A Human in Every APE: Delineating and Evaluating the Human Analysis Systems of Anti-Phishing Entities

Bhupendra Acharya and Phani Vadrevu

University of New Orleans, New Orleans, LA 70148, USA
bacharya@uno.edu, phani@cs.uno.edu

Abstract. We conducted a large-scale evaluation of some popular Anti-Phishing Entities (APEs). As part of this, we submitted arrays of CAPTCHA challenge-laden honey sites to 7 APEs. An analysis of the "click-through rates" during the visits from the APEs showed strong evidence for the presence of formidable human analysis systems in conjunction with automated crawler systems. In summary, we estimate that as many as 10% to 24% of URLs submitted to each of 4 APEs (Google Safe Browsing, Microsoft SmartScreen, Bitdefender and Netcraft) were likely visited by human analysts. In contrast to prior works, these measurements present a very optimistic picture for web security as, for the first time, they show presence of expansive human analysis systems to tackle suspicious URLs that might otherwise be challenging for automated crawlers to analyze.

This finding allowed us an opportunity to conduct the first systematic study of the robustness of the human analysis systems of APEs which revealed some glaring weaknesses in them. We saw that all the APEs we studied fall prey to issues such as lack of geolocation and client device diversity exposing their human systems to targeted evasive attacks. Apart from this, we also found a specific weakness across the entire APE ecosystem that enables creation of long-lasting phishing pages targeted exclusively against Android/Chrome devices by capitalizing on discrepancies in web sensor API outputs. We demonstrate this with the help of 10 artificial phishing sites that survived indefinitely despite repeated reporting to all APEs. We suggest mitigations for all these issues. We also conduct an elaborate disclosure process with all affected APEs in an attempt to persuade them to pursue these mitigations.

1 Introduction

As web-based social engineering attacks continue to rise in number and variety, it has become imperative for security organizations to invest in systems that inspect web sites for signs of maliciousness. Such systems are commonly referred to as **Anti-Phishing Entities (APEs)** and play a critical and omnipresent role in preventing web users from visiting harmful websites. For example, the Google Safe Browsing (GSB) service is a popular APE that receives URL reports from users around the world and verifies them. After verification, GSB sends malicious URLs to URL blocklists that are currently deployed in about 4 billion devices of users around the world [20]. It is to be noted

that despite the name, APEs are not only meant for thwarting phishing attacks and have the responsibility of identifying and blocking all kinds of malicious web content.

Given the scale of the web, it is reasonable to expect that a large proportion of visits from these APEs are from fully automated **web security crawler bots** which would likely be using machine learning techniques or carefully crafted heuristics to detect whether or not a given candidate page is malicious. In order to foil such attempts of these bots, attackers have begun to design and use phishing pages that are fitted with CAPTCHAs in the initial landing pages [14,23]. Since CAPTCHAs are inherently designed to prevent bots from bypassing them, they can be used effectively as a cloaking vector against APEs. Thus, it has now become imperative for APEs to augment security crawler bots with **human analysis systems** where human security analysts manually inspect (a subset of) web sites reported to them. Another important reason we expect APEs to have human analysis subsystems is for evaluating the potential false positive and false negative cases that might inevitably result from the bot-based automated analysis systems. Thus, these human analysis systems are vital for functioning of APEs. In addition to such importance, human analysis systems are by nature very expensive to maintain due to high labor costs. Given this, it is crucial for organizations to maintain robustness of these expensive human analysis systems to make sure they are not subject to targeted evasion attacks. To the best of our knowledge, there has been no study to date that focuses on evaluating the robustness of these human analysis subsystems of APEs.

In this paper, we attempt to fill this knowledge gap. We conduct the largest study thus far on delineating the human-based visits made by APEs. The most closely related work to ours is [13] which conducted a small scale study to detect the capability of APEs to overcome CAPTCHA-based blockages. The study limited each one of 7 popular APEs to 6 test phishing sites fitted with a CAPTCHA. However, unfortunately, quite contrary to our expectations laid out above, this study has found that none of the studied APEs were capable of clicking through CAPTCHAs in potential phishing pages. For this paper, we attempted to repeat this experiment albeit on a much larger scale. For this, we leveraged a scalable APE evaluation design methodology recently proposed in [10]. This allowed us to submit multiple test site sets of 100 sites each with different CAPTCHAs to each of 7 popular APEs. We also utilized this opportunity to also collect a wider range of data that captures the dynamic behavior of APEs when visiting websites. This data included all mouse, touch and key press events as well as events garnered from other advanced web sensor APIs such as Gyrometer, Accelerometer etc. To the best of our knowledge, this is the first work to collect and analyze such biometric data from APEs. With a similar setup, we also performed a user study involving 433 users in order to contrast this biometric data with that of regular users in order to gauge the existence of anomalies that can be abused by attackers for evasion attacks in the future.

In complete contrast to the results from [13], our study showed the existence of a vibrant and powerful ecosystem of human analysts being employed by at least 4 of the 7 popular APEs that we studied. These are Microsoft SmartScreen, Google Safe Browsing (GSB), Bitdefender and Netcraft. Our conservative estimates based on CAPTCHA-solving rates show that these APEs are capable of arranging as many

as 10-24% of submitted URLs to be visited manually by a human analysts. Thus, our such high numbers show present-day APEs in a very positive light for the first time in terms of their efforts to support formidable manual analysis systems. These measurements present a very optimistic picture for web security as, for the first time, they show presence of expansive human analysis systems to tackle suspicious URLs that might otherwise be challenging for automated crawlers to analyze.

These new findings thus also allowed us to perform the first systematic study of the robustness of these human analysis systems which revealed some glaring weaknesses in them. We saw that all the APEs we studied fall prey to the same issues such as lack of geolocation and client device diversity that exposes their human systems to targeted evasive attacks. Significantly, our analysis reveals that even cumulatively, the APEs are being affected by these issues. While prior works such as [17] have helped improve network and device diversity [10], we find that these changes have not carried over to the human analyst subsystems. For instance, while some APEs are using more than 40 different countries as sources for their bot visits, all of them seem to be using only one or two countries for sourcing traffic from human analysts. Similarly, we observed that large APEs such as GSB and Outlook were only using APE-specific browsers (such as ChromeOS and Microsoft Edge respectively) for their human analysts despite them lacking in general popularity. Also, none of the APEs are using mobile user agents for human vetting thus exposing users to potential "mobile-only" malicious web pages that completely avoid human analyst systems. In the case of Google Safe Browsing, we also saw evidence for some timing-based blind spots as no human analyst visits have occurred during the weekends.

Interestingly, we also found that most of these problems are generally not associated with the automated bot systems of these APEs. This indicates that these issues can likely be fixed easily for the expensive human inspection systems as well. However, as an exception to this, we found one issue that is currently affecting all APE systems purported to be visiting from Android/Chrome devices. Namely, we found that none of these visits were giving away web sensor API data upon page load in sharp contrast to most real Android/Chrome devices that do so in their default configuration (as per our user study). We show that this discrepancy can be leveraged to create long-lasting phishing pages tailored towards this very popular platform. We suggest mitigations for all these issues. We conducted an elaborate disclosure process with all APEs in an attempt to persuade APEs to pursue these mitigations and help make a practical impact in improving the security posture of all APEs.

In summary, our contributions with this paper are as follows:

1. Delineation: With the help of a large-scale study, we demonstrate for the first time, evidence for industry-wide use of elaborate human analysis systems by APEs.
2. Evaluation: We conduct the first systematic evaluation of the robustness of these human systems and find multiple serious issues across different APEs that expose them to targeted evasive attacks.
3. Impact: We suggest practical mitigations to resolve these issues and conduct an elaborate disclosure process with the affected APEs.

2 System Description

2.1 System Overview

The predominant goal of our project was to measure the prevalence of any human analysis systems being deployed by popular APEs. If we found the presence of any such human analysis systems, we had also planned to conduct forensic analysis on such systems in an effort to evaluate their robustness to evasion attacks. In order to support these goals, we built a measurement and data collection system for APEs. An overview of this system is presented in Fig. 1.

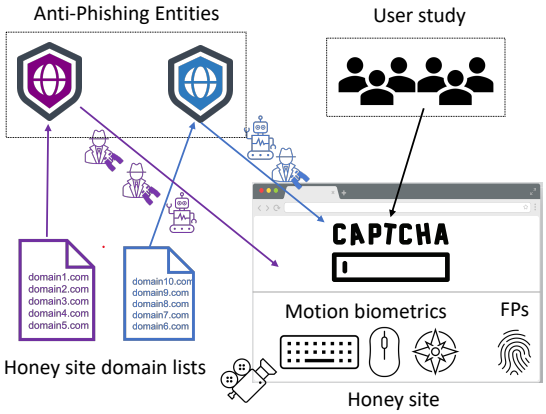


Fig. 1: System Overview

The main component of our system is the honey server that is capable of serving several identical honey sites that are designed for the measurements in this project. Each honey site is fitted with a CAPTCHA in order to help us determine if the visitor is human. The ethical considerations behind this design are discussed in Sec. 5. We relied upon the APE evaluation approach proposed in [10] in order to make our system scalable. Concretely, we registered multiple domain names and made them all point to the same honey server. We then separated these domain names into disjoint sets and submitted them to various APEs. These submissions thus elicited visits from both the bot and human subsystems of these APEs. For each visit, we leveraged the event knowledge of whether a relevant CAPTCHA challenge has been solved or not as a data point to infer whether that visitor is human or bot.

Besides deploying CAPTCHA challenges, we also equipped each honey site with two kinds of forensic recording capabilities as discussed below.

1. Motion biometrics. We embedded JavaScript code in our honey sites to listen and record multiple Web API events generated from UI devices such as mouse events (e.g. `click`, `mousemove` etc.), keyboard events (e.g. `keyup`, `keydown` etc.), and Touch events (e.g. `touchstart`, `touchmove`). Furthermore, upon page load, we

register event listeners for Web Sensor APIs to record data from common mobile-specific devices such as gyroscope, accelerometer, magnetometer and light sensors. As some of this data (such as movement data) is continuously generated as long as visitors are on the honey site, we added the ability to offload the collected data to our server every 500 ms in order to minimize the risk of losing data due to unexpected network issues. We later analyze this collected data to reveal some interesting insights.

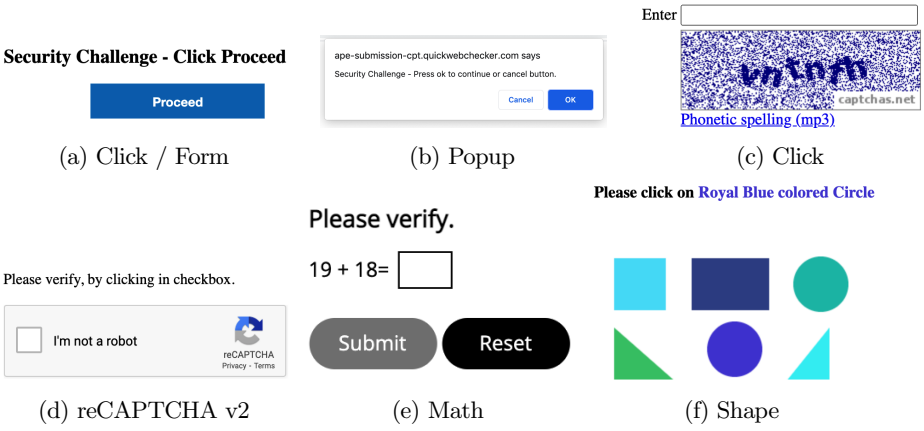
2. Browser fingerprints. We also fitted each of the honey sites with capabilities to collect different valuable browser fingerprints from the visitors. Our plan was to utilize this data to correlate visits across different honey sites. Source IP address and HTTP request headers are some of the basic browser fingerprints that we collect from a visitor. Apart from this, we also collect some other sophisticated browser fingerprints. Recently, it has be shown that popular APEs can be easily identified (and evaded) [10] based on HTML5 API-based fingerprinting techniques such as Canvas [16] and WebGL fingerprints. For this reason, we utilized the browser fingerprinting code implementation in [10] to collect these two additional data points from the visitors to our honey sites.

User study. While soliciting visits from APEs to our honey sites allowed us to collect data from both bots as well as human analysis systems, we also collected similar data from real humans. This enabled us to compare and contrast the set up used by the human analysis systems of APEs in the context of how well they blend in with systems used by regular humans. For this, we set up a user study in which we requested each participant to solve the CAPTCHA challenges presented to them. More details about this are described later in this section.

2.2 CAPTCHAs for honey sites

As mentioned previously, we wanted to use the ability of the visitors to solve CAPTCHA challenges as a key data point to positively identify human visitors from APEs to our honey sites. However, depending on only a single type of challenge to differentiate between humans and bots is risky as APEs might utilize sophisticated bots or other specialized solutions that might break such a challenge. Hence, instead of relying on just a single challenge, we use an array of 7 CAPTCHAs ranging from those that are easily by-passable to commercial as well as custom-built variants. We describe them below along with our rationale for choosing them.

Easy CAPTCHAs. We crafted three CAPTCHAs which can potentially be easily circumvented by a bot. These CAPTCHAs require the user to simply click on a regular HTML clickable button or a form element or a JavaScript `confirm()` dialog box. We refer to these as **Click**, **Form** and **Popup** CAPTCHAs respectively. Click and Form challenges have to be solved by clicking on an element. They visually look alike and are shown in Fig. 2a. These can be brute-forced easily by any web crawling that uses an automated browsing tool (for example, Selenium [6] or Puppeteer [4]) to click on all "clickable" elements in a page. When implementing these two CAPTCHAs, we used a JavaScript function as the `onclick` event handler for the buttons to record CAPTCHA success. However, these functions can also be called directly giving the appearance of a successful button click to our backend server. Thus, this provides for another simple bypass mechanism for bots which might be configured to blindly



(a) Click / Form (b) Popup (c) Click

(d) reCAPTCHA v2 (e) Math (f) Shape

Fig. 2: Various CAPTCHAs deployed in honey sites used for studying APEs.

call event handler functions even without making any clicks whatsoever. The Popup challenge pages trigger a function on page load which a `Window.confirm()` DOM API call. If this call returns `true`, then passes a success indicator to our backend server. For this to happen, the browsing agent needs to click on (or presses `return` key) on the "OK" button on the JavaScript dialog that pops up (Fig. 2b). However, we point that some previous security crawler setups have managed to automate JS dialogs interactions with the help of in-browser code changes [21] or browser extensions [15] which can be some potential ways to bypass this CAPTCHA in an automated fashion.

Commercial CAPTCHAs. As the CAPTCHAs above can potentially be brute-forced or bypassed by an automated agent, these are not sufficient to confidently separate humans from bots being used by APEs. Hence, we also used two CAPTCHAs that are much more sophisticated and similar to the ones used in real websites. The first one we used is a **text** CAPTCHA service provided by `captchas.net` [1] (Fig. 2c). The second is a "behavioral" CAPTCHA from Google named **reCAPTCHA** (version-2) [5]. This CAPTCHA asks visitors to click on a button to verify that they are not a bot during which time a risk analysis engine checks static as well as dynamic behavioral patterns of the visitor to determine if they are a bot or not (Fig. 2c). In case of suspicion, the engine will lead the visitor to an image-grid based visual CAPTCHA which the visitor will need to solve to prove they are human [5].

Custom CAPTCHAs. While the above mentioned commercial CAPTCHAs are not very easy to solve in an automated fashion, it is not an impossible endeavor. For example, researchers have demonstrated that an earlier version of Google's re-CAPTCHA can be broken easily by using a Convolutional Neural Network (CNN) model [12]. More recently, it has been demonstrated that Generative Adversarial Networks (GANs) can be used to solve generic versions of text CAPTCHAs at the same rate as humans [22]. Furthermore, it is also possible for APEs to seek cooperation from CAPTCHA providers in order to bypass them. This is especially possible if the APE as well as the CAPTCHA provider are from the same organization. For example,

Google's Safe Browsing service can potentially have an internal arrangement with Google's reCAPTCHA service so as to allow their crawlers to automatically bypass all their requests. In order to account for such potential automated bypasses, we crafted two custom in-house variants of CAPTCHAs: Math and Shape CAPTCHA.

The **Math** CAPTCHA asks the visitor to give the output of a simple arithmetic operation. It is to be noted that our intention here was to keep this CAPTCHA's functionality similar to existing Math CAPTCHAs such as [3] while at the same time using our own code for implementation. In addition, we also built a new custom CAPTCHA named **Shape** CAPTCHA. It simply asks the visitor to click on a random geometric shape made of a random color (Fig. 2f).

Thus, we design our CAPTCHA challenges above and separate them into 3 categories with an expectation that they are going to be increasingly difficult to be solved by an automated bot. However, we clarify that our goal with these challenges is not to create any CAPTCHA that is impossible for bots to solve. We concede that even our custom challenges can be tackled successfully by APEs if they tailor their bots after analyzing some samples of our challenges. Instead, our goal here is simply to utilize multiple CAPTCHAs so as to enable collection of data on click-through rates from multiple vantage points and then correlate this information with other forensic data from these visits in order to estimate the extent of human analysis components being deployed by present-day APEs.

2.3 Experimental Setup

We will now describe the experimental setup we used for the measurements and analysis we performed for this paper.

Anti-phishing entities. We tried to cast a wide net in terms of the anti-phishing entities we consider for this work in order to ensure our results are generalizable across the spectrum of APE providers. The prior work [13] which first attempted to study the human analysis systems of APEs focused on 6 providers in their main analysis. Out of these 6, PhishTank has disabled new user registrations to their web portal for submitting URL reports and was hence left out of consideration for this paper. We included the remaining 5 providers in our analysis: Google Safe Browsing (GSB), Microsoft SmartScreen, APWG, Netcraft and OpenPhish. Further, we also added two other APEs in our analysis: Bitdefender and ZeroCert. As in [10], using Python and Selenium-based browser automation, we created a system to be able to send URL reports containing links to our honey sites to the web portals of these 7 providers at a preset frequency.

Honey sites. For each of the 7 APEs and 7 CAPTCHAs we considered, we created 100 honey sites. Thus, throughout our experimentation period, we submitted 700 honey sites to each of the 7 APEs. We utilized a distinct and new domain name for each honey site but due to financial restrictions we could only register 10 new second-level domain names (TLD+1) for this entire experiment. We created unique TLD+2 domains for the honey sites uniformly under these 10 TLD+1 domains. These domains were registered under the popular `.com` TLD as well as other extensions often abused by phishing actors such as `.xyz`, `.site` and `.club`. We relied on wildcard DNS records and `.htacess` rewrite rules in the web server to support these TLD+2 domains for all the 4900 sites utilized in our study. We note that this practice

of relying upon a handful of TLD+1 domain names to conduct a large-scale analysis of APEs has been demonstrated to be a viable method in [10]. We avoid discussing this design rationale here for brevity but instead refer interested readers to [10] for detailed arguments as well as experimental results supporting this approach.

We spread out these 700 URL submissions to each APE over a five week period in March and April 2021 averaging about 20 per day.

User study. For our user study, we first sought an IRB exemption from our university and then used Amazon's Mechanical Turk (MTurk) platform to recruit participants for clicking through the CAPTCHAs on our honey sites. For this, we created a dedicated honey site with 7 different web pages for each of the 7 CAPTCHA challenges. As explained before, these pages are identical to the pages on honey sites distributed to the APEs. We set up this study as 7 different user surveys on MTurk in order to ensure that no same participant visits a particular challenge page more than once. We conducted this study over a one week period in the first week of April 2021. Overall, each of our 7 web pages were visited 210 times thus generating a total of 1470 visits in the user study. Based on data provided by MTurk and IP address information, we were able to attribute these visits to 433 unique participants from about 26 different countries all over the world.

3 Delineation of Human Analysis Systems

The experiment described previously yielded data about the click-through rates of APEs for various CAPTCHAs as shown in Table 1. In the table, the first "Visits" sub-rows for all APEs show information about the number of visits (i.e. HTTP sessions) that were made to the honey sites of different categories that we submitted. Specifically, the first sub-columns (marked by 👫) show the number of visits in which the given category's CAPTCHA challenge was solved successfully and is hence indicative of a human visit. The second "All" sub-columns represent the total number of visits made for each of the 100 sites submitted in a given category irrespective of whether the challenge was solved. It is to be noted how many of these numbers are more than 100 as APEs often make repeated visits to a submitted site. The next "Sites" sub-row for each APE shows these same counts at a "site-level" i.e. the number of unique sites whose CAPTCHA pages were clicked-through or visited by APEs out of a maximum of 100 for each category. Analysis of the number of unique sites visited shows that our submission module has successfully solicited requests from APEs in most cases. Except for Bitdefender, most other APEs have visited most of the sites we submitted to them. APWG, Netcraft, OpenPhish and ZeroCert in particular, have visited all 700 sites that we submitted to them at least once. Across all APEs, we can infer that many sites are visited multiple times by the large amount of visits for each category of submitted sites. These results indicating very high scan-back rates and repeated visits agree with prior recent results from [10].

CAPTCHA / APE		Easy						Commercial				Custom				Summary	
		Click		Form		Popup		Text		reCAPTCHA		Math		Shape		👫	All
		👫	All	👫	All	👫	All	👫	All	👫	All	👫	All	👫	All		
APWG	Visits	0	455	0	471	0	2575	0	466	0	466	0	471	0	473	0	5377
	Sites	0	100	0	100	0	100	0	100	0	100	0	100	0	100	0/700 (0%)	700/700 (100%)
Bitdefender	Visits	19	610	19	713	~~71~~	32 735	14	754	11	740	20	509	14	447	129	4508
	Sites	15	38	17	33	~~28~~	27 38	14	46	11	44	16	34	12	37	112/700 (16%)	270/700 (39%)
GSB	Visits	37	236	36	243	~~192~~	8 226	23	238	28	228	32	238	26	249	190	1658
	Sites	29	98	31	97	~~92~~	5 92	23	97	27	96	27	97	26	99	168/700 (24%)	676/700 (97%)
SmartScreen	Visits	27	156	30	90	44	86	12	96	18	87	18	89	11	90	160	694
	Sites	22	84	29	84	44	83	12	86	18	79	18	88	11	85	154/700 (22%)	589/700 (84%)
Netcraft	Visits	~~111~~	12 902	~~116~~	16 899	0	1306	15	813	1	571	11	929	15	837	70	6257
	Sites	~~99~~	12 100	~~100~~	16 100	0	100	15	100	1	100	11	100	15	100	70/700 (10%)	700/700 (100%)
OpenPhish	Visits	0	588	0	579	0	583	4	47366	0	584	0	586	1	577	5	50863
	Sites	0	100	0	100	0	100	3	100	0	100	0	100	1	100	4/700 (0.5%)	700/700 (100%)
ZeroCERT	Visits	0	208	0	204	~~57~~	0 186	0	134	0	130	0	155	0	154	0	1171
	Sites	0	100	0	100	~~57~~	0 100	0	100	0	100	0	100	0	100	0/700 (0%)	700/700 (100%)

Table 1: APE/CAPTCHA success rates; The 👫 column columns shows information about successful click-through visits while "All" column shows all visits. "Visits" sub-rows show # of HTTP Sessions while "Sites" sub-rows show the # of unique sites visited. APEs that can be deduced to have significant human analysis components are highlighted in blue. Gray cells indicate cases where APEs are successfully using some automated solutions to solve CAPTCHAs. We accounted for these cases and deducted the numbers with the help of a clustering process as described in Sec. 3

One key thing to observe from Table 1 is that 4 APEs, namely, Bitdefender, GSB, SmartScreen and Netcraft, have had significant and consistent success in clicking through the entire spectrum of our challenges including customized CAPTCHA challenges. This indicates high likelihood of a human analysis component in their systems. These 4 APEs are highlighted in blue color in the table and we focus most of our attention on these in the rest of this paper. Other than a couple of exceptions, these 4 APEs have been able to click-through at least 10% of the submitted sites across all challenges. This result indicates a stark contrast from the results presented in [13] which showed that practically none of the APEs studied were able to solve the CAPTCHAs presented to them. While we cannot ascertain the reason for this difference, we surmise that this could either be due to the small scale of their experiments which occluded these insights or the early timeline of their experiments at which time APEs have potentially not yet deployed the human analysis systems.

When looking at the success rates of submitted sites, 5 CAPTCHA/APE combinations (highlighted in gray) stand out in terms of anomalously high click-through rates in comparison to other numbers in the same rows[1]. For example, GSB made 192 successful visits to Popup challenge pages while the number of successful visits to all other challenges were only around 30. Similarly, Bitdefender visited 71 Popup pages successfully while the rest of the pages had a maximum of 20 visits. Netcraft made 111 and 116 successful visits to Click and Form challenge pages while the remaining categories solicited only about 15 successful visits. Finally, ZeroCERT was only successful in solving Popup challenges. It is to be noted that all these 5 cases involve Easy category CAPTCHAs which as we mentioned previously can easily be tackled by automated crawler setups. We suspected that this is the case with these pages and conducted a clustering analysis that helped us identify such cases in a generic, non-heuristic manner. We discuss this below.

3.1 Filtering automated crawler visits

As discussed in the previous section, our honey sites collected the Canvas and WebGL browser fingerprint information as well as motion biometrics data from all the visits. For each of the APEs, we leveraged this information to cluster all the successful CAPTCHA solving visits that were made to our sites. The results of this clustering process were very insightful. We noticed that across multiple APEs, there were a few clusters where all of the component visits had no motion biometric data whatsoever (i.e. keyboard, mouse or touch data) despite solving the challenge each time. Since it is not possible for a human analyst to solve these challenges without such movement we consider these as actions of automated analysis systems. Interestingly, for each APE, all these "no-motion" clusters corresponded with the 5 APE/CAPTCHA combinations (the gray cells in the table) which we already suspected. As a result, we decided to discount all the visits that were parts of these clusters from the human analysis components of our study. The resulting numbers after these deductions are shown in the table itself.

[1] Please refer to the numbers that were struck through in the gray cells in Table 1

It is to be noted how these deductions greatly decreased the numbers in the suspected cases and made the human visit counts for all 4 APEs ultimately much more uniform across different challenges. These deductions show that GSB, Bitdefender and ZeroCERT used bots to solve Popup CAPTCHA challenges and that Netcraft used bots for solving large numbers of Click and Form CAPTCHA. This is likely using the mechanisms we already discussed in Sec. 2.2. Interestingly, after these deductions we can see that Netcraft has a 1 to 1 correspondence for sites and visits from human systems for all challenges. This means that with Netcraft, human analysts typically do not revisit a visited site unlike in the case of the other APEs. Another interesting thing to note in is the absence of any evidence for bot-assisted challenge solving for GSB's reCAPTCHA challenges despite both GSB and reCAPTCHA services belonging to Google. GSB's automation seemed to only be limited to Popup challenges.

3.2 Human systems' impact analysis

The last column of Table 1 indicating the summary statistics of human visits for each APE paints a very optimistic picture. In summary, the 4 APEs were able to visit between 10% and 24% of the submitted sites. These are very impressive numbers given that APEs such as Google Safe Browsing receive billions of URL reports everyday [2]. This shows that current APEs have likely invested large amounts of financial resources towards building these expansive human analysis systems. Assuming that each APE has an independent mechanism for deciding if a submitted site will be viewed by human analyst, we can also compute the probability for a candidate site that is submitted to all the APEs to be inspected by a human. Concretely, let p_i denote this probability for each APE. Then, the summary probability can be computed by $1-\prod_{i=1}^{4}(1-p_i)$ which gives a formidable probability value of 43% for human analysis of a candidate site. In addition, it should also be noted that we only made a single submission of report to APEs for each honey site from a single end point. In the real world, large-scale social engineering attacks will often trigger repeated reporting of the complicit URLs from diverse sources. In such a case, it is possible for this manual analysis probability to be even more than computed here.

4 Evaluation of Human Analysis Systems

The previous section demonstrated that multiple APEs built elaborate human analysis systems which are arguably much more expensive to maintain and scale than their bot counterparts. Given this, it is very important to ensure that these systems are robust and do not carry any undue weaknesses. In this section, we focus on studying the prevalence of any such issues which can potentially enable targeted cloaking attacks against human analysis systems in the future.

4.1 Geolocation-based evasion attacks

Prior works have advocated the use of diverse geolocations for APEs when visiting candidate phishing sites in order to thwart geolocation-based cloaking techniques

used by attackers [17]. More recent work has shown that many APEs have in fact heeded this recommendation and heavily diversified the network infrastructure used for visits to the candidate sites [10]. We now investigate if the diversification has also made its way to the human analysis systems used by APEs. For this, we compare the network diversity of visits coming from human analysts with the overall network diversity of the visits. These results can be seen in Table 2 which shows the distinct number of IP addresses (IP) as well as countries associated with these IP addresses in the two sets of data for each APE. We also repeat the numbers about site visit counts for the two sets in order to present a context for this comparison.

APE	**Sites**		**IP**		**Country**	
	🕴	**All**	🕴	**All**	🕴	**All**
APWG	-	700	-	3793	-	14
Bitdefender	112	270	37	940	1 (Romania)	40
GSB	168	676	85	843	1 (India)	2
SmartScreen	154	589	100	326	1 (India)	13
Netcraft	70	700	14	1633	2 (UK: 97%)	49
OpenPhish	4	700	-	4452	-	63
ZeroCERT	-	700	-	3	-	1

Table 2: Table demonstrating the lack of diverstiy of geolocation in requests made by the human analysis systems across different APEs.

Firstly, we can see that the data in the "All" column are mostly in agreement with prior results in [10] and show that APEs are persisting to invest in diversification of the network infrastructure used for vetting websites. For instance, OpenPhish used 4452 IP addresses and APWG used 3793 IP addresses to visit the 700 sites we submitted to them. Similarly Netcraft used 1633 addresses for crawling the 700 sites submitted to them. However, their human analysis system, on the other hand, visited 70 sites using only 14 different IP addresses showing a vast difference in the ratio of IP addresses used to the number of sites visited between the two cases (2.33 for all vs. 0.2 for the human system). Same is the case for IP addresses of other APEs as well. These differences become even more stark when we consider the country associated with these IP addresses. While Bitdefender uses 40 different countries all over the world for visiting candidate phishing sites, their human analysis system uses only IP addresses belonging to Romania for this purpose. Similarly, 97% of Netcraft's human system visits (68/70 visits) are from UK, although overall, they use IP addresses from 49 different countries. It is to be noted here that Bitdefender is head quartered in Romania while Netcraft is headquartered in UK which likely explains why these countries were chosen by them for hosting their human analysis systems. On a similar note, GSB and Microsoft SmartScreen are only using IP addresses from India for all their human analysis system visits. Thus, even though all of these are global companies with users all over the world,

an attacker can easily avoid their elaborate human analysis systems by simply ignoring potential victims from a handful of countries. Concretely, if an attacker can set up an evasive malicious site that specifically serves benign content to India, Romania and UK, our results show that majority of human analysis visits can be evaded.

Mitigations. We strongly recommend APEs to adopt network request diversification infrastructure for their human analysis systems to avoid geolocation-based evasion attacks discussed above. This can be easily achieved with the help of solutions such as commercial VPNs which can provide support for switching between multiple IP addresses globally. Our results above which indicate the general visits of APEs coming from a large number of geolocations already points to the fact that APEs are already potentially using such mechanisms for their bot analysis systems. Upon further analysis of the data pertaining to visits from diverse geolocations, we found an interesting case study regarding Bitdefender. We noticed that 54 honey sites submitted to Bitdefender had HTTP sessions in which each session involved HTTP requests from IP addresses in 2-5 diiferent countries all over the world. Collectively, we noticed that IP addresses from 13 different countries were used for this. While switching IP addresses mid-session might unfortunately make these mechanisms conspicuous and thus make the IP addresses fingerprintable by attackers, nevertheless, this is still a step in the right direction. Proper implementation of such IP address switching mechanisms (without any side channels such as session cookie sharing) in human analysis systems can make them thwart geolocation-based evasion attempts by attackers. We thus recommend APEs to pursue such tactics.

4.2 User Agents

Another important issue pointed out in [17] was the effectiveness of certain "device-type" based cloaking attacks on APEs. For example, their research showed that a phishing website set to distribute malicious content only to mobile (Android/iOS) user-agents tends to be very resistant to blocklisting. As a result, APEs have evolved tremendously to improve user agent diversity as was evidenced in [10]. We also attempted to measure this with our our data. Table 3 demonstrates this. The second column in the table shows the popularity of OS/Browser combinations as per our user study. While the source of our user study participants (Amazon MTurk), might have biased the data towards more desktop users than normal, we believe that the proportion of users using various platforms in desktop and mobile platforms is still a good indicator of the popularity of user agents as it falls in line with results from larger studies [7]. The "All Requests" part of the table shows the probability that a domain visited by any of the APEs will be done so with a particular `User-Agent` header as per our data. We marked any probability value less than 0.1 in red and any value more than 0.5 in green to highlight good and bad values. Note that these values often sum up to more than 1 as the same domain submitted to an APE is often visited from multiple platforms. For the "Combined" column, we treat these individual probabilities as being related to a random event and obtain the combined probability for a domain that is submitted to all APEs to be visited by a particular user agent. To clarify, assume p_i is the probability for a user agent to be visited by an APE i. We compute the combined probability for a visit from the same user

agent by computing $1-\prod_{i=1}^{n}(1-p_i)$. The results confirm those in [10] that APEs have largely evolved to improve diversity in the user agents they use to vet candidate phishing pages. We can see that even the lowest combined probability is about 0.25 for Windows/Opera thus showing that all popular user agents are adequately represented (cumulatively) by the APEs. Of particular note is Bitdefender and OpenPhish both of which have a significant amount of visitors from a diverse set of user agents.

		All Requests								Human System				
OS/Browser	User Study	APWG	Bitdefender	GSB	SmartScreen	Netcraft	OpenPhish	ZeroCERT	Combined	Bitdefender	GSB	SmartScreen	Netcraft	Combined
Windows/ Chrome	279/433 (0.64)	0.51	0.74	1.00	0.10	0.95	1.00	0.00	1.00	0.82	0.00	0.18	0.14	0.87
Windows/ Firefox	32/433 (0.07)	0.82	0.32	0.00	0.00	0.46	0.14	0.00	0.94	0.12	0.00	0.00	0.01	0.14
Windows/ Edge	18/433 (0.04)	0.14	0.07	0.08	0.91	0.00	0.14	0.00	0.94	0.02	0.00	0.82	0.00	0.83
Windows/ Opera	6/433 (0.01)	0.00	0.14	0.00	0.00	0.00	0.12	0.00	0.25	0.00	0.00	0.00	0.00	0.00
Android/*	40/433 (0.09)	0.04	0.14	0.00	0.00	0.69	0.14	0.00	0.78	0.00	0.00	0.00	0.00	0.00
macOS/*	33/433 (0.08)	0.09	0.16	0.00	0.00	0.26	1.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
ChromeOS/*	13/433 (0.03)	0.00	0.00	0.38	0.00	0.00	0.00	0.00	0.38	0.00	1.00	0.00	0.00	1.00
Linux/*	7/433 (0.02)	0.01	0.28	0.00	0.00	0.34	0.17	0.43	0.78	0.05	0.00	0.00	0.84	0.85
iOS/*	5/433 (0.01)	1.00	0.18	0.00	0.00	0.97	0.15	0.00	1.00	0.00	0.00	0.00	0.00	0.00

Table 3: Lack of diversity in the browsing agents used for human analysis systems of APEs. The numbers in the cells indicate the probability that a given APE will visit a submitted site with a given `User-Agent` header. Values above 0.5 are in green and below 0.1 are in red.

The corresponding probability that a human analyst will use a particular user agent is on the last section of the table. We only consider the 4 main APEs which had human visitors for this. These probabilities are also combined similar to the other section and are in great contrast with the other one. In particular, we can see that there is not a single human request from Android, iOS, macOS, Windows/Opera user agents from even a single APE. This leaves all these popular platforms exposed to targeted evasion attacks that completely avoid human analysis. Note that as this includes both Android and iOS, this finding means that all mobile users can be exposed to targeted phishing attack pages that can evade all human analysis. It is also interesting to see the user agent diversities for human analysis systems of individual APEs. We note that all GSB systems simply use Google's own Chrome OS based systems for all of their human analysis despite them being not very popular in the

wild. Similarly, SmartScreen's human analysis systems are predominantly using the Microsoft's own Edge browsers for this. Both of these APEs are thus largely using uncommon browsing agents leading to potential cloaking attacks. It is also interesting to see how Bitdefender also has a great lack in diversity in their human systems (compared to their general requests) although they fare better than Google and Microsoft by atleast using the most popular "Windows/Chrome" user agent for all their visits thus protecting at least a majority of users from targeted evasion attacks.

Mitigations. We strongly recommend APEs to improve user agent diversity for their human analysis systems. However, this is arguably a complicated problem. While simple measures such as adopting user-agent changing extensions [8,9] might seem to solve this problem on the surface, APEs will face a risk of creating new browser anomalies which miscreants can abuse to fingerprint or evade analysis systems [10]. Another more viable solution is to truly improve diversity in the systems used by human analysts by adopting diverse browser/OS platforms for their systems. At the very least, all popular desktop and mobile platforms used by majority of users should be covered by these analysis systems.

4.3 Timing blind spots

We next measured the time it takes for human analysts of different APEs to first visit a submitted site. The median times for different APEs varied with Bitdefender and SmartScreen being the fastest APEs with less than 4 hours of human response time. GSB was the slowest with a median turnaround time of about 30 hours. This is very slow compared to GSB's overall median response time of only about 34 minutes; but this is understandable as this figure includes automated crawlers which are expected to be more responsive. However, we were still intrigued by this relatively low response time of GSB's human analysis system in comparison to other APEs and investigated this further. For this, we mapped the time of visits from human analysts into Indian Standard Time (IST) as we saw previously that all GSB human visits were from IP addresses in India. The graph depicting the day of these visits in Figure 3 shows that none of the 190 visits from GSB happened over the weekends (per IST). This is likely because the human analyst system was being run in a typical office like setup that does not operate over the weekends. However, this can be abuse by attackers. For example, a social engineering attacker who starts a campaign on Friday night can effectively have two full days before the attack is analyzed by a human from GSB. This leaves a sufficiently large time gap for a large-scale campaign.

Mitigations. Such timings blind spots need to be plugged by APEs by promoting capabilities for human analysis at least on a daily basis. If such changes are infeasible due to financial restrictions, then another potential approach could be to share candidate phishing URL data with other APEs that have complementary human analysis capabilities.

4.4 Sensor API-based mobile evasion attacks

As described in Sec. 2, one of the novelties of our experimental setup was the collection of biometric data from APEs such as information about keyboard, mouse events

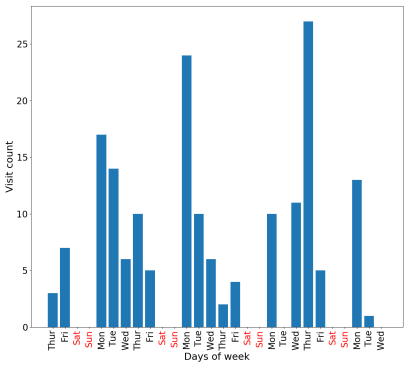


Fig. 3: Daily visits of GSB analysts

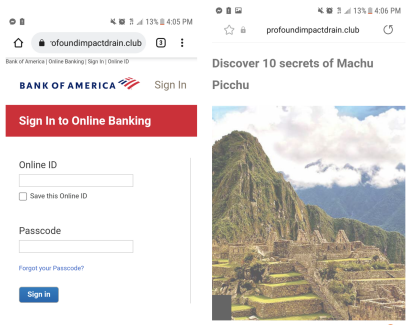


Fig. 4: Uncloaked and cloaked sensor API-based phishing pages

as well as other events from sensor devices. Therefore, we explored the possiblity of developing evasive attacks against APEs by using this data. Interestingly, while human visits from APEs did result in keyboard and mouse data, we were unable to collect any sensor data from any of the APEs we studied. Note that none of the APE visits had any web sensor data even when they proclaimed to be coming from mobile user agents. This is anomalous as our tests with multiple Android/Chrome devices showed that they all emit sensor data from the `Accelerometer` as well as `deviceMotion` web APIs in their default settings. In order to remove personal bias, we also performed the same analysis on 35 users in our MTurk study and found that all of them were also emitting data from the `accelerometer` API and `deviceMotion` APIs thus showing that many users do not change the default settings on these devices. Our data analysis also showed that all the affected Android/Chrome devices emit the first movement data with in the first few seconds (median time = 2.5 seconds). On the other hand, all the user agents for whom the `accelerometer`/`deviceMotion` web APIs are inaccessible due to user permission issues or lack of support (such as in the case of APEs) throw an error message during page load itself. We thus concluded that it will be possible to create effective phishing sites capitalize on these anomalies to target Android/Chrome users. Given that Android/Chrome is one of the most popular mobile user agents (35/45 mobile users in our study), this attack can leave a lot of users exposed to pages that conduct targeted social engineering attacks with impunity.

Evaluation. We implemented the cloaking logic in JavaScript and PHP and registered 10 new domain names to host 10 evasive phishing sites for evaluation. All of these sites show a "Bank of America" phishing page for any Android/Chrome user agent that emits `accelerometer` and `deviceMotion` data. On the other hand, if there is any error in accessing these web APIs or if there is a 10 second timeout without any such data emissions, we display a cloaked page to the user as in Fig. 4. As baseline, we also created 5 phishing sites that show the same phishing page without any cloaking logic. We started the evaluation in July 2021 when we reported all 15 sites to the 7 APEs we considered in this paper. On the very first day of this experiment, after our reporting,

all 5 baseline sites got blocked in all major web browsers with GSB blocklisting them in as little as 2.5 hours. This figure is in line with prior studies [17,10] and shows that the phishing pages we created are considered as malicious by APEs. Upon noticing that none of the 10 evasive sites got blocked, we continued to report these 10 sites daily over the next two weeks to all 7 APEs. Despite this, all of these 10 sites remained unblocked indefinitely until now spanning a period of more than seven months.

Mitigations. We showed that it is possible to create evasive phishing attacks against Android/Chrome platform due to two main reasons: (1) None of the APEs' human analysis systems perform testing using Android/Chrome devices. (2) None of the APE's general visitors that purport to use Android/Chrome user agents (likely bots) emit web sensor API data similar to how the real devices behave. Solving either of the problems will help thwart this attack vector. As already discussed earlier, the mitigation for (1) is to simply support more diversity in the devices used by human analysis systems of APEs. To handle (2), one potential solution is to improve the automated crawler technology used by the APEs emit fake web sensor data similar to real devices. We already saw similar methods being used by Netcraft in a lot of their failed visits to our CAPTCHA pages where the visitors were inputting random repeated textual content into text boxes. APEs should adopt similar approaches for the web sensor APIs as well. A much simpler and complementary approach to eliminate this entire attack surface is for Android or Chrome developers to disable this default configuration of allowing websites access to these sensor APIs. This action which will be in line with other mobile OSs (such as iOS) and browsers (such as Firefox) will also have the added benefit of improving user privacy as prior research has shown that these APIs can be used for fingerprinting attacks [11].

Other attacks. While most of the above evasive attacks were mainly focused on the human analysis systems, our data analysis also revealed some weaknesses in what are most likely the automated crawler systems being used by APEs. For example, all of GSB's visits that were solving Popup challenges from non-India IP addresses were solving the challenges in less than a second. This is in sharp contrast to the solution times for GSB's human analysts as well as our user study participants that took at least 10 seconds. Such timing discrepancies can easily be utilized to identify APE bot IP addresses and them to a blocklists in phishing kits as is often done in the wild [18]. Similarly, we notice that all of the APEs ecosystems (human+bot) continue to lack heavily in diversity of browser fingerprints which keeps them prone to evasive attacks as proposed in [10]. We avoid discussing this in detail here as similar issues that affect all APE ecosystems have been tackled in earlier works. Instead, we chose to only focus on those issues that largely affect the human analysis systems of APEs in this paper.

5 Discussion

Conservative estimates. In this paper, we primarily relied upon the event of whether a visitor is solving a particular CAPTCHA challenge in order to deduce if that visitor is human. However, there might be human analysts who do not solve

some CAPTCHA challenges as a result of a notion that our pages are not suspicious enough to warrant inspection. We thus concede here that the numbers presented in Sec. 3 might be a lower-bound on the actual size of human analyst systems being used by APEs. This only means that the APEs are spending even more resources to host such analysis systems than what was implied earlier and is hence even more important to take measures to protect them form evasion attacks.

Further, since the cases in which analysts are solving CAPTCHA challenges can be considered a random subset of all cases of human visits, we expect the same outcomes as in Sec. 4 when analyzing the set of all human visits as well. In order to demonstrated this, we used GSB as a case study. We repeated the browser fingerprint-based clustering we described earlier for all visits from GSB irrespective of whether the visit resulted in a CAPTCHA solution. 3 of the 4 clusters that we determined in Sec. 3 to be coming from humans have now "expanded" on their size and included several visits where the challenges were not solved but shared the same fingerprints. In total, these clusters were covering 258 distinct sites (in 295 visits) instead of the 168 distinct sites where challenges were solved thus indicating an even healthier human-analysis rate of 37%. However, as expected, all the weaknesses remain the same with this expanded dataset of probable human visits as well. For example, all the 295 visits were from Indian IP addresses with Chrome OS user agents none of which have happened over the weekends.

Industry disclosure. We have conducted an elaborate industry disclosure process with all four affected APEs. As part of this, we have disclosed all of our findings in the form of detailed reports describing our experimental procedures and the weaknesses we discovered in their human analysis systems. In the case of Google, we have also disclosed information about the evasive phishing attacks we were able to launch against the Android/Chrome ecosystems which can easily be addressed by turning off the default emission of web sensor API data similar to other platforms. The response from APEs has been positive with one APE mentioning that multiple internal bug reports have been filed as a result of our disclosures.

Ethical considerations. Our APE evaluation setup involved sending honey site URLs to APEs. We limited these submissions to only about 20 per day which is much smaller in comparison to the large number of suspicious URLs vetted by these APEs everyday. Further, based on our server logs, we estimate that the total time spent by the human analysts of all the APEs vetting our CAPTCHA-laden honey pages is only about 1.2 hours. We thus argue that the small temporary overhead experienced by the APEs during our experimental period far outweighs the security benefits gained by the insights we present in this study. Our setup is also similar to prior studies on APEs such as [17,19,10] all of which involved submitting similar honey URLs to APEs. Similar to these prior works, we made the phishing pages described in Sec. 4.4 non-functional in order to prevent accidental sensitive information input from random visitors. Finally, our user study received exemption from the university IRB board and we also sought prior approval from the participants describing all the data collection methodologies before directing them to the CAPTCHA challenge pages.

6 Related Work

As discussed in Sec. 1, despite phishing being an old problem, only recently has the research community begun to focus its attention on studying the robustness of APEs with studies such as [17,19,13,23,10]. Among these, only [13] (and a small part of [23]) have tried to focus on the ability of APEs to overcome CAPTCHA-based cloaking challenges. However, they have concluded that APEs were largely incapable of solving such challenges. In sharp contrast to this, our study showed strong evidence that multiple APEs do in fact have ability to solve a large portion of such challenges there by indicating presence of elaborate human analysis systems complementing their automated crawler infrastructure. We attribute this difference to either the gap in the timelines of these studies during which time APEs could have potentially improved or the larger scale of our evaluation experiments. We relied on a recently proposed APE evaluation methodology [10] for achieving this scale. Our discovery of human analysis systems in the APEs thus allowed us an opportunity to conduct the first systematic study of the robustness of the human analysis systems of APEs which revealed several weaknesses in these systems as presented in Sec. 4.

7 Conclusion

We conducted a large-scale study that tests the ability of 7 popular APEs to overcome CAPTCHA-based challenges. Through this, we provide strong evidence for the presence of an elablorate human analysis system in 4 of the 7 APEs we studied. These are: GSB, SmartScreen, Bitdefender and Netcraft. While this measurement bodes well for the web security community, unfortunately, our study went on to show some grave weaknesses which can be abused by future attackers to launch evasive attacks against these elaborate human analysis systems. Interestingly, many of these weaknesses can be easily mitigated and are already being done so in most of the automated crawler systems of these APEs. We thus provided recommendations to APEs for doing the same with the expensive and vital human analysis systems as well. Our work in this paper is therefore crucial to help improve the current security posture of Anti Phishing Entities (APEs).

Acknowledgements: This work was inspired by a comment from an anonymous reviewer at IEEE SSP 2021 where we submitted our prior work [10]. We thank Wingate Jones for help in exploring AI-based techniques to evade APEs. We also thank all the anonymous reviewers for their very helpful feedback. This work was partly supported by the National Science Foundation (NSF) under grant CNS-2126655.

References

1. Free CAPTCHA-Service. http://captchas.net/
2. Google transparency report. https://transparencyreport.google.com/safe-browsing/search, [Accessed 13-Jan-2022]
3. Math Captcha. https://www.jotform.com/widgets/math-captcha
4. Puppeteer. https://github.com/puppeteer/puppeteer, [Accessed 13-Jan-2022]

5. ReCAPTCHA demo. https://www.google.com/recaptcha/api2/demo
6. Selenium. https://www.selenium.dev/, [Accessed 13-Jan-2022]
7. Statscounter: Browser market share. https://gs.statcounter.com/browser-market-share
8. User-agent switcher and manager. https://chrome.google.com/webstore/detail/user-agent-switcher-for-c/djflhoibgkdhkhhcedjiklpkjnoahfmg
9. User-agent switcher and manager. https://addons.mozilla.org/en-US/firefox/addon/user-agent-string-switcher/
10. Acharya, B., Vadrevu, P.: Phishprint: Evading phishing detection crawlers by prior profiling. In: 30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021. pp. 3775–3792. USENIX Association (2021)
11. Das, A., Borisov, N., Caesar, M.: Tracking mobile web users through motion sensors: Attacks and defenses. In: NDSS (2016)
12. Goodfellow, I.J., Bulatov, Y., Ibarz, J., Arnoud, S., Shet, V.D.: Multi-digit number recognition from street view imagery using deep convolutional neural networks. In: 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings (2014)
13. Maroofi, S., Korczynski, M., Duda, A.: Are you human?: Resilience of phishing detection to evasion techniques based on human verification. In: IMC '20: ACM Internet Measurement Conference, Virtual Event, USA, October 27-29, 2020. pp. 78–86. ACM (2020)
14. Maroofi, S., Korczyński, M., Hesselman, C., Ampeau, B., Duda, A.: Comar: Classification of compromised versus maliciously registered domains. In: 2020 IEEE European Symposium on Security and Privacy (EuroS&P). pp. 607–623. IEEE (2020)
15. Miramirkhani, N., Starov, O., Nikiforakis, N.: Dial one for scam: Analyzing and detecting technical support scams. In: 22nd Annual Network and Distributed System Security Symposium (NDSS. vol. 16 (2016)
16. Mowery, K., Shacham, H.: Pixel perfect: Fingerprinting canvas in html5. Proceedings of W2SP **2012** (2012)
17. Oest, A., Safaei, Y., Doupé, A., Ahn, G.J., Wardman, B., Tyers, K.: Phishfarm: A scalable framework for measuring the effectiveness of evasion techniques against browser phishing blacklists. In: 2019 IEEE Symposium on Security and Privacy (SP)
18. Oest, A., Safei, Y., Doupé, A., Ahn, G.J., Wardman, B., Warner, G.: Inside a phisher's mind: Understanding the anti-phishing ecosystem through phishing kit analysis. In: 2018 APWG Symposium on Electronic Crime Research (eCrime). pp. 1–12. IEEE (2018)
19. Peng, P., Yang, L., Song, L., Wang, G.: Opening the blackbox of virustotal: Analyzing online phishing scan engines. In: Proceedings of the Internet Measurement Conference. pp. 478–485 (2019)
20. Roy-Chowdhury, R.: Google: How we keep you safe online every day. https://blog.google/technology/safety-security/how-we-keep-you-safe-online-every-day/ (2020)
21. Vadrevu, P., Perdisci, R.: What you see is not what you get: Discovering and tracking social engineering attack campaigns. In: Proceedings of the Internet Measurement Conference. pp. 308–321 (2019)
22. Ye, G., Tang, Z., Fang, D., Zhu, Z., Feng, Y., Xu, P., Chen, X., Wang, Z.: Yet another text captcha solver: A generative adversarial network based approach. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018. pp. 332–348. ACM (2018)
23. Zhang, P., Oest, A., Cho, H., Sun, Z., Johnson, R., Wardman, B., Sarker, S., Kapravelos, A., Bao, T., Wang, R., et al.: Crawlphish: Large-scale analysis of client-side cloaking techniques in phishing. In: 2021 IEEE Symposium on Security and Privacy (SP)