test20 and Test21 (see Figure 3d in Appendix C). We prepare two more complex Rust programs for Test22. One program can be compiled. The other violates Rust's grammar and cannot be compiled. Test22 randomly presents one program to each participant and asks the participant which piece of Rust grammar is violated or whether the program can be compiled (see Figure 3d in Appendix C).

Cognitive Performance Measures. We further compare valid and invalid responses on two cognitive performance measures, response time (RT) [26] and rating-scale responses [38].

We record how much time a participant spends on each individual page on Qualtrics. We can further compute the time spent on each phase and on the whole survey based on the per-page data. Then we can conduct page-, phase-, and survey-level comparisons. Remember that we have different types of questions in different phases (e.g., Phase 1.1 asks for demographic information, Phase 2 poses complex technical questions). Thus, response time in each phase reveals respondents' characteristics as a function of *knowledge type*. Moreover, we design two types of rating-scale questions, 10-point scales (e.g., self-rated Rust expertise) and 21-point scales [27] (e.g., self-rated mental workload), which can help examine the patterns of respondents' answers.

3.3 Response Validity Labeling

We collected responses for the Rust survey from September 22, 2021 to October 5, 2021 and got 289 complete and 420 incomplete responses. We conducted the MTurk survey from October 6, 2021 to October 20, 2021 and got 162 complete and 63 incomplete responses.

We manually review the answers to open-ended questions to decide the responses' ground-truth labels (i.e., valid or not), without referring to the techniques discussed in the previous section. Specifically, we define a response as invalid if it does not provide meaningful information in its answer an open-ended question and does not show a genuine attempt to do so. In other words, it is fine for a valid response to give an imperfect answer, but we expect that the answer is relevant to the question and is not copied from other submissions or sources.

Rust Survey. We design four detailed labeling criteria (see Appendix A) to decide whether a response is valid or invalid. Following these criteria, two paper authors first label each response individually. Then, they discuss their labeling choices and settle down each label through multiple rounds of discussion. Eventually, the two authors identify 50 valid responses and 239 invalid responses among the 289 complete responses. Appendix A provides more details.

MTurk Survey. Two paper authors follow similar (not exactly the same) criteria to examine open-ended question answers in the MTurk survey independently. Then another paper author resolves inconsistent decisions. In total, we identify 46 out of the 162 complete responses as valid and label the remaining 116 as invalid.

4 RESULTS: RUST SURVEY

This section evaluates the anti-fault tests and compares valid and invalid responses on response time and rating-scale responses.

4.1 Observed Useful Tests

Table 1 shows the evaluation results of the tests described in Section 3.2. If a response is labeled as invalid following the methodology in Section 3.3 and is also reported as invalid by a test, we count it as a true positive (TP) for that test. We count or compute other metrics in Table 1 accordingly. Overall, many individual tests (Test1–22) can achieve a high precision (>90%), but none of them have a high recall. However, two ensemble tests (Test26 and Test28) have a high precision and a high recall simultaneously (both >90%).

Redirect Page. As pinpointed by Test1 in Table 1, 23 responses do not have a UID and all of those are invalid responses (TPs). Since the UID is the last component of the URL to our Qualtrics survey, we anticipate fraudulent respondents submit the invalid responses using copied URLs and the URLs' UID parts are not (fully) copied.

Eighty-four responses do not have an HTTP referrer header and thus are reported as invalid by Test2, including 83 true positives and one false positive. There are two possible reasons why the referrer header of a response is empty: 1) the respondent copies the survey URL to a browser to access the survey, instead of directly clicking the URL in the redirect page, and 2) the respondent is privacy-conscious and turns off the referrer feature in her browser, which is probably the reason of the false positive.

VPN Detection. As shown in Table 1, both Test3 and Test4 identify many VPN or proxy usages and also demonstrate that legitimate users may use VPNs or proxies to protect their privacy.

Most (49/67) timezone mismatches pinpointed by Test3 are cases where the browser timezone is different from the IP timezone, but both of them are in the same country (most likely the U.S.). For the remaining 18 mismatches, the browser timezone is in a country (e.g., China) different from the IP timezone (e.g., the U.S.). This result shows that the participants are more likely to use VPNs to hide their identities, but not their nationalities.

Test4 identifies 20 IP mismatches, including 14 invalid responses and 6 valid responses. Moreover, it detects 179 cases where the WebRTC protocol is disabled. Among these cases, 175 are invalid responses and 4 are valid responses. Since the WebRTC protocol is likely to be turned off by data center firewalls [13], we suspect the 179 responses are submitted from virtual machines in data centers.

Activity History. Among the four IP databases, MinFraud reports the largest number of true positives (190) and also has the highest precision (0.960). The IP database on VirusTotal reports the lowest number of true positives (8) and has the lowest precision (0.033). Since a public IP address is likely to be shared by many people and used for malicious activities previously, the four IP databases all report false positives. The fraud detection functionality of RelevantID (Test9) detects 111 invalid responses and only reports two false alarms. Its precision is higher than the four IP databases, which is likely because RelevantID detects fraud in a finer granularity by considering individual participants, instead of considering users behind the same IP as the same.

Duplication Detection. The four duplication detection tests designed by us (Test10–13) all report zero false positives, but they capture different numbers of true positives. Test13 identifies duplicated responses using browser fingerprints and is the most effective among the four tests. Collecting browser fingerprints is too invasive for most surveys to conduct, which is why many fraudulent respondents do not hide their browser fingerprints. Test11 inspects the cookies used to access the survey. It is the least effective test and only pinpoints 10 invalid responses, showing that fraudulent respondents may be aware of clearing their browsing histories before submitting duplicate responses.

The duplication detection of RelevantID (Test14) reports 20 true positives and two false positives. RelevantID identifies distinct survey participants relying on many different factors. The two false positives are likely due to multiple weak factors, instead of a single strong factor (e.g., IP address, browser fingerprint).

Finding 1: *Fraudulent respondents are less likely to hide their browser fingerprints, compared with their IPs and survey cookies.*

Automation Detection. Test15 is the reCAPTCHA test at the end of the survey. We consider the 10 incomplete responses blocked at Test15 as detected or prevented by Test15, count all 289 complete responses as not detected, and ignore the remaining incomplete responses that terminate earlier than the reCAPTCHA test. In total, Test15 successfully prevents 8 invalid survey submissions. When networking conditions are bad, the picture selection stage of reCAPTCHA may not appear, which is probably why Test15 blocks two legitimate participants from submitting responses and reports two false positives.

Test16 requires participants to wait for five seconds to proceed. Similar to Test15, we only consider responses blocked at or passing Test16, and ignore all other responses that stop earlier. Test16 prevents one response, which is indeed an invalid response.

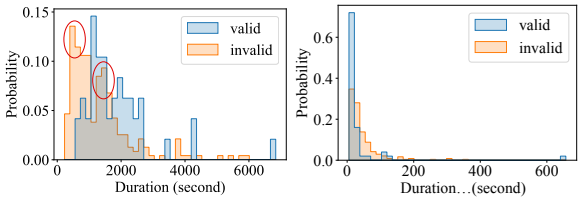
Both Test15 and Test16 can be passed with a certain level of human intervention. They are at the end of our relatively long Rust survey. We suspect that responses reaching the end of the survey involve some human intervention. Thus, most of them (i.e., complete responses) can pass Test15 and Test16.

Test17 detects resolutions unlikely to be used by humans (i.e., resolutions containing odd numbers). It detects 82 invalid responses and reports zero false positives.

Finding 2: *Invalid complete responses can pass the automation prevention (Test15 and Test16) in the Rust survey, and thus they are likely to contain human interventions.*

Attention Check. The attention check (Test18) detects 187 true positives, but unfortunately has 21 false alarms. The correct answer of Test18 is the italicized option "Government", but Test18 also has an incorrect option "Information Technology". 16 out of the 21 false alarms choose "Information Technology". Since our expected survey participants are Rust programmers, it is likely that the participants of the false alarms indeed work in information technology, and they do not pay enough attention to our instructions and simply choose the option that best matches themselves.

As discussed in Section 3.2, each participant randomly takes Test18 in one of the three survey phases. The numbers of participants conducting Test18 in Phases 1, 2, and 3 are 85, 99, and 105 respectively. The proportion of the participants who answer Test18 correctly in Phase 2 (39%) is significantly higher than the proportions in Phase 1 (24%, $p = 0.02$) and in Phase 3 (21%, $p = 0.004$),



(a) Whole Survey (b) Phase 1.1

Figure 1: Response Time Distributions. The two peaks of the bimodal distribution are highlighted in Figure 1a.

which is likely because Phase 2 is meant to test participants' Rust knowledge, so participants are more attentive in this stage.

Consistency Check. Test19 reports 104 true positives and four false positives, both of which are fewer than Test18. For the four false positives, the participants choose "Programmer/Software Engineer" to answer the first question of Test19, which matches our expectations. However, they choose a different option (e.g., "Systems Architect", "Manager") to answer the second question. We anticipate that those participants did not notice we were checking consistency and thought combining two different options together could better explain their job duties.

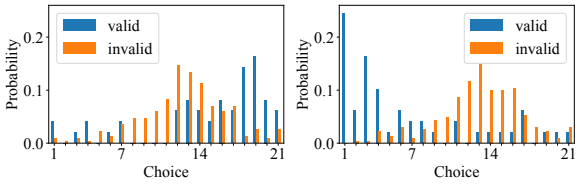
Finding 3: *Respondents who spend enough effort on a survey (e.g., providing meaningful answers for open-ended questions) can still make mistakes when answering attention checks or consistency checks.*

Rust Knowledge. As shown in Table 1, Test20–22 identify 161, 195, and 206 true positives, respectively. The different TP numbers can be explained by the different difficulty levels of the tests. Test20 asks participants to select which letter is the output, while Test21 requires participants to pinpoint which string is the output. Test21 is more difficult and thus detects more invalid responses. Participants can answer Test20 and Test21 using their knowledge of other programming languages. However, Test22 concerns Rust's unique safety rules and requires Rust knowledge. Thus, it captures the most invalid responses. In Test22, we randomly present one of two Rust programs to each participant. We compare the two programs' precision and recall and do not find any significant difference.

Ensemble. We design six ensemble tests to combine the results of individual tests. If one individual test considers a response is invalid, then the ensemble test also detects the response as invalid.

As shown in Table 1, four ensemble tests have a precision larger than 0.9, and two ensemble tests (Test26 and Test28) have a precision and a recall both larger than 0.9. Test26 combines the three Rust-knowledge tests, and its high precision and recall demonstrate the effectiveness of domain knowledge in pinpointing fraudulent responses. Test28 combines two technical tests (Test4 and Test13) and the consistency check. Although both its precision and recall are larger than 0.9, they are smaller than the precision and the recall of Test26. Test24 has a slightly higher precision than Test23 and the same recall as Test23, showing that combining simple technical tests can achieve a detection performance as good as a commercial technique (i.e., RelevantID). However, RelevantID is largely a black box, and it is easier to interpret Test24's detection results.

Finding 4: *Domain knowledge is most effective in detecting fraudulent responses, while combining technical tests with psychological tests can also achieve good results.*



(a) NASA TLX Successful (b) NASA TLX Frustration

Figure 2: Scale Rating Distributions.

Finding 5: *Combining simple technical tests can deliver a fraud detection performance as good as a commercial tool, while reporting more explainable detection results.*

4.2 Cognitive Performance Measures

We measure valid responses and invalid responses in terms of the time spent on each page, each survey phase, and the whole survey, and their choices on all rating-scale questions. Tables 2, 3, and 4 in Appendix B show the details. Due to the limited space, this section only presents critical findings and two case studies.

Response Time. We first check the time spent on each phase and the whole survey. Statistical t tests show that valid respondents spend more time on the whole survey and Phase 2 ($\alpha = .05$). This is expected, since Phase 2 contains difficult technical questions and valid respondents need time to think.

We then examine the response-time histograms. As shown by the blue bars in Figure 1a, the response time on the whole survey is skewed to the right for valid respondents, which is in agreement with a previous study [34]. Interestingly, the distribution of invalid respondents is bimodal (red circles in Figure 1a), suggesting a possible combination of two different respondent groups (e.g., two groups with different levels of automation).

Figure 1b shows the response-time distributions for Phase 1.1. Unlike the previous distribution, the response time of valid respondents is not skewed to the right. One possible reason for this is that the demographic questions in Phase 1.1 are routine questions and thus take minimal effort from valid participants. Instead, invalid respondents show more spread out toward longer response time, indicating some automated bots are used to generate answers, since bots are less sensitive to questions' difficulties.

Finding 6: *Response time distributions can help differentiate valid respondents from invalid respondents, and valid respondents are more sensitive to questions' difficulties.*

Rating-scale Responses. Figures 2a and 2b show the distributions of the successful and frustration measures of the NASA Task Load Index (TLX) [28]. Valid respondents show opposite patterns on the two contrasting questions, revealing an implicit consistency. However, invalid respondents show similar central tendencies on the two figures with scores primarily falling between 11 and 14. A previous study showed that automated bot responses for scale ratings are uniform [10]. Thus, the central tendency indicates a certain amount of human intervention in answering those questions.

Finding 7: *Invalid responses show a central tendency in making rating scale selections, which can potentially help separate them from valid responses.*

Case Studies. We identify 44 invalid responses submitted with the same UID. We carefully inspect these responses to understand whether they are submitted by a bot or a human being. On one hand, the responses have the same screen resolution 1364×615, which is an uncommon resolution for a physical monitor. Moreover, all of them answer open-ended questions with one single word ("secret"), which could be easily synthesized by a bot. In addition, these responses spend a significantly longer time (38.7s) solving the reCAPTCHA test than valid responses (8.9s, $p < 0.001$). These observations demonstrate the characteristics of bot responses.

On the other hand, 20 out of the 44 invalid responses pass the consistency check. Moreover, the survey times of the responses do not overlap with each other. All of them are submitted in two nine-hour time windows. Both time windows start at the same time on two consecutive days. It seems that a human working for nine hours is responsible for the responses.

We notice another invalid response that conducts 36 mouse clicks in 0.001 seconds and performs more than one hundred clicks on five survey pages, which is a clear symptom of bot behaviors. This response uses "na" to answer all open-ended questions. It is detected by the attention check, but it passes the consistency check and the reCPATCHA test.

Finding 8: *Some invalid responses show the characteristics of both bots and human beings, and they are likely to be generated by hybridizing automated bot programs and human intervention.*

5 RESULTS: MTURK SURVEY

Compared to the main study, our second study is more resilient against fraudulent users/bots by using the crowdsourcing platform MTurk. *First*, all MTurk workers need to register accounts at MTurk (verified by email and credit card information). This immediately raises the cost for operating a large number of bots. *Second*, MTurk has its built-in system to track workers' performance/reputation (e.g., number of tasks completed, approval rate) and verify their locations (e.g., for recruiting users from a given country). *Third*, MTurk can enforce that each worker can only take a survey once (based on worker ID). To these ends, we expect the MTurk results to be different from the Rust study. For example, we hypothesize the MTurk survey is more likely to encounter fraudulent *human* workers than automated bots.

We have adjusted a few fault tests for our MTurk survey. First, on the *redirect page*, since each MTurk worker already has her own worker ID, we directly use this ID to detect duplicated participation. Second, for *consistency check*, we adapt the questions by asking about each participant's "most frequently used social network application". Third, we exclude the *knowledge checks* since we do not require the participants to have any domain knowledge.

5.1 Ineffective Tests

As shown in Table 1, we observe that some of the tests are no longer relevant in comparison with the main study.

First, tests on the *redirection page* become irrelevant. The UID (Test1) in this case is the MTurk worker ID, which is always available. Also, we rarely see unexpected referrers (Test2)—the 3 FPs under this test are users who have an "empty" referrer, likely due to the "referrer hiding" setting in their browsers.

Second, *duplication*-based tests are less effective. This is because there is no incentive to take the survey twice (as workers will only get paid once by the MTurk platform). We only find one "valid" worker who has submitted the survey twice (i.e., the 2 FPs for Test10, 11, 12, 14). A closer inspection shows the two submissions have the same worker ID and almost identical answers. We suspect this participant thought the first submission was unsuccessful and thus tried again. Interestingly, the browser fingerprint test (Test13) shows effectiveness, which will be explained later in Section 5.2.

Third, *automation*-based tests (like reCAPTCHA) are also ineffective (Test15–17). This confirms that we are more likely to encounter fraudulent human users than automated bots on MTurk.

Finding 9: *Using the MTurk platform likely reduces bot participation, and thus duplication- and automation-based tests are less effective.*

5.2 Effective Tests

As shown in Table 1, the effective tests seem to be centered around three types of fraudulent participants: (1) those who hide/forge their true residential location, (2) those who control multiple MTurk accounts to take the survey, and (3) those who are not attentive during the survey.

First, *VPN* tests (Test3-4) return a large number of true positives (65, 112) and also some false positives (18, 38). Recall that our survey has been restricted to the U.S. region (to attract English speakers). We find that some participants outside the U.S. have used VPN to bypass MTurk's location restrictions. Using WebRTC, we can only recover the browser IP addresses behind the VPN/proxy addresses for 13 participants. Five of these participants are located in India, and the remaining eight are from the U.S.. The false positives are those who made efforts to answer questions (even though they connected to our survey via VPNs or proxies). Because of the VPN usage, the *IP* related tests (Test5–9) are also affected.

Second, the *fingerprint* test (Test13) under the duplication category returns surprising results. There are 48 (41 TPs +7 FPs) worker IDs that contain duplicate fingerprints (16 unique fingerprints). Based on the worker IDs and unique fingerprints, we group them into 14 groups[1]. On average, we find that each person controls 3.4 accounts. While creating multiple MTurk accounts requires using different email addresses and credit cards, with these accounts, the worker can now take a survey multiple times for extra payments. A further investigation shows that worker IDs within the same group share the same screen size, operating system, and browser (but not always the same browser versions). Also, within a group, submissions do not overlap in *time*, which means the survey is completed *sequentially* using the different worker IDs that the person controls. Most of their answers to open-ended questions are either very short (1–2 words) or clearly copied from Google search results. The "invalid" answers have a high level of duplication across multiple accounts from the same person. Note that seven worker IDs are marked as false positives because they actually made a reasonable effort to answer the questions. However, such duplicated submissions still contaminated the survey results (i.e., can be discarded).

[1]We first group these worker IDs into 16 groups based on unique fingerprints. We then find 3 worker IDs that belong to multiple groups—likely meaning this worker has switched browsers during the survey. After merging, we have 14 groups.

Third, tests under the *psychology* category (Test18 and Test19) are also effective to reflect the attentiveness of participants. The results are consistent with those of the main study.

Finally, like the main study, we compare the survey completion times between valid and invalid participants. Similarly, we find a significant difference between completion times, with invalid participants (14.3±1.1 minutes) completing the survey faster than valid participants (20.4±2.4 minutes, $p = 0.02$).

Finding 10: *Browser fingerprints are still effective to detect multiple accounts controlled by the same users on MTurk; VPN tests can identify participants that try to bypass location checks, but these tests can generate false positives.*

5.3 Summary of MTurk Results

Overall, the need for account registration and MTurk's worker ID system make it costly to run a large number of bots. Instead, the threat model is shifted from bot detection to fraudulent human detection. Our analysis shows that fraudulent users tend to use VPN services to bypass MTurk's location check, and register multiple MTurk accounts to take the survey for additional payments.

Compared to the main study, detecting fraudulent participants in this study is not necessarily easier. This is because the survey is open to general Internet users and thus the highly effective *domain knowledge* questions in the Rust survey are no longer applicable. Also, bot detection tests do not work well on humans. As such, we recommend combining several high-performing tests to jointly filter out invalid responses. As shown in Table 1, the two different ensembles (Test27–28) are quite effective. Combining WebRTC (Test4), MinFraud (Test6), browser fingerprinting (Test13), and psychology tests (Test18–19) achieves a 0.739 precision and a 1.000 recall. As discussed in Section 5.2, the precision is affected by the false positives—many of whom have lied about their location or submitted duplicated answers (using different worker IDs), which makes their answers less valuable.

6 DISCUSSION AND CONCLUSION

Detecting and preventing invalid responses in online surveys is a critical challenge. We systematically evaluate the effectiveness of 22 individual tests in preventing and detecting fraudulent responses across two different online surveys, one requiring domain knowledge published on online forums and the other without expertise requirements on Amazon MTurk. We label the responses based on answers to open-ended questions and evaluate the effectiveness of each test based on precision and recall. To characterize respondents' behaviors, we also examine their cognitive performance measures (e.g., response time). The results indicate that each test itself is not sufficient, yet tests based on domain knowledge show the best performance. We then propose ensembles that combine tests and cognitive performance measures in case studies. The combined tests result in a better performance that is comparable to commercial techniques. Moreover, the complementary evaluation with tests and response time distributions reveals the presence of both bot respondents and invalid human respondents, and both bot behavior and human behavior in individual invalid responses.

Additional information about respondents through measuring their behavior (e.g., response time) sheds light on the characteristics of human cognitive origin [19, 45], but the automated nature of bots [20] requires more nuanced fraudulent detection strategies. For example, researchers could consider a phase-wise survey design, in which each phase consists of questions of the same type (i.e., general or domain-specific). In this case, a post-hoc analysis could be conducted to detect different behavior (bot vs. human) at question-, phase-, and survey-levels.

Our results show that there is no perfect strategy for preventing and detecting invalid respondents for online surveys. Given the limitation of individual tests, it is imperative that researchers use test ensembles or complementary methods. During survey design, candidate methods should be evaluated based on the research goals (e.g., domain knowledge required or not) and the balance of preventing/excluding invalid respondents and deterring valid respondents (e.g., whether to use multiple attention checks).

Limitations. Several threats to internal validity exist. The ground-truth labeling is done by manual coding of open-ended question answers to be independent from studied tests. We could make mistakes in such a process, affecting the results. Also, we do not think this methodology is the best way to gauge the validity of survey responses and recommend considering more factors to filter out invalid responses in real user studies. Moreover, Finding 2 suggests a respondent passing automation detection indicates possible human intervention, but recent studies [30, 44] find automated tools with AI techniques can bypass automation detection products.

There are threats to external validity. Although we explicitly allowed invalid responses, e.g., publishing the survey link on social media and removing MTurk worker restrictions, the invalid respondents in both surveys (Rust: 239; MTurk: 116) may not be representative of the population of invalid respondents. For example, the invalid human respondents for the Rust survey probably have programming backgrounds since we posted the survey mainly on Rust relevant forums. Thus, the test results and cognitive performance measures may reveal the characteristics of respondents with particular expertise. Nonetheless, we believe the two surveys provide a meaningful probe into preventing invalid responses.

Some of the tests use respondents' information through the computer and Internet, e.g., IP addresses. The collection and use of such information preclude the surveys from being anonymous. However, the purpose of our work is to inform effective invalid response detection, which is justified. We also discussed the data collection with each local IRB, and each study received an exemption for IRB review. Finally, we obtained respondents' informed consent at the very beginning of the survey, in which we explicitly describe our research purposes and our data collection.

Adaptive Countermeasures. If adversaries become aware of our tests, they can make adaptations. For instance, both bots and fraudulent humans may use privacy-preserving tools to prevent fingerprinting scripts or produce fake fingerprints [3, 24]. Also, advanced Optical Character Recognition (OCR) models and NLP tools can be used to recognize and process text in images to pass attention checks. As this cat-and-mouse game evolves, we believe that basic anti-fraud techniques are still necessary to filter out low-quality, high-volume bot traffic. In the meantime, future work can look into how to design tests around domain knowledge and human cognition to stay robust against advanced fraudulent entities.

REFERENCES

[1] Yasemin Acar, Michael Backes, Sascha Fahl, Doowon Kim, Michelle Mazurek, and Christian Stransky. 2016. You Get Where You're Looking for: The Impact of Information Sources on Code Security. In *Proceedings of the 37th IEEE Symposium on Security and Privacy (S&P '16)*. San Jose, CA, USA. https://doi.org/10.1109/SP.2016.25

[2] Nasser Mohammed Al-Fannah. 2017. Making defeating captchas harder for bots. In *Proceedings of the 2017 Computing Conference*. London, United Kingdom. https://doi.org/10.1109/SAI.2017.8252183

[3] Babak Amin Azad, Oleksii Starov, Pierre Laperdrix, and Nick Nikiforakis. 2020. Short Paper - Taming the Shape Shifter: Detecting Anti-fingerprinting Browsers. In *Proceedings of the 17th Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA '20)*. Lisboa, Portugal. https://doi.org/10.1007/978-3-030-52683-2_8

[4] Babak Amin Azad, Oleksii Starov, Pierre Laperdrix, and Nick Nikiforakis. 2020. Web Runner 2049: Evaluating Third-Party Anti-bot Services. In *Proceedings of the 17th Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA '20)*. Lisboa, Portugal. https://doi.org/10.1007/978-3-030-52683-2_7

[5] Hala Assal and Sonia Chiasson. 2019. 'Think Secure from the Beginning': A Survey with Software Developers. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Glasgow, United Kingdom. https://doi.org/10.1145/3290605.3300519

[6] Hui Bai. 2018. Evidence that a large amount of low quality responses on MTurk can be detected with repeated GPS coordinates. https://goo.gl/19KCHG. (Accessed on 10/22/2021).

[7] Rebecca Balebako, Abigail Marsh, Jialiu Lin, Jason I. Hong, and Lorrie Cranor. 2018. The Privacy and Security Behaviors of Smartphone App Developers. https://doi.org/10.1184/R1/6470528.v1

[8] Benjamin Birnbaum, Gaetano Borriello, Abraham D. Flaxman, Brian DeRenzi, and Anna R. Karlin. 2013. Using Behavioral Data to Identify Interviewer Fabrication in Surveys. In *Proceedings of the 2013 CHI Conference on Human Factors in Computing Systems (CHI '13)*. Paris, France. https://doi.org/10.1145/2470654.2481404

[9] Michael H Birnbaum. 2004. Human research and data collection via the Internet. *Annual Review of Psychology* 55 (2004), 803–832. https://doi.org/10.1146/annurev.psych.55.090902.141601

[10] Erin M Buchanan and John E Scofield. 2018. Methods to detect low quality data and its implication for psychological research. *Behavior Research Methods* 50, 6 (2018), 2586–2596. https://doi.org/10.3758/s13428-018-1035-6

[11] Hao Cheng, Yelong Shen, Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2021. UnitedQA: A Hybrid Approach for Open Domain Question Answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL/IJCNLP '21)*. Bangkok, Thailand. https://doi.org/10.18653/v1/2021.acl-long.240

[12] Cisco. 2021. What Is a VPN? - Virtual Private Network - Cisco. https://www.cisco.com/c/en/us/products/security/vpn-endpoint-security-clients/what-is-vpn.html. (Accessed on 10/21/2021).

[13] Daniel Cubley. 2019. Microsoft Teams Calls Intermittently Dropping due to Firewall Protocol Inspection. https://www.risual.com/2019/10/microsoft-teams-calls-intermittently-dropping-due-to-firewall-protocol-inspection/. (Accessed on 10/22/2021).

[14] Anastasia Danilova, Alena Naiakshina, Stefan Horstmann, and Matthew Smith. 2021. Do you Really Code? Designing and Evaluating Screening Questions for Online Surveys with Programmers. In *Proceedings of the 43rd International Conference on Software Engineering (ICSE '21)*. Virtual Event, Spain. https://doi.org/10.1109/ICSE43902.2021.00057

[15] Anastasia Danilova, Alena Naiakshina, and Matthew Smith. 2020. One Size Does Not Fit All: A Grounded Theory and Online Survey Study of Developer Preferences for Security Warning Types. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering (ICSE '20)*. Seoul, South Korea. https://doi.org/10.1145/3377811.3380387

[16] Marcel Das, Peter Ester, and Lars Kaczmirek. 2018. *Social and behavioral research and the internet: Advances in applied methods and research strategies*.

[17] Sean A. Dennis, Brian M. Goodson, and Christopher A. Pearson. 2019. Online Worker Fraud and Evolving Threats to the Integrity of MTurk Data: A Discussion of Virtual Private Servers and the Limitations of IP-Based Screening Procedures. *Behavioral Research in Accounting* (2019). https://doi.org/10.2308/bria-18-044

[18] Dmitry. 2021. Pydnsbl: Async dnsbl spam lists checker based on asyncio/aiodns. https://github.com/dmippolitov/pydnsbl. (Accessed on 10/21/2021).

[19] Franciscus Cornelis Donders. 1969. On the speed of mental processes. *Acta Psychologica* 30 (1969), 412–431. https://doi.org/10.1016/0001-6918(69)90065-1

[20] Marc Dupuis, Emanuele Meier, and Félix Cuneo. 2019. Detecting computer-generated random responding in questionnaire-based data: A comparison of seven indices. *Behavior Research Methods* 51, 5 (2019), 2228–2237. https://doi.org/doi.org/10.3758/s13428-018-1103-y

[21] FingerprintJS, Inc. 2021. Fingerprintjs. https://github.com/fingerprintjs/fingerprintjs. (Accessed on 10/21/2021).

[22] Google. 2021. Choose your privacy settings - Computer - Google Chrome Help. https://support.google.com/chrome/answer/114836. (Accessed on 10/21/2021).

[23] Google. 2021. reCAPTCHA. https://www.google.com/recaptcha/about/. (Accessed on 10/21/2021).

[24] Daniel Goßen, Hugo Jonker, Stefan Karsch, Benjamin Krumnow, and David Roefs. 2021. HLISA: Towards a More Reliable Measurement Tool. In *Proceedings of the 21st ACM Internet Measurement Conference (IMC '21)*. Virtual Event, USA. https://doi.org/10.1145/3487552.3487843

[25] Marybec Griffin, Richard J Martino, Caleb LoSchiavo, Camilla Comer-Carruthers, Kristen D Krause, Christopher B Stults, and Perry N Halkitis. 2021. Ensuring survey research data integrity in the era of internet bots. *Quality & Quantity* (2021), 1–12. https://doi.org/10.1007/s11135-021-01252-1

[26] Christopher M Harris, Jonathan Waddington, Valerio Biscione, and Sean Manzi. 2014. Manual choice reaction times in the rate-domain. *Frontiers in Human Neuroscience* 8 (2014), 418. https://doi.org/10.3389/fnhum.2014.00418

[27] Sandra G Hart. 2006. NASA-task load index (NASA-TLX); 20 years later. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*.

[28] Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*. Vol. 52. 139–183. https://doi.org/10.1016/S0166-4115(08)62386-9

[29] David J Hauser and Norbert Schwarz. 2016. Attentive Turkers: MTurk participants perform better on online attention checks than do subject pool participants. *Behavior Research Methods* 48, 1 (2016), 400–407. https://doi.org/10.3758/s13428-015-0578-z

[30] Md Imran Hossen, Yazhou Tu, Md Fazle Rabby, Md Nazmul Islam, Hui Cao, and Xiali Hei. 2020. An object detection based solver for google's image recaptcha v2. In *Proceedings of the 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID '20)*. Virtual Event, Spain.

[31] Imperium. 2020. RelevantID: Enjoy a next-generation approach to ID validation. https://www.imperium.com/relevantid/. (Accessed on 10/21/2021).

[32] Ipregistry. 2021. The Trusted Source for IP Address Data (geolocation and threat) - Ipregistry. https://ipregistry.co/. (Accessed on 10/21/2021).

[33] Hugo Jonker, Benjamin Krumnow, and Gabry Vlot. 2019. Fingerprint Surface-Based Detection of Web Bot Detectors. In *Proceedings of the 24th European Symposium on Research in Computer Security (ESORICS '19)*. Luxembourg.

[34] Craig Leth-Steensen, Zmira King Elbaz, and Virginia I Douglas. 2000. Mean response times, variability, and skew in the responding of ADHD children: a response time distributional approach. *Acta Psychologica* 104, 2 (2000), 167–190. https://doi.org/10.1016/s0001-6918(00)00019-6

[35] Xigao Li, Babak Amin Azad, Amir Rahmati, and Nick Nikiforakis. 2021. Good bot, bad bot: Characterizing automated browsing activity. In *Proceedings of the 42nd IEEE Symposium on Security and Privacy (S&P '21)*. Virtual Event, USA. https://doi.org/10.1109/SP40001.2021.00079

[36] MaxMind. 2021. minFraud Overview | MaxMind. https://www.maxmind.com/en/solutions/minfraud-services. (Accessed on 10/21/2021).

[37] MTurk. 2021. Amazon Mechanical Turk. https://www.mturk.com/. (Accessed on 10/21/2021).

[38] Peter M Nardi. 2018. *Doing survey research: A guide to quantitative methods*.

[39] The Tor Project. 2021. Tor Project | Anonymity Online. https://www.torproject.org/. (Accessed on 10/21/2021).

[40] Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT '21)*. Virtual Event, USA. https://doi.org/10.18653/v1/2021.naacl-main.466

[41] Qualtrics. 2021. Fraud Detection. https://www.qualtrics.com/support/survey-platform/survey-module/survey-checker/fraud-detection/. (Accessed on 10/22/2021).

[42] John J Shaughnessy, Eugene B Zechmeister, and Jeanne S Zechmeister. 2000. *Research methods in psychology*.

[43] Melissa Simone. 2019. Bots started sabotaging my online research. I fought back - STAT. https://www.statnews.com/2019/11/21/bots-started-sabotaging-my-online-research-i-fought-back/. (Accessed on 10/21/2021).

[44] Suphannee Sivakorn, Iasonas Polakis, and Angelos D. Keromytis. 2016. I am Robot: (Deep) Learning to Break Semantic Image CAPTCHAs. In *Proceedings of the 1st European Symposium on Security and Privacy (EuroS&P '16)*. Saarbrücken, GERMANY. https://doi.org/10.1109/EuroSP.2016.37

[45] Robert J Sternberg et al. 1999. *The nature of cognition*. MIT Press.

[46] Stefan Stieger and Ulf-Dietrich Reips. 2010. What are participants doing while filling in an online questionnaire: A paradata collection tool and an empirical study. *Computers in Human Behavior* 26, 6 (2010), 1488–1495.

[47] Andie Storozuk, Marilyn Ashley, Véronic Delage, and Erin A Maloney. 2020. Got bots? Practical recommendations to protect online survey data from bot attacks. *The Quantitative Methods for Psychology* 16, 5 (2020), 472–481.

[48] Peng Sun and Kathryn T. Stolee. 2016. Exploring Crowd Consistency in a Mechanical Turk Survey. In *Proceedings of the 3rd International Workshop on CrowdSourcing in Software Engineering (CSI-SE '16)*. Austin, Texas, USA. https://doi.org/10.1145/2897659.2897662
[49] VirusTotal. 2021. VirusTotal - Home. https://www.virustotal.com/gui/home/upload. (Accessed on 10/21/2021).
[50] Luis von Ahn, Manuel Blum, and John Langford. 2004. Telling Humans and Computers Apart Automatically. *Commun. ACM* (2004), 56–60. https://doi.org/10.1145/966389.966390
[51] Luis Von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum. 2008. recaptcha: Human-based character recognition via web security measures. *Science* 321, 5895 (2008), 1465–1468.
[52] W3C. 2020. WebRTC 1.0: Real-Time Communication Between Browsers. https://www.w3.org/TR/webrtc/. (Accessed on 10/22/2021).
[53] Dustin Wood, Peter D Harms, Graham H Lowman, and Justin A DeSimone. 2017. Response speed and response consistency as mutually validating indicators of data quality in online samples. *Social Psychological and Personality Science* 8, 4 (2017), 454–464.
[54] Christina Yarrish, Laurie Groshon, Juliet Daisy Mitchell, Ashlyn Appelbaum, Samantha Klock, Taylor Winternitz, and Dara G Friedman-Wheeler. 2019. Finding the signal in the noise: Minimizing responses from bots and inattentive humans in online research. *The Behavior Therapist* 42, 7 (2019), 235–242.
[55] Penghui Zhang, Adam Oest, Haehyun Cho, Zhibo Sun, RC Johnson, Brad Wardman, Shaown Sarker, Alexandros Kapravelos, Tiffany Bao, Ruoyu Wang, Yan Shoshitaishvili, Adam Doupé, and Gail-Joon Ahn. 2021. CrawlPhish: Large-scale Analysis of Client-side Cloaking Techniques in Phishing. In *Proceedings of the 42nd IEEE Symposium on Security and Privacy (S&P '21)*. Virtual Event, USA. https://doi.org/10.1109/SP40001.2021.00021

APPENDICES

A LABELING CRITERIA

Rust Survey. The survey contains three open-ended questions.

- **Q1**: "Usually one object is owned by one owner. What issues is this mechanism designed to solve?"
- **Q2**: "In your own words, please explain the above compiler error. Critically, please describe how the safety rule is violated, e.g., what program control constructs and data structures are involved, how the safety rule is applied to the context, and why the Rust compiler highlights some code."
- **Q3** is the same as **Q2**, but it is asked after showing participants an enhanced version of compiler error messages.

We label a response of the Rust survey as invalid if it satisfies one of the following four criteria: ① it has an answer irrelevant to the open-ended question, while the escaping answers (e.g., "I don't know") are considered relevant; ② it has an answer copied from other Internet sources (e.g., Google search); ③ all its answers are escaping answers; or ④ it has an answer that has more than ten words and is identical to another answer within the same (or in a different) response.

For criterion ①, answers to **Q1** matching it include those that do not mention any of "memory management/memory safety/use after free/double free", and answers to **Q2** (or **Q3**) matching it include those not discussing the provided compilation error or the related code snippets.

For criterion ②, we identify an answer to **Q3** that is directly copied from an irrelevant Stack Overflow post (Stack Overflow 67042725), which demonstrates an ingenuine effort to fill in the survey and an obvious violation of our survey requirements (i.e., not referring to external sources).

For criterion ④, we find a sentence with 24 words ("reasons. The data on the deep copy heap can be time-consuming, and the first variable needs to be invalidated to avoid secondary release") that appears in the **Q1** answers of eight submissions. It is extremely unlikely for honest participants to independently come up with the exact same sentence. Instead, this is more likely to be a fraudster using multiple identities to take the survey many times (for compensation).

In total, we code 150 responses with irrelevant answers (criterion ①), five responses with answers copied from the Internet (criterion ②), 76 responses with all escaping answers (criterion ③), and eight responses with identical answers (criterion ④).

MTurk Survey. The survey contains four open-ended questions.

- **Q1**/**Q2**/**Q3**: "What aspect of this profile most influenced your ratings?"
- **Q4**: "Please describe the strategies you used to assess profiles."

Criteria ①–④ are similarly applied to the MTurk survey with a few changes due to the different study and question design that is employed in each survey. Specifically, answers matching ① are those not referencing any factor (e.g., appearance or content) about the social media profile for **Q1**–**Q3**, and those not describing a behavior they use while assessing profiles for **Q4**. Additionally, since valid answers are relatively shorter in the MTurk survey, multiple answers are used together to determine whether a response is a duplicate of another. Thus, criterion ④ is altered in the following way: a response satisfies ④, if it has a subset of answers that has more than seven words and is *identical* to another subset of answers (excluding the escaping answers) within the same (or in a different) response.

For the MTurk survey, we code 81 responses with irrelevant answers (criterion ①), 24 responses with answers copied from the Internet (criterion ②), three responses with all escaping answers (criterion ③), and eight responses with identical answers (criterion ④).

B COGNITIVE PERFORMANCE MEASURES

We compare valid respondents with invalid respondents using t tests ($\alpha = 0.05$). Tables 2, 3 and 4 show the results. In these tables, "*", "**", and "***" represent that the p value is less than 0.05, 0.01, and 0.001 respectively.

Table 2: Average Response Time in Seconds in the Rust Survey. Some survey procedures (e.g., the consent form and welcome instructions) are not included in any phases. The Rust survey contains two pages that require participants to wait for several seconds to proceed the survey. The first page's wait time is two seconds. The second page (Test16) has a wait time of five seconds.

	Valid	Invalid	p value
Phase 1.1: demographics	35.6±13.5	44.6±3.2	0.52
Phase 1.2: previous experience	198.1±17.9	230.9±15.6	0.17
Phase 2: Rust knowledge	1503.1±210.6	476.8±33.2	***
Phase 3: post session	94.3±6.2	65.3±4.5	***
Whole Survey	2033.1±237.3	1466.7±106.7	*
Waiting for 2 seconds	32.2±15.2	104.9±24.4	*
Waiting for 5 seconds	7.5±0.1	10.9±0.8	***

Table 3: Average Ratings of 10-Point Scale Questions. In the Rust survey, we ask participants how difficult it is to comprehend a programming error three times. The first time is before showing participants the compiler error messages. The second time is after showing participants the original version of compiler error messages (v1). The last time is after showing participants the enhanced version of compiler error messages (v2).

	Valid	Invalid	p value
Rust programming experience	5.8±0.3	6.1±0.1	0.30
All programming experience	7.1±0.2	5.4±0.1	***
Difficulty level (first time)	6.5±0.4	6.5±0.1	0.90
Difficulty level (second time)	4.9±0.4	6.8±0.1	***
Difficulty level (third time)	4.3±0.4	6.8±0.1	***

Table 4: Average Ratings of 21-Point Scale Questions. In the Rust survey, we examined the effort that participants spent during Rust program evaluation. We asked participants the six NASA TLX questions after showing them the original compiler error messages (v1) and after showing them the enhanced version of compiler error messages (v2). p: p value.

	Error Message (v1)			Error Message (v2)		
	Valid	Invalid	p	Valid	Invalid	p
Time	7.8±0.9	13.0±0.3	***	6.9±0.8	13.1±0.2	***
Successful	14.4±0.9	12.8±0.3	0.08	14.9±0.8	12.5±0.3	**
Frustration	7.9±1.0	13.0±0.3	***	6.4±0.9	13.0±0.3	***
Hard	8.5±0.9	13.1±0.3	***	7.7±0.8	13.2±0.3	***
Physical	6.8±0.8	12.8±0.3	***	5.8±0.7	12.8±0.3	***
Mental	9.1±0.9	12.9±0.2	***	8.1±0.8	13.1±0.2	***

C SURVEY QUESTIONS

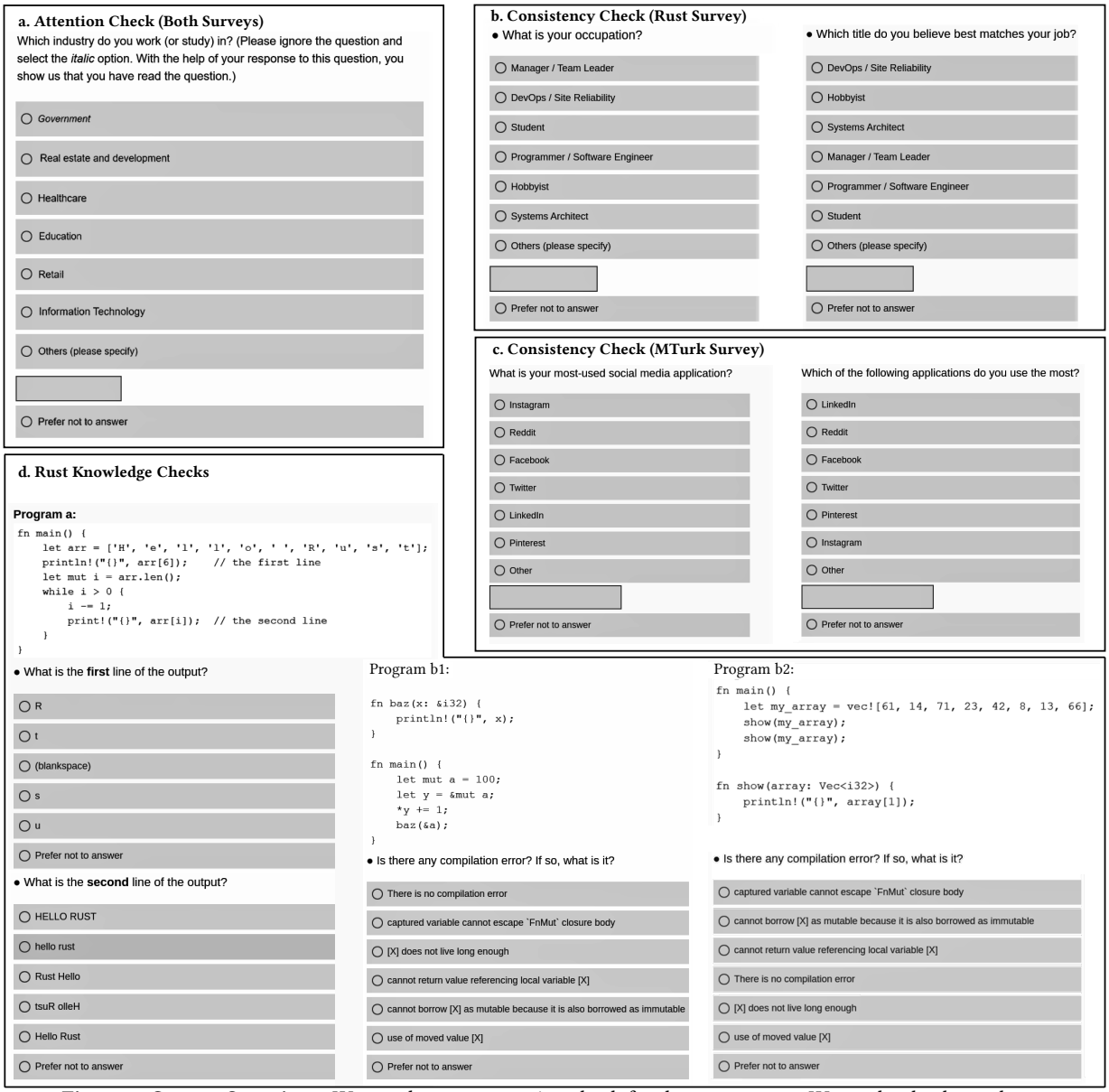


Figure 3: Survey Questions. We use the same attention check for the two surveys. We randomly choose between Program b1 and Program b2, and only present one to each participant.