Surveylance: Automatically Detecting Online Survey Scams

Amin Kharraz* †, William Robertson*, Engin Kirda*
*Northeastern University
†University of Illinois Urbana-Champaign

***Abstract*—Online surveys are a popular mechanism for performing market research in exchange for monetary compensation. Unfortunately, fraudulent survey websites are similarly rising in popularity among cyber-criminals as a means for executing social engineering attacks. In addition to the sizable population of users that participate in online surveys as a secondary revenue stream, unsuspecting users who search the web for free content or access codes to commercial software can also be exposed to survey scams. This occurs through redirection to websites that ask the user to complete a survey in order to receive the promised content or a reward.**

In this paper, we present SURVEYLANCE, the first system that automatically identifies survey scams using machine learning techniques. Our evaluation demonstrates that SURVEYLANCE works well in practice by identifying 8,623 unique websites involved in online survey attacks. We show that SURVEYLANCE is suitable for assisting human analysts in survey scam detection at scale. Our work also provides the first systematic analysis of the survey scam ecosystem by investigating the capabilities of these services, mapping all the parties involved in the ecosystem, and quantifying the consequences to users that are exposed to these services. Our analysis reveals that a large number of survey scams are easily reachable through the Alexa top 30K websites, and expose users to a wide range of security issues including identity fraud, deceptive advertisements, potentially unwanted programs (PUPs), malicious extensions, and malware.

I. INTRODUCTION

The growth and popularity of the Internet has brought enormous benefits to the marketing research industry. Targeted marketing surveys bring in more than $21 billion in annual revenue [13] by providing insights into what customers are thinking in a specific business sector. To reach potential customers, marketing research companies strive to directly communicate with end-users by conducting online surveys. Such surveys establish a reliable communication channel with consumers and allow companies to analyze the value of a product, perform pricing research, predict demand, or develop effective marketing strategies. In fact, most Internet users will have received a survey request via email at one time or another. To encourage users to take the time and fill out a survey, a common technique is to promise rewards in the form of gift cards, free access to popular online services, or free electronic gadgets such as an iPad.

While such online surveys are very useful for marketing research companies, unfortunately, attackers have also discovered online surveys as a profitable attack vector against users. In fact, there have been reports of online survey scams that aim to recruit unsuspecting users and trick them into releasing sensitive information [14], [28]. In particular, a technically-unsophisticated user who searches online for free content such as video streaming channels, access codes to popular software programs, or free gift cards may end up on these pages, and may be tricked into willingly providing sensitive information to attackers with the hope of acquiring interesting content. In addition, the attackers may also engage in illegal activities such as infecting the victims with malware.

It is known that online survey scams are actively being used by attackers. However, the details of the attacks that are launched, the modus operandi of the attackers, and the reports on the different social engineering tricks the attackers are deploying are mostly anecdotal. While there do exist some reports about such scams by security companies [34], [25], [40], [41], these reports have only looked at a handful of online survey scams, have only provided a manual analysis, and have not followed any systematic or scientific processes.

Understanding how online survey scams work in the real-world is important, and a careful, systematic analysis is beneficial for a number of reasons. First, we can understand how these websites are operated and shed light on the malicious practices used by attackers to monetize these scams. Second, based on the understanding we have gained, we can develop better techniques for detecting survey scams automatically, and prevent users from falling victim to such attacks.

In this paper, we conduct several empirical studies to identify the main participants in the survey scam ecosystem as well as the common malicious practices used by scammers to launch successful attacks. We leverage this knowledge in the development of a new system, called SURVEYLANCE, that is able to identify survey scams using machine learning techniques. The scalable, distributed infrastructure we have built allowed us to perform a long-term experiment by running SURVEYLANCE for several months to collect data on online survey scams.

During this period, SURVEYLANCE identified 8,623 websites, called survey gateways, that directed victims to 318,219 online survey scam pages. The experiments show that SURVEYLANCE works well in practice by achieving a true positive [TP] of 94.8% with 1.2% false positives [FPs]. We used the output of SURVEYLANCE and performed more than 380,000 visits to the detected survey gateways using a number of browser profiles to investigate the security implications of online survey scams on typical users.

The analysis of the extracted dataset reveals several concerning facts about survey scams on the Web. For example, our empirical analysis shows that more than 40% of the survey scams are reachable from the Alexa top 30K websites. Our findings illustrate that survey gateways actively fingerprint victims before redirecting them to survey scam websites – called *survey publishers*. These websites attempt to identify users, prompt customized messages, dynamically select offers based on user location, and store specific data about the user for future visits.

Our large-scale experiment also shows that survey scams expose users to a wide range of security issues including identity fraud, deceptive advertisements, potentially unwanted programs (PUPs), scareware, and malware. The families with the most significant distribution campaigns include PUPs like Somoto, Amonetize, InstallCore, or malware that deceived users to install adware or pay a subscription fee for resolving a critical security issue on their machines. Our investigations reveal that survey scammers host their infrastructures primarily in Brazil, Eastern Europe, and Russia.

To our knowledge, this work is the first comprehensive study of the online survey scam ecosystem. Perhaps the most important finding in this paper is practical evidence that shows that survey scams are a serious and under-explored security threat. These attacks are designed to specifically target end-users rather than vulnerable systems. Therefore, developing security tools that can decrease the users' exposure to these attacks is vital. We show that scammers monetize their operations mainly by distributing PUPs and deceptive advertisements. In this ecosystem, ad networks receive legitimate traffic and everyone, including the scammers, advertisers, and ad networks profit. Unfortunately, the victims instead suffer a virulent impact on their security and privacy.

In summary, this paper makes the following contributions:

- We develop SURVEYLANCE, a tool that detects websites involved in survey scam services. While this class of attacks introduces similar threats to more traditional web-based attacks such as phishing or malware websites, it is less well-studied, and few or no details exist on how to detect these attacks. Our large-scale evaluation demonstrates that our technique works well in practice, and achieves promising results (a true positive [TP] rate of 94.8% with 1.2% false positives [FPs]).
- We identify multiple entities involved in the survey scam ecosystem by incorporating SURVEYLANCE to passively crawl websites, collect network traces, and perform classification. We envision multiple deployment models for SURVEYLANCE to disrupt the operation of survey scams. The output of SURVEYLANCE can be used to augment blacklists or anti-malware mechanisms (e.g, Google Safe Browsing) in major browsers to reduce the exposure of users to these websites. Furthermore, the trained model can be incorporated as a browser extension that monitors the content of the visited websites, and notifies the user whether the website presents a threat.
- We used the output of SURVEYLANCE, and performed a long-term study on the malicious practices used in online survey scams. We show that end-users are exposed to a wide range of threats such as identity fraud, PUPs, malware, and scareware. We also show that these entities use several techniques to monetize their businesses (e.g. injecting overlay ads, pop-ups). The results of our empirical study also shows that a significant number of these websites are easily reachable. For example, more than 40% of the survey gateways which redirect users to survey scam pages are reachable through the Alexa top 30K websites.

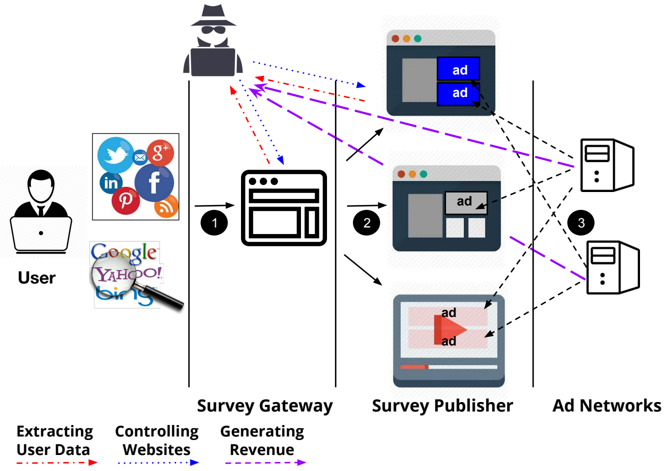


Fig. 1: The ecosystem of survey scam services. (1) the survey scam gateway requests information (e.g., sensitive user data) from the user such as a social security number, (2) it redirects the user to a survey publisher website, and (3) survey publishers may use ads in addition to extracting information from users.

The rest of the paper is structured as follows: in Section II, we provide background information on the survey scam ecosystem. In Section III, we set the design goal, discuss the feature set we used in SURVEYLANCE as well as the implementation details. Section IV describes our data collection methodology. In Section V, we evaluate the detection capabilities of our approach. An analysis of our measurements is presented in Section VI. We discuss the limitations of our approach in Section VII, and present related work in Section VIII. Finally, Section IX concludes the paper.

II. BACKGROUND

In this section, we set the stage for our work by providing background on survey scam services as well as defining the key terminology.

A. *Online Surveys*

Today, it is not uncommon to receive requests to complete a survey from legitimate, well-known businesses – for instance, an airline [36], [27]. Content publishers can also insert third-party scripts that invite users to fill out a survey. Survey owners also employ targeted advertisement services [9] on social media to distribute invitations to users. While it is common among businesses to conduct surveys to access direct feedback from users, adversaries can also use the same concept to direct users to domains under their control.

B. *Survey Scam Services*

In this paper, we focus on cases where a victim is exposed to a survey scam by issuing specific search engine queries, or by clicking on links designed to lure them to a scam (e.g. to watch a newly released movie, a live sports event, or to download free content). After filling out the survey, the user does not receive the promised content, but is instead exposed to one or more malicious activities. This can include leakage of sensitive personal information, redirection to other malicious pages, downloading potentially unwanted programs (PUPs), or being exposed to malware. Figure 1 shows an overview of the survey scam ecosystem which consists of three main participants: survey gateways, survey publishers, and advertisers. We arrived at this model through a manual analysis of several survey scams. We describe the role and monetization model for each party below.

Survey gateways are primarily designed to convert a visiting user to a potential victim by encouraging him to accept a survey request. Furthermore, a survey gateway serves as a scheduler which assigns a survey to the visiting user, similar to a typical ad network that matches ads to users. Our investigation confirms that these websites attempt to identify users, customize the messages shown to users, and store specific data for future visits using the browser LocalStorage APIs (see Section B). An unsuspecting user may proceed to fill out a survey for several reasons. She may complete a survey to receive the free version of the content she was looking for in the first place, or to receive rewards after accepting an appealing invitation during her browsing session.

Survey publishers supply tailored surveys to survey gateways. Once the user agrees to participate in the survey, the gateway redirects the user to a page that publishes the survey and asks the user to proceed. In addition to collecting sensitive user information (e.g. home address, phone number) which can be exploited in other adversarial activities, survey publishers earn money by driving users to ad-tracking sites or affiliate programs [28] before actually proceeding to the survey. However, unlike traditional publishers that have to generate content to attract users, survey scam services attract users with enticing but fake rewards; extract sensitive information; and expose users to malicious binaries and rogue advertisements which defraud users and monetize adversaries' businesses at the expense of users' security.

C. A Motivating Example

When the user visits a survey gateway, she is requested to provide some information such as age, gender, or email address. The survey gateway then directs the user to a survey publisher. In Section B, we show that this operation relies on identifying users and generating new IDs to prompt customized messages or dynamically select enticing offers. After completing the survey, the user is asked to claim her reward by following a link. This link can redirect the user to an affiliate program, a scam page, or a malware website. For example, clicking the button on **10086sjt.com** redirected us to another website that delivered a sample of **PUP.Optional.LoadMoney**.

Note that the example we discussed here is not synthetic and is a simplified version of this specific web-based social engineering attack. Our experiments, discussed at length in VI, show that these attacks occur quite frequently. In this ecosystem, all the involved parties, including ad networks, scammers, and advertisers, profit. Unfortunately, the victim instead suffers a negative impact on their security and privacy. Figure 2 illustrates a case study of online survey scams and how a user is exposed to malicious advertisements or malware.

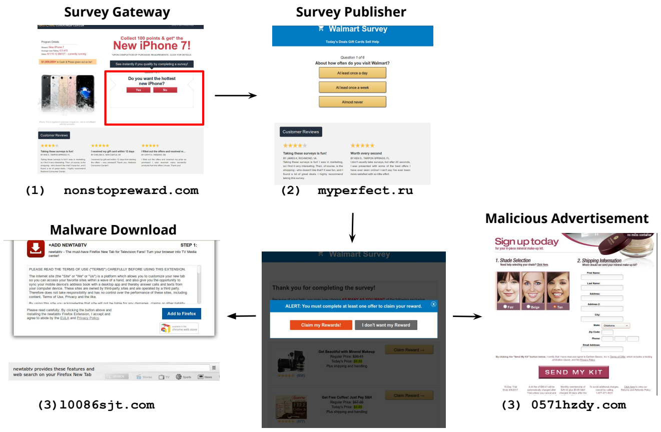


Fig. 2: A case study of online survey scams. The survey gateway (1) asks a set of questions from the victim, and (2) redirects the victim to a survey publisher website. (3) After completing the survey, the victim is exposed to advertisements or malware.

III. SURVEYLANCE

In this section, we describe the architecture of SURVEYLANCE by providing details on feature extraction, classification, and our prototype implementation.

A. Overview

SURVEYLANCE uses a classification model that is specifically designed to distinguish survey gateways from normal websites including benign survey pages. We target survey gateways as they are usually the entry point to several different survey publishers. One of the design requirements of the system is to avoid relying on easily-evadable features such as IP addresses or domain names, and instead incorporating features that directly target the conceptual operations of survey scams (e.g., types of inputs, particular images). These features rely on the look and feel of these sites, their source code, HTTP requests, HTTP responses issued to and received from these sites, as well as the redirection chains involving these sites.

B. Feature Set

To construct a detection model, we rely on extracting information from the content of websites, network traffic, widgets, and the overall presentation of the page. In the following, we provide more details on the features, and our intuition for choosing them.

1) Indicative images: The main page of survey gateways is usually well-designed and is comparable to a typical user-friendly website. Adversaries usually make extensive use of images to encourage users to fill out a survey – such as logos that indicate *Satisfaction Guaranteed* or *Guaranteed Income*. The presence of such images, along with other features, can be a good indicator of survey gateways. To employ this indicator as a feature, we extracted all the images from our labeled survey scam dataset and clustered them using a perceptual hash function [53]. We first used structural similarity testing [49] as our image comparison technique. However, we found that the perceptual hash function was more robust to small image changes as the images in survey gateways are often small and their representation often stays intact. SURVEYLANCE computes the centroid perceptual hash value as the centroid of the cluster, which is representative of all the images in the cluster. For a given website, after extracting all the included images, SURVEYLANCE computes the perceptual hash value of each image, and compares the value with the centroid of the perceptual hash values of all the clusters. If the Hamming distance between two perceptual hash values is less than 0.17 (an experimentally-derived threshold), we label the image as indicative of a survey gateway. We then report the number of indicative images and incorporate this value as a numeric feature. The intuition here is that the pages that contain more indicative images are more likely to be survey gateways.

2) User input fields: Survey gateways usually require users to enter their personal information such as the home address, employer, email address, or phone number, using textfields before redirecting them to a particular publisher. SURVEYLANCE extracts the total number of textfield input tags in the website. The rationale is that while survey gateways claim that surveys are anonymous, they nevertheless attempt to elicit Personally Identifiable Information (PII) from users. This is borne out in our experiments which show that more than 83% of samples had at least four textfields that required sensitive information including Social Security Number and payment information. SURVEYLANCE reports the total number of textfields as a numeric feature.

3) Third-party scripts: We expect benign pages, including benign survey pages, to have a lower ratio of third-party inclusions to decrease the risk of unwanted information leakage to third-parties. In fact, third-party scripts have, by default, full control over the content of pages in which they are included. Such code can potentially inspect and modify values that a local JavaScript would be able to do. We performed an analysis on the number of third-party inclusions on labeled survey gateways and benign survey pages (see Appendix A). Our analysis shows that survey gateways include a significantly larger number of third-party scripts (e.g., advertisements). SURVEYLANCE looks for third-party script inclusions in the HTML code of survey gateways and uses the ratio of third-party links to the total number of links as a feature.

4) Link length: Our manual analysis shows that survey gateways use advertisements as a major source of revenue. We extract all <a> HTML tags and calculate the length of the string in each link. The intuition in using this feature is that these links tend to pass more and longer parameters which are mainly used to track clicks, fingerprint users for ad-retargeting, and carry the publisher ID or LocalStorage keys and values. We calculate the mean and maximum link length of third-party links.

5) Website structure: While the main page of survey gateways is often presented convincingly, these websites are usually undeveloped since they are solely designed to expose users to security threats. As a consequence, it can be the case that these websites do not follow common practices, such as defining dedicated directories for specific purposes. For example, a typical website is comprised of HTML, CSS, image, and JavaScript files in different folders forming a directory tree. Therefore, the system searches the source code and finds indications of directory presence. For this goal, SURVEYLANCE extracts all the local inclusions and parses the inclusion paths. The system uses this as a boolean feature in our detection where the value 1 means that the contents of the website are structured and maintained as a directory tree.

6) Web content: Survey gateways usually do not contain a large volume of content, and mainly include enticing images to lure users into filling out a survey. A large fraction of text in survey gateways is, in fact, the included URLs that are not visible to users. SURVEYLANCE computes the ratio of the volume of text in the links' <a> HTML tags to the total volume of text in the page. We use this ratio as a numeric feature.

7) Sequence of words: SURVEYLANCE seeks the presence of particular word sequences that are indicative of survey gateways. The intuition is that sequence of words that appear more frequently in survey pages than in non-survey pages can be used to mark potential survey scam websites – for example, *guaranteed reward*, *easy income*. To this end, we generate n-grams by varying the length of n from $n = 2$ to $n = 6$ out of any text found in the main body of labeled survey gateways. We then select the most prevalent n-grams by measuring their importance in the labeled dataset using Term Frequency – Inverse Document Frequency (TF-IDF) [32]. SURVEYLANCE measures the frequency of the selected n-grams and incorporates these frequencies as a set of numeric values representing the frequency of n-grams ($1 < n \leq 6$).

8) Redirection mechanisms: Survey gateways use redirection mechanisms to redirect users to other websites. In fact, benign websites may use the same mechanism to enable load balancing or fault tolerance, but our experiments show that when this feature is considered together with the other features, it can enhance the classification results. We report the value of this feature as a boolean value where 1 means that redirection was observed while visiting the website.

9) Third-party requests: SURVEYLANCE calculates the ratio of the number of HTTP requests to third-party domains and the total number of HTTP requests.

10) Third-party responses: In addition to the number of HTTP requests, SURVEYLANCE calculates the ratio of the incoming traffic from third-party sources and all the incoming traffic. This is a particularly useful feature when survey publishers include ads on embedded videos.

11) Image size: It is very common in fraudulent survey websites to include several logos to create the impression that the website is legitimate. Furthermore, these websites include images of specific rewards or offers that a user can claim after completing the survey. We observe that these images are often small (on average 2 KB) in most of survey gateways. Therefore, we measure the mean and maximum size of images found on a webpage, and incorporate these measurements as a set of numeric features.

12) Number of frames: During the course of manual experiments, we noticed that the presence of `<iframe>` elements is very common in survey gateways to embed videos or pop-ups to show advertisements. SURVEYLANCE extracts all `<frame>` and `<iframe>` elements present on a page and its child frames, and incorporates this data as a numeric feature by counting the number of frames and iframes.

C. Prototype Implementation

SURVEYLANCE consists of three independent modules: (1) a crawling module which manages browser instances and serves as a data extractor, (2) a classification module which assigns a label to a given URL, and (3) a form filler module that automatically completes surveys in survey publishers. In the following paragraphs, we provide more details on the implementation details of each module.

1) ***Crawling Module****:* To manage browser instances, we developed a scheduler which is responsible for instantiating the browser instances with a pre-specified configuration setting. It also assigns a crawling job to each browser instance which consists of 10,000 websites. The scheduler restarts each browser instance after completing the crawling job to reduce the potential risks of a compromised browsing instance. In addition, we modified the user-agent properties of the browser instance to emulate a typical user browsing the web using a Microsoft Windows OS. We developed a custom Chrome extension which relies on the Chrome debugging protocol, and operates on top of the DevTool Extension API [10] . This approach provides instrumentation, inspection, and profiling of Chromium and enables us to access all the functionality of DevTool as well as DOM and DOM Events of a page for data collection. The custom extension allows us to have nearly full coverage of browser interactions with a given website in order to collect HTML source code and network traces to construct redirection chains – the paths that show how a user is exposed to a survey scam. We use the collected data to construct the feature vectors and perform classification using the classification module.

To increase the level of interaction with websites while visiting a page, the crawler scrolls downwards to trigger the loading of any dynamic content. The crawler remains on each page for a fixed period of 60 seconds before clearing its entire state, restarting, and proceeding to the next site. We updated the browser extension to automatically find the required fields, and populate the fields with the data that satisfies each element type in a given survey page. We present more details on this extension in Section III-C3.

2) ***Classification Module****:* Our approach requires construction of a classifier that can analyze the data collected by the crawler module, and reliably detect survey gateways using the features described in Section III-B. That is, the classifier should take a URL as an input, build a feature vector from the crawled data, and assign a label showing whether a page is a survey gateway.

To construct the detection model, we first need to select an appropriate learning algorithm that minimizes the false positives. Furthermore, it should be efficient in the detection phase to avoid impacting the performance or scalability of the endpoints. To this end, we tested multiple classification algorithms, and found that a random forest [5] classifier produced the best detection results. In fact, the random forest classifier tended to be more robust than other models with respect to outliers, and was relatively more efficient in the detection phase. To construct the classification model, we used the random forest implementation provided by scikit-learn [35]. As mentioned in Section III-B, our approach requires extracting visible texts from a given URL. To extract the natural language data presented in a given HTML page, SURVEYLANCE uses Python Natural Language ToolKit (NLTK) [26]. SURVEYLANCE then extracts anchor elements using PyQuery [29], and calculates the length of textual content, length of textual content in links, and the total length of the original HTML source code.

SURVEYLANCE implements HTTP Archive 1.2 specification [43], and stores the network traffic traces as HAR objects where each entry has timing, request, and response information. During the classification phase, SURVEYLANCE parses each HAR object, and analyzes the HTTP requests and responses. The system traverses over data objects by applying a regular expression to differentiate between first and third party requests and responses, and calculate the corresponding feature values such as HTTP request and response size.

3) ***Survey Filler Module****:* As a part of our experiments, we use the detected survey gateways to reach survey publishers, and study the types of threats to which a user may get exposed. To interact with the survey publishers and populate the survey forms with appropriate data, SURVEYLANCE uses a form filler module that injects content scripts in the context of the visited web page. SURVEYLANCE first retrieves the identifier of the inspected window, and sends a message to the background page which calls `executeScript` to run the form filler module. The extension finds all inputs (i.e., `input`, `textarea`, `select` in `'XXXX:enabled:not([readonly])'` as well as `'[contenteditable]'`). Here, *readonly* is a selector that simply checks if the corresponding attribute is defined on JQuery elements. After identifying the element type of the input by using JQuery element (`jQueryElement.attr('type')`), the extension decides how to generate the input. To this end, the extension seeks to find a pre-defined set of keywords in the ID of the corresponding input fields; and calls the appropriate input generation function based on the detected keyword. The pre-defined set of keywords is constructed per input field by observing a set of survey forms in the labeled dataset. Unsurprisingly, handling different types of input fields is a non-trivial task, and we had to manually verify some of the fields to be able to create the pre-defined set of possible IDs that developers might use in each website. At a high level, we defined at least two attributes for most of the input fields. The first attribute, referred to as *match*, was the pre-defined list of possible IDs that a website uses for a specific input field, and the second attribute was *sanitized_name* which we used to call the corresponding method that generates the input. For example, if the extension finds a field that contains *'integer', 'numeric', 'number', 'qty', 'price', 'quantity', 'total'*, it calls the *number generator* module with a pre-specified value range. Since some of the input fields required a specific format (e.g, `MM-DD-YYYY`) or a value range (e.g., age) to pass the registration phase, we were careful to generate inputs that abide by these constraints. We also noticed a number of hidden fields or CAPTCHAs in some of survey publisher websites, and decided to ignore such cases. Our form filler handled most of the potentially required element types in the registration pages such as checkboxes, dates, email addresses, radio buttons, texts, URLs, and elements of similar nature. For `textarea` elements, the form filler randomly generated a string with a maximum length of 30 characters. For websites that required a user registration, we created a set of credentials that pass most username and password selection policies. However, there were several cases that we were not able to cover due to almost unlimited value possibilities for input field IDs which are determined by developers of websites.

IV. Data Collection

In this section, we discuss our data collection methodology to conduct the experiments, and evaluate the effectiveness of Surveylance.

A. Sources of Survey Gateways

Constructing a reliable source of labeled data to run our experiments was quite challenging as there was no central repository, blacklist, or previous large-scale analysis in this specific area. One of the first questions that arises is: How are end-users redirected to online survey scam pages? There is evidence that users are usually directed to web-based social engineering attacks, including online survey scams, by being exposed to malicious advertisements, as shown by recent studies on malvertising [22], [44], [51] and social engineering attacks [23]. Note that scammers could trick users into clicking on direct links to scam pages. However, this approach would result in a shorter active lifetime in light of increasing detection capabilities of search engines and blacklist operators. Furthermore, this approach may not be as scalable as leveraging malicious advertisements where ads can be simply included into several independent websites, and be delivered to millions of users.

Therefore, in this paper, instead of directly searching for survey gateways or publishers, we use a more generic approach. More specifically, we search for websites that are more likely to include malicious advertisements, and that are the representative of what a typical user may be redirected to in normal browsing sessions. These websites can be used as the starting point of different types of social engineering attacks including online survey scams. We take advantage of the findings of specific recent studies on web-based social engineering attacks [23], [18], [46], and look for websites that leverage a combination of *deception* and *persuasion* to attract users while taking part in malwaretising practices.

We specifically search for pages that attempt to attract normal users by embedding enticing content and encouraging users to make risky decisions (e.g., clicking on a link, downloading a file). To this end, we incorporated the *Google Trends* service to construct a set of popular items in various search categories. Note that we are not claiming that the Google Trends service exposes users to particular websites such as survey gateways. Instead, we use the trendy keywords as gateways to malicious advertising since scammers traditionally target technically-unsophisticated users who usually search the Web for free access to popular resources [23], [46]. *As these keywords are used by real users around the world for different purposes and are indexed based on their popularity, they can be referred to as a representative set of what real users may search online.* Our approach to collect an initial set of survey gateways is similar to prior work [31], [12] that leverages the infrastructure of search engines to find malicious webpages. However, in this work, we used *Microsoft Cognitive Services*, which provide a web search API [21] to programmatically search and retrieve the search results.

We generated a list of the 1,000 most popular searched items covering multiple categories such as business, technology, and sports. We extracted the first 50 search results for each search query, and collected 5,173 unique websites after processing the search results. For example, the search term *"Harry Potter – Novel Series"*, which was indexed as a popular searched item, led us to `scanlib.com`. We found four different survey gateways, each of which asked us to complete surveys and receive *Amazon Kindle Coupon* and *Costco* gift cards. In order to build the initial set of survey gateways, we crawled these websites by clicking on the links, recording the redirection chains, and taking a screenshot of the landing page.

We ran this experiment two times by disabling the Google Safe Browsing (GSB) mechanism in the first run, and enabling it in the second run. Our intuition was that the users' exposure to web-based social engineering attacks including survey scams should diminish in the presence of the GSB. Note that the GSB does not specifically identify survey scams; rather, it protects users from being exposed to suspicious links that can lead them to several types of security threats. We were able to confirm 1,538 websites in the GSB-disabled mode, while in the GSB-enabled mode we identified 704 survey gateways. *Since we observed a noticeable difference in the number of manually-confirmed survey gateways between the two experiments, we left the GSB enabled as a real-world browser setting for the rest of the experiments in this paper.* Furthermore, we used only the survey gateways detected in the GSB-enabled experiment as our initial seeds.

B. Sources of Benign Survey Pages

To collect benign survey pages, we first created a list of 20 reputable survey services that are constantly ranked among the Alexa Top 20K websites [2]. Next, we crawled the main page of the Alexa top 12K websites, and extracted any third-party links that belonged to the survey services in the list. We collected 2,457 benign survey pages and empirically noticed that news websites, reputable businesses, and online stores – that respectively constitute 47%, 35%, and 11% of the benign survey pages – are the main consumers of benign survey services. Table I shows the most common survey services that we observed in the Alexa top 12K websites.

V. Detection Evaluation

We evaluated Surveylance with two experiments. The goal of the first experiment is to demonstrate that the system can detect known survey gateways, while the goal of the second experiment is to demonstrate that Surveylance can detect previously unknown survey gateways. Figure 3 illustrates a high-level view of our experiments.

Popular Survey Services	#
mypoint.com	570 (23.2%)
mysurvey.com	533 (21.7%)
creationsrewards.net	427 (17.4%)
inboxdollars.com	317 (12.9%)
oneopinion.com	204 (8.3%)
swaybucks.com	174 (7.1%)
i-say.com	128 (5.2%)
others	103 (4.2%)
Total	2,457 (100%)

TABLE I: The distribution of benign survey services among the Alexa top 12K websites.

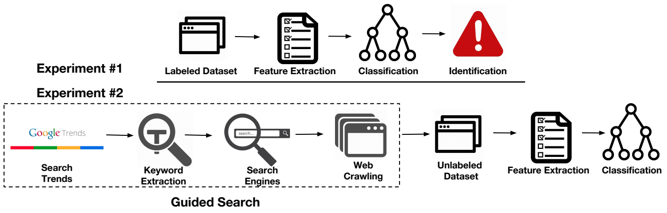


Fig. 3: A high-level view of the experiments.

A. *Constructing Labeled Dataset*

To evaluate the performance of the classifier, we created two different datasets that we carefully labeled. We now provide the details of each dataset.

a) ***Balanced Dataset (Set A):*** This dataset contains an equal number of survey gateways and benign survey pages. Our labeled dataset contains 700 survey gateways as well as 700 benign survey pages (see Section IV).

b) ***Imbalanced Dataset (Set B):*** In addition to the balanced dataset, we ran another experiment to evaluate the performance of SURVEYLANCE on an imbalanced dataset. We would like to test SURVEYLANCE with this dataset as, in reality, there are more benign websites than survey gateways, and an imbalanced distribution of the labeled dataset can bias the performance of the classifier towards the benign cases. To evaluate the performance of the feature set, we built a dataset with an imbalance ratio of 1 to 10 which contains 700 survey gateways and 7,000 benign pages. To collect the benign dataset, we used three types of benign webpages. First, we randomly selected 2,000 benign survey pages from the previously compiled list (see Section IV). Second, we added 2,000 pages out of 8,653 *registration pages* in the Alexa top 20K websites, and finally, we incorporated 3,000 random pages from 20K Alexa websites. *An evaluation of* SURVEYLANCE *on such an imbalanced dataset not only shows* SURVEYLANCE*'s ability to distinguish between survey gateways from benign survey pages, but also determines the classifier's performance on entirely different websites (e.g., cnn.com) presented to the classifier.*

We collected the benign pages from the Alexa top 20K websites since our assumption is that if a domain has consistently appeared in the Alexa top 20K websites for a year, it would most likely not be involved in malicious activities. The main intuition is that these websites are usually well-maintained and better protected against new attacks. We used the registration page of highly reputable websites as these pages often require similar types of information from the user, and can be considered as another type of relevant, benign cases. To find the websites that include registration and login pages, we crawled the candidate websites, and marked those websites that contained forms with the input type "password".

Metric	SVM		Random Forest	
	set A	set B	set A	set B
TPR	94.1 %	96.8%	95.8	97.7 %
FPR	2.8%	3.8%	0.6%	0.9 %
AUC	94.7 %	95.1%	97.9%	98.2 %

TABLE II: Results of a 10-Fold cross-validation on two classifiers, Support Vector Machines (SVM) and Random Forest (RF) using the labeled sets of A and B.

B. *Experiment #1: Testing* SURVEYLANCE *with the Labeled Dataset*

1) ***10-Fold Cross-Validation:*** To evaluate the detection accuracy of SURVEYLANCE, we performed a 10-fold cross-validation on the labeled dataset A and B. We ran the experiment using Support Vector Machines (SVM) and Random Forest (RF) to find out which classification algorithm achieves better results on the labeled datasets. We set the maximum number of trees in our RF classifier to 100 trees in order to mitigate over-fitting issues on the training datasets. As shown in Table II, SVM achieved an especially high detection rate on the imbalanced dataset B. However, we selected Random Forest as our default classifier to test the unknown dataset, since it performed relatively better on both the balanced and the imbalanced datasets.

2) ***Feature Ranking:*** We performed another experiment on the balanced dataset (Set A) to measure the relative contribution of the features used in our classification model. We used a *recursive feature elimination* (RFE) approach to determine the significance of each feature. We divided the feature set into three different categories: *Content-based*, *Traffic-based*, and *Image-based* features. The procedure started by incorporating all the features while measuring the FP and TP rates. Then, in each step, a feature with the minimum weight was removed, and the FP and TP rates were calculated to quantify the contribution of each feature. Table III ranks all the features with the most important one at the top. The capitalized letters in the second column indicates the feature categories: C for Content-based features, T for Traffic-based features, and I for Image-based features. For easier interpretation, we calculated the score ratio by dividing the score values with the largest one. The score ratio of each feature simply shows how much the corresponding feature can contribute to identify positive and negative cases in the labeled dataset. The results of the experiment show that 5 out of 6 content-based features are among the top ten features. This result is quite encouraging, as these features can easily be collected once the page is loaded into the browser instance, imposing less operational overhead on the classifier compared to traffic-based or even image-based features which mainly rely on third-party inclusions.

3) ***Classification Evasion:*** Similar to other defense mechanisms, adversaries may attempt to evade SURVEYLANCE. To

Rank	Cat	Feature	Type	Score Ratio
1	C	Sequence of words	Ordinal	100%
2	C	Number of user input fields	Ordinal	83.2%
3	I	Presence of indicative images	Ordinal	65.4%
4	C	Website content	Continuous	61.5%
5	C	Third-party script ratio	Continuous	33.6%
6	T	Link length mean	Continuous	28.5%
7	T	Page redirection	Categorical	27.3%
8	C	Web structure	Categorical	22.3%
9	T	Link length max	Ordinal	11.5%
10	I	Image size mean	Ordinal	8.3%
11	T	Third-party request ratio	Continuous	6.9%
12	T	Third-party response ratio	Continuous	6.1%
13	C	Number of frames	Ordinal	5.7%
14	I	Image size max	Ordinal	3.3%

TABLE III: Ranking of feature importance in SURVEYLANCE (C for Content-based, T for Traffic-based, and I for Image-based category).

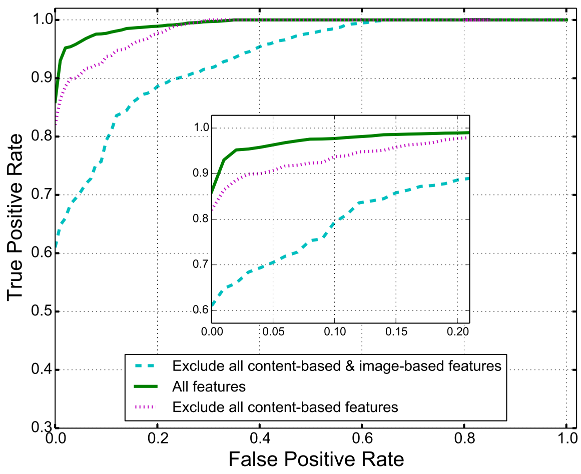


Fig. 4: Detection results of SURVEYLANCE on the labeled dataset.

this end, we evaluated SURVEYLANCE's performance under different evasion scenarios by excluding the corresponding features from the detection model. The results of the analysis are shown in Figure 4. The green curve represents the ROC curve of SURVEYLANCE which incorporates all the features into the detection model. The velvet curve exhibits the ROC curve if adversaries evade the traffic-based features, rank 6, 7, 9, 11, 12 in Table III. Excluding this feature set from the detection model is a reasonable assumption, as an adversary may avoid embedding third-party scripts in the survey gateways to evade SURVEYLANCE at the cost of not making any revenues from third-parties. As shown, SURVEYLANCE's performance degrades, but it still achieves a relatively high level of detection accuracy. This analysis suggests that the traffic-based features are important, but the system still achieves good detection results in the absence of traffic-based features.

We next considered excluding image-based features (rank 3, 11, 15) with the assumption that an adversary completely removes the images that were observed during our training process. In this case, SURVEYLANCE relies solely on content-based features (rank 1, 2, 4, 5, 8, 13) to capture survey gateways as adversaries need to incorporate relevant content to encourage visiting users to contribute in completing surveys. Failing to do that would make such attacks less friendly, and would impair the attack capability as fewer users may contribute in completing surveys and ultimately fall victim to such attacks. The blue ROC curve illustrates SURVEYLANCE's performance with traffic-based and image-based features excluded (8 features). We observe that the system achieves a lower detection accuracy, suggesting that image-based features help achieve lower false positives despite their relatively lower ranks. We believe that removing the enticing, indicative images that promise rewards could significantly influence the attackers' efficacy as they require an extra effort to create new pages with entirely new sets of images. In Section V-C3, we provide more details on how SURVEYLANCE should be re-trained to maintain the detection accuracy high. We conclude that the content-based features, as well as some of the features in image-based, traffic-based features (i.e., indicative images), and page redirection are the more reliable features of SURVEYLANCE for increasing the cost of evasion.

C. Experiment #2: Detecting Unknown Survey Gateways

In this experiment, we used the trained model in the previous experiment to classify URLs that have not been observed in the training phase. To create a new set of testing data, similar to our approach in the training phase, we made use of the Google Trends results. As mentioned earlier, we used the Google Trends for two primary reasons: (1) the Google Trends service relies on daily search queries generated by real users, so the list is a subset of real search queries in different categories; and (2) this approach minimizes the risk of the over-fitting problem, as the unlabeled dataset will be generated using keywords that have not been observed in our training phase. To this end, we collected English search terms for a period of 14 days. The searched items include a variety of topics such as business, technology, sports, and entertainment categories. We created a list of the 10,000 most popular search items (which were queried at least 300,000 times) to incorporate into our data collection process. After querying the search items using the Microsoft Web Search API, we collected 23,124 URLs from the search results. SURVEYLANCE extracted 2,301,733 third-party URLs in those pages, and visited each URL using the crawler module described in Section III-C. For each crawled webpage, a feature vector (see Section III-B) was extracted to assign a label to the page indicating the page relevance to a survey gateway. SURVEYLANCE reported 8,623 survey gateways by crawling 2,301,733 URLs. In order to further study the threat, we used the survey gateways to reach survey publishers, and analyze the types of threats that users may be exposed to by agreeing to complete a survey. Table IV exhibits the number of survey gateways as well as survey publishers we found in our experiments. We provide more details on survey publishers in Section VI.

*1) **Evaluating False Positives**:* Since we did not have a labeled ground truth in the large-scale experiment, we cannot provide an accurate precision-recall analysis. Hence, we performed a semi-automated approach to verify the false positive cases. Accordingly, we inspected the screenshots of pages that were detected as survey gateways. These screenshots were captured in the data collection phase during our crawling process. To verify the results, we wrote a script to programmatically open the screenshots of the pages that were detected as

Survey Gateways	(#)
Seeds	700
Guided Search (Candidate URLs)	2,301,733
URLs Classified as Survey Gateways	54,938
Unique Domains	8,623
False Positive Rate	1.2%
Detection Rate	94.8%
Survey Publishers	(#)
Unique Domains	19,123
URLs classified as Survey Publishers	318,219
Survey Completed	131,277

TABLE IV: The number of survey gateways and publishers we observed in the large-scale experiment.

survey gateways, and one of the authors manually checked the screenshots to see whether the corresponding page is in fact a survey gateway or not. The entire process to validate all the 8,728 detected survey gateways, from automatically loading each image, checking the content of the screenshot of the page to see whether it is correctly identified as a survey gateway, to closing the image took approximately 17 hours of work (7 seconds per image). We confirmed that Surveylance correctly reported 8,623 out of 8,728 detected survey gateways. Therefore, Surveylance achieved a false positive rate (FPs) of 1.2% (105 false detections out of 8,728 reported cases). Our further analysis revealed that all those cases were parked domains that included on average 17 third-party inclusions. These websites were assigned a high similarity score mainly because their HTTP network traffic was very similar to survey gateways.

Our results also show that adversaries follow very similar techniques to create online survey scams. More particularly, in order to be successful, adversaries inevitably need to frequently use inviting content or images to encourage users to take part in such fraudulent activities. Surveylance uses these limitations for survey scam defense purposes, and utilizes features (e.g. content-based, image-based features) that are specifically defined to detect these traits.

*2) **Evaluating False Negatives**:* Determining an accurate analysis on false negative cases is also a challenge since manually checking 2,301,733 URLs is not a feasible task. In the following paragraphs, we provide an *approximation* of false negatives for Surveylance.

In our experiments, false negative cases occur when a URL is, indeed, a survey gateway, but Surveylance fails to identify it as a malicious case. To reduce the manual effort of analyzing the false negative cases, we defined a semi-automated approach to pre-filter a large number of less relevant cases, and checked only the cases that were more likely to be false negatives. To this end, we created 6 clusters of survey gateways using our labeled dataset based on the similarity of their content, and ordered the words in each cluster based on their usage frequency. We selected the 10 most common words in each cluster as they were discriminative enough to correctly determine to which cluster a page belonged. As mentioned earlier, survey gateways do not usually have a large volume of text, and a large fraction of visible text in these websites is to lure users to take part in completing a survey. Given that, if a URL is a survey gateway, it should contain some degree of content similarity to the constructed clusters. We automatically computed the content similarity between a given page and the generated clusters by calculating the cosine of the angle produced by the word sets of the page and the clusters. The cosine similarity measure is a proven technique to model the frequency of words in a document using the Vector Space Model (VSM) [33]. This technique is frequently used in document indexing [38] and information retrieval [30]. In cosine similarity, if the content of two websites share exactly the same tokens, the angle will be 0, and the similarity score becomes 1. When the source HTML of two websites do not share any token, the angle becomes orthogonal, and the similarity will be 0.

We empirically observed that the cosine similarity score of 100% of the detected survey gateways (true positive cases) in the second experiment was more than 0.53. Therefore, to identify false negative cases, we had to verify all the cases that were assigned a similarity score of less than 0.53. Thus, we had to manually verify thousands of pages to measure the false negative cases. To narrow our analysis, we used the imbalanced labeled dataset to empirically *approximate* a score range which was more likely to contain false negative cases. We observed that the pages with similarity scores of less than 0.3 were very unlikely to be survey gateways. In fact, we observed that the pages with similarity scores less than 0.3 were very unlikely to contain input fields that required any information from visiting users similar to survey gateways. Therefore, we narrowed our analysis to the cases that had the similarity score between [0.3, 0.53) in our large-scale experiment. This approach decreased the size of potential survey gateways that were not detected by Surveylance to 3,013 cases. We checked the screenshots of these pages, and found 323 undetected survey gateways. Our further analysis showed that these URLs contained indicative images that we had not observed in our training phase. Therefore, the Hamming distance of the perceptual hash value of those images was between [0.19, 0.24), which was more than the experimentally-derived threshold. In Section VII, we provide more details on the limitations of Surveylance. We conclude that the system can enhance the protection capabilities against this class of social engineering attacks (with a true positive rate of 94.8% and a false positive rate of 1.2%).

*3) **Re-training the Model**:* For a real-world deployment, similar to other techniques, Surveylance requires tuning to be able to pick up on new trends, as the underlying measured scam phenomena will highly likely evolve over time. Therefore, a practical deployment of the system requires periodic re-training. To simulate a practical deployment, we started the experiment with the balanced dataset (set A), and varied the length of the testing period to determine how often we need to re-train the model to keep the detection rate constantly high, with under a 1% false positive rate. Unsurprisingly, less frequent re-training (i.e., longer testing period) resulted in less accurate detection. However, our analyses show that, based on the labeled dataset and subsequent data collection, training Surveylance with a dataset similar to set A and re-training every 12 days was sufficient to maintain the detection rate over 93%. This means that every 12 days, we verified the detection results using the procedures we explained in Sections V-C1 and V-C2 to identify the false positive and false negative cases, and re-train the detection model. The *re-*

training process, including the false positive and false negative analysis, usually took on average 4.5 hours each time over the course of experiment. Note that, since the required re-training periods may vary across different datasets, additional analysis should be performed by testing different re-training periods when SURVEYLANCE runs on a new dataset.

VI. ANALYSIS OF SURVEY SCAM SERVICES

In this section, we use the detected survey gateways in our large-scale analysis, and provide insights into the interaction between survey gateways and publishers, as well as the infrastructure used by perpetrators of survey scams. Then, we discuss techniques to inspect and identify possible abuses in these websites including deceptive advertisements, threats such as malicious code, and other fraudulent activities.

A. *Reachability of Survey Gateways*

To better understand online survey attacks, we study reachability of survey gateways in these attacks by analyzing the redirection chains extracted during our data collection phase. The reachability path of survey gateways can be viewed as a set of nodes that constitute a path starting from the first advertisement link and ending at the survey gateway. The defined chain simply illustrates the sequence of URLs followed by the victims to arrive to survey gateways. Table V shows the list of top advertisers that redirected users to survey gateways when clicked on the ads (i.e., the first domain on the redirection chain). Some of the advertisers such as `adcash.com` have been abused by adware in the past. We also observed `doubleclick.net` in 8% of the redirection chains. This is likely due to the fact that the majority of survey gateways look less aggressive or even suspicious compared to classic types of web-based social engineering attacks such as phishing websites for which blacklist operators utilize more mature techniques to detect. To extend our reachability analysis, we checked the reputation of the first node of each redirection chain. The analysis revealed that more than 40% of the survey gateways in our dataset were reachable from the Alexa top 30K. Figure 5 illustrates the distribution of the detected survey gateways among top websites.

In addition to performing an analysis of the first node in the redirection chain, we also analyzed the final node which is, in fact, the survey gateway. Table VI exhibits the most popular survey gateways we observed in our experiments. Over 40% of the survey gateways redirected users to survey publishers that encouraged victims to download legitimate software that was bundled with adware or PUPs. We provide more details on potential threats that typical users may be exposed to in Section VI-C. We performed another analysis by interacting with survey gateways to infer potential information flow to survey gateways. Please refer to Appendix B for further details.

B. *Survey Scam Domain Owners*

A question that arises is: who registers the domain names to operate survey scam services, and how are these websites managed? To answer this question, we obtained access to WHOIS records of `.com` and `.net` domains through a well-known domain registration authority. During the period that we had access, we were able to examine WHOIS records for 22,453 websites, including 2,421 survey gateways and 20,032 survey publishers. We extracted the registrant, administrator, and technical contact details from the WHOIS records and created domain clusters that contained similar registrant name, email address, and organizations using the Levenshtein distance. Among the records considered in this experiment, 1,098 domains did not contain an email address in their WHOIS records. Furthermore, 5,845 domains used anonymous WHOIS services, which prevented further analysis.

Rank	Entry Point	Percentage
1	sitescout.com	11.2%
2	onlickads.net	10.7%
3	spotxchange.com	10.2%
4	adcash.com	8.8%
5	doubleclick.net	8%
6	stickyads.com	7.8%
7	adform.com	6.3%
8	propellerads.com	6.1%
9	adify.com	4.3%

TABLE V: The list of top advertisers that redirected users to survey gateways. More than 70% of the survey gateways in our dataset were reachable from these advertisers.

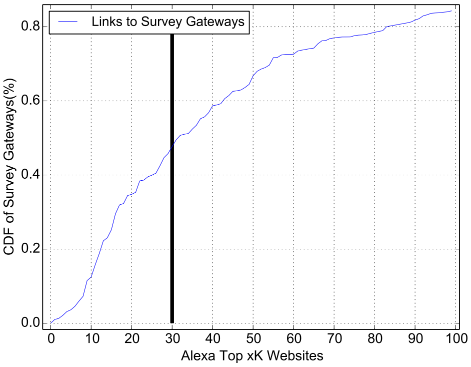


Fig. 5: The reachability of survey gateways from top websites. More than 40% of the survey gateways in our dataset were reachable from the Alexa top 30K websites.

The remaining 15,510 WHOIS records were clustered into 388 groups, where 85% of the clusters had at least 24 domains with very similar WHOIS records. We found 2,721 domains that did not have identical WHOIS records. However, these domain names satisfied 12 different regular expressions that we defined for similarity checks, suggesting that they were registered by the same identities. For example, the contact email of 881139.com was 1cangmige@qq.com, whereas the contact email of 331655.com was 406954261@qq.com. Computing the Levenshtein distance of the email addresses was not very useful in catching scenarios similar to this. However, they satisfy the same regular expression, i.e., $\wedge[0-9]\{6\}\$$ while being resolved to the same network address.

Our analysis of the clusters shows that 11% of survey gateways, which expose users to thousands of survey publishers, can be directly reached by visiting `scanlibs.com` or `ebook-dl.com`. Both of these domains are ranked in the Alexa Top 100K. On average, 32% of the visitors of these websites come from a search engine which implies that these

Rank	Survey Gateways	Main Threats
1	sweepstakescentralusa.com	PUPs, Scareware, Mal. Ext
2	idolreviieews.com	PUPs, Scareware, Mal. Ext
3	rewardzoneusa.com	PUPs, Mal. Ext
4	wesafesw.com	PUPs
5	pushcrew.com	PUPs
6	revcontent	Malware, Mal. Doc
7	blpmovies.com	Mal. Ext
8	rewardproductzone.com	Scareware, Data Exfiltration
9	linkbucks.com	PUPs, Malware
10	nonstopreward.com	PUPs
11	widgetbucks.com	Malware, Data Exfiltration
12	amarktflow.com	Data Exfiltration
13	episodetvseries.com	Data Exfiltration

TABLE VI: A list of more popular survey gateways. PUPs and malicious extensions are the main security threats introduced by survey scams.

Country	Survey Gateways	Survey Publishers
Brazil	18%	21%
Czech	8%	7%
India	9%	9%
Luxembourg	12%	15%
Panama	15%	18%
Russia	16%	20%
US	8%	3%
Rest of the World	14%	7%
Total	1,702 (100%)	13,808 (100%)

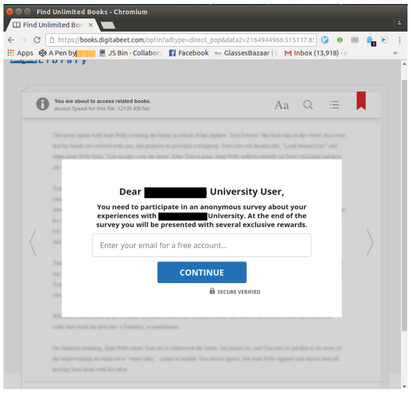
TABLE VII: Geographical distribution of underlying hosting infrastructure of survey scam services based on 15,510 valid WHOIS records.

websites are highly-connected with benign websites in the Alexa Top 100K.

We performed an experiment to test whether there is a relationship between survey gateways and publishers based on their WHOIS records. Although we observed 41 survey publishers that resolved to the IP addresses that 3 survey gateways were using, the results of our analysis did not confirm that the relationship is significant. We found 10,029 IP addresses hosting survey gateways and publishers which had low historical reputation (based on a blacklist comparison), as they were extensively used for malicious purposes (e.g., hosting malicious domains). The results support the folk wisdom that attackers have limited resources, and frequently use the same underlying infrastructure for multiple purposes.

We found that 68% of the survey publishers resolved to 11 /24 network addresses. This finding suggests that there are individuals with relatively large collections of survey scam websites, and that they use a limited set of infrastructures and addresses to carry out their attacks.

We performed another experiment to gain insights into the geographical locations of survey scam services by analyzing the distribution of countries in which these websites were hosted. Table VII shows the geographical distribution by country of survey scam services that we detected. The results clearly imply that the distribution of survey services is centered mainly around Russia, Eastern Europe, Central and South America. For instance, we found that the incidence of survey scams in some European countries – including the Czech Republic and Luxembourg – is twice as high than the United States. Overall, Panama, Brazil, and Russia were the most popular hosting locations for survey scam services, accounting for 49% of all the gateways and 59% of publishers we observed.


Find Unlimited Books - Chromium

https://books.digitabeet.com/optin?adtype=direct_pop&data2=2164944966.515117.8!

You are about to access related books.
Access Speed for this file: 12105 KB/Sec

Dear [redacted] University User,

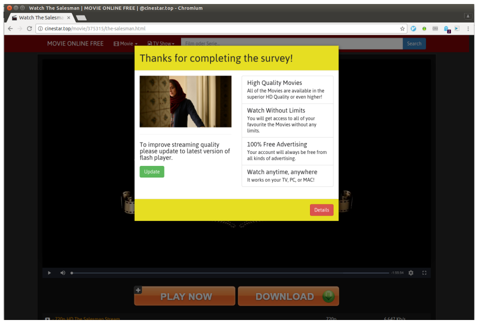
You need to participate in an anonymous survey about your experiences with [redacted] University. At the end of the survey you will be presented with several exclusive rewards.

Enter your email for a free account...

CONTINUE

SECURE VERIFIED

Fig. 6: An example of a widget that asks the user to complete a survey before accessing the content. The page sends the user to another registration page via multiple redirections after the user successfully completes the survey.


Watch The Salesman | MOVIE ONLINE FREE | @cinestar.top - Chromium

cinestar.top/movie/375315/the-salesman.html

Thanks for completing the survey!

To improve streaming quality please update to latest version of flash player.

Update

High Quality Movies
All of the Movies are available in the superior HD Quality or even higher!

Watch Without Limits
You will get access to all of your favourite the Movies without any limits.

100% Free Advertising
Your account will always be free from all kinds of advertising.

Watch anytime, anywhere
It works on your TV, PC, or MAC!

Details

Fig. 7: An example of a survey scam page that a user is exposed to. Clicking on the update button results in downloading malware.

C. *Social Engineering, Deceptive Advertisements*

As mentioned earlier, adversaries behind the survey scam ecosystem use a variety of techniques to monetize their business (e.g. injected ads, pop-ups, redirection). In the context of this paper, we performed an analysis of unavoidable overlays shown to users in these websites. We observed that overlay ads and widgets are significantly used both in survey gateways and publishers. For example, we found cases where a user was presented with overlay widgets which blocked most of the screen, and required the user to accept completing a survey to be able to proceed. Figure 6 illustrates an example that the widget super-imposed on the page, without an exit button, asking the user to complete a survey. In another case, as shown in Figure 7, the user is asked to either update the flash player, or click on the terms and conditions button, which is a fake button that redirects the user to another registration page. In 21% of the cases, the user is exposed to overlay ads which were transparently injected into a page on top of each other with a fake close button. Such deceptive practices can lure a user into clicking on a potentially malicious ad or downloading a potentially malicious binary. In the following section, we explain attack scenarios where survey scammers attempt to make permanent changes to the user's system environment in addition to extracting personal information.

During the course of our experiments, we noticed that the interaction with survey publishers usually results in opening multiple webpages that display advertisements. This is simply achieved by setting an `EventListener` on submission button clicks. Our initial crawling results showed that 63% of the survey publishers inject ads as transparent iframes. When the user clicks on a submit button, she, in fact, clicks on the overlay iframes. We updated the crawler code to identify the `<iframe>` elements and click on the overlay. The crawler also logged any redirections to other domains or downloaded binaries, and captured screenshots of the opened webpages.

In fact, manually checking the landing page and identifying the type of the page is a non-trivial task. Therefore, to determine the type of the page that a user is directed to with minimal human intervention, we leverage some image processing techniques to label them based on the visual appearance of the page. To this end, we crawled the 700 survey gateways in our labeled dataset using three different browser vendors – Chrome, Firefox, and Internet Explorer – to increase the analysis coverage, and also to decrease the effects of standard browser-based cloaking mechanisms. We collected 1,802 URLs as a result of submitting survey forms. These pages were opened in a new window or in the same window via page redirection. To analyze these URLs and automatically label them, we clustered the URLs by checking the structural similarity [49] among the visual appearance of the screenshots. We considered the structural similarity of the page since our initial analysis showed that the destination pages that a user is redirected to, after filling out the survey scams, is a finite set of malicious webpages. We exploited the high structural similarity among these webpages given the fact that this measure produces a high similarity score for two images with minor changes in color, scale, or ratio alteration. We categorized the perceived functionality of the pages into four clusters: *survey, scam, adult, or another registration page*. In 366 cases, the opened URL required the user to download a binary in order to receive the content of interest. In these cases, checking the structural similarity of the page did not reveal much about the type of binary to which a user may be exposed. We cross-checked the reputation of the downloaded binaries with VirusTotal, and defined three other categories based on the reports we received: *malicious document, malware, and PUP*.

The procedure to manually label all the 1,802 URLs took 67 hours of work. However, it saved us hundreds of hours of manual work for the more comprehensive experiment which we describe later. After the manual clustering, we measured the precision and recall by varying the value of the structural similarity threshold in order to determine the best structural similarity threshold for the 4 categories that we defined. Empirically, we assigned the threshold value $\beta = 0.82$ since at this value we were able to generate tighter clusters of websites representing different classes of threats. This experiment allowed us to determine configuration parameters for the automatic clustering performed in the large-scale experiment we describe later. *Note that this is an independent experiment, based on the output of* Surveylance, *to study the types of threats introduced by survey publishers, and is different from the experiments discussed in Section V with the goal of evaluating the detection capabilities of the classification model in* Surveylance.

Category	Chrome	Firefox	Internet Explorer	Average
Surveys	7.2%	8.4%	7.3%	7.6%
Scams	10.6%	11.5%	15.2%	12.4%
PUPs	48.2%	40.2%	38.1%	42.2%
Malware	3.2%	3.3%	6.8%	4.4%
Mal. Docs	3.3%	4.5%	2.3%	3.4%
Adult	23.3%	26.3%	25.6%	25%
Reg. Pages	4.2%	5.8%	4.7%	4.9%
Total	15,161(100%)	14,792(100%)	15,864(100%)	–

TABLE VIII: Possible cases after filling out a survey through survey publishers. A significant number of incidents result in downloading PUPs.

To conduct a more comprehensive analysis, we performed the same experiment on 318,219 survey publishers that were reachable from the 8,623 survey gateways detected by Surveylance (see Table IV). Since we were not able to automatically complete the survey in pages that required filling a CAPTCHA, we removed 3,209 survey publishers from our experiments. In this experiment, out of 318,219 survey pages, we were able to fill out 131,277 unique surveys using three different browsers (Chrome, Firefox, and Internet Explorer). Since the main goal of the experiment was to identify the types of threats after completing the surveys, we carefully analyzed what a visiting user is shown after filling out the surveys. We used the same clustering threshold ($\beta = 0.82$) to categorize the remaining URLs. Table VIII represents the result of the experiment after verifying the binaries with VirusTotal.

On average, 46.2% of the time, a click on the submit button resulted in opening pages that led an unsuspecting user to a PUP-hosting webpage. These pages attempted to trick the visiting user by claiming that she needed to install special software, or update the current version of her program to view the intended content. Figure 7 shows an example of such a webpage that was opened after Surveylance automatically completed the survey. This page asks the user to download a malicious binary which installs a backdoor on the user machine. Our analysis of 131,277 successfully filled survey suggests that some browsers have greater exposure to PUP-hosting webpages. Specifically, Chrome is the most exposed browser to PUPs though survey scam websites. One reason for this could be that, as reported in recent security studies, adversaries tend to target more popular browsers for survey scams similar to malvertising [44], [51]. We also observed scenarios where victims were shown other types of scam pages. In such cases, victims were asked to enter highly sensitive information, such as a Social Security Number or a credit card number, along with other information to receive "rewards" (Figure 10 in Appendix C). We also found 118 cases where pop-up widgets claimed that the visitor's computer was infected with malware. These websites are entry points to technical support scams which have recently been explored in prior work [22].

Based on our analysis, one can conclude that survey scams expose victims to a wide range of security threats. As shown in Table VIII, given the significance of the discovered abuses, it is evident that survey scammers tend to distribute malicious binaries, PUPs, and redirect users to other survey pages and adult content to monetize their operations in addition to stealing a user's sensitive information. As shown in Table IX, we collected 2,612 unique binaries (unique MD5s)

PUP/Malware	#
Total Binaries (unique MD5s)	2,612
Distinct Binaries	954
Unknown to VT	521

TABLE IX: Summary of suspicious binaries collected from survey publishers.

Name	Incidence	Binary Type
amonetize	350 (13.4%)	PUP
dridex	94 (3.6%)	Banking Trojan
loadmoney	255 (9.8%)	PUP
installcore	325 (12.4%)	PUP
ircbot	69 (2.7%)	Malware
kovter	23 (1%)	Ransomware
musix search	59 (2.3%)	Extension
opencandy	161 (6.2%)	PUP
somoto	875 (33.7%)	PUP
sport score	98 (3.8%)	Extension
search by zooms	162 (6.2%)	Extension
zeus	138 (5.3%)	Malware
Total	2,612 (100%)	–

TABLE X: The list of unique downloaded binaries collected after filling out surveys in survey publishers.

by visiting 22,057 URLs that delivered a binary, yielding 954 distinct polymorphic files. Of the distinct files, 521 samples were not previously submitted to AV scanners in VirusTotal. Table X shows the type of binaries obtained during the analysis of the detected survey scams. As mentioned earlier in this section, the labeling schema is not always consistent among AV scanners. For example, two scanners may generate labels of `PUP.Optional.LoadMoney` and `PUP.Gen!pac` for the same instance of the `loadmoney` PUP family. To avoid such inconsistencies, we used majority voting to label a sample. As shown, 75% of all the samples belong to PUPs, such as `somoto`, `amonetize`, `opencandy`, and `loadmoney`. While we observed other types of binaries such as malicious extensions and malware such as `zbot` (Zeus Trojan) or `kovter` (ransomware), the number of these instances was not significant compared to PUPs.

VII. DISCUSSIONS AND LIMITATIONS

In this paper, we completed 131,277 survey scams using different browser settings, performed more than 390,000 browser visits to these websites, and analyzed more than 1.2 TB of web traffic. Our analysis empirically shows that scammers, unsurprisingly, use online survey scams to distribute malware, PUPs, and other attacks. The analysis of the downloaded binaries shows that the distribution of PUPs is more popular than other types of malicious programs. This finding, in fact, is in agreement with prior work from Nelms et al. [23] where the authors analyzed a large number of social engineering attacks in the network traffic of their organization. Furthermore, our evaluation in Section V demonstrates that SURVEYLANCE achieves practical and useful detection results on a large, real-world dataset. Given the extent of the observed abuse, we envision multiple deployment scenarios for SURVEYLANCE which can potentially disrupt the operation of online survey scams in the wild. For example, SURVEYLANCE produces as output a list of survey gateways by passively crawling websites, collecting network traces, and performing the classification. The output can be used to augment blacklists or anti-malware mechanisms (e.g., Google Safe Browsing) in major browsers to reduce the exposure of users to these websites. The trained model can also be incorporated as a browser extension that monitors the content of the visited websites, and notifies the user if the website is detected as a survey scam gateway. For an attentive user, this information is likely sufficient to help determine whether the website presents a threat. Such deployment scenarios of SURVEYLANCE would potentially raise the difficulty bar for scammers, and increase the development costs of such attacks.

Although our survey scam detection technique works well in practice, it is possible that scammers may observe the advances in defenses, and adapt their attacks accordingly. In the following, we discuss the limitations of SURVEYLANCE, and the resulting implications.

First, recall that SURVEYLANCE is a supervised learning approach where attackers can change their web design strategies to evade the detection features of the detection model. Unfortunately, this is a limitation of all supervised learning techniques, including our approach where the detection model needs to be trained constantly (see Section V-C3). For example, all the false negative cases we observed in the large-scale experiment were using images that we had not observed in our training phase. Hence, adversaries can potentially exploit this fundamental limitation, and develop websites that significantly differ from the current types of survey gateways which we used in the training phase. For example, attackers may try to evade content-based features, the features with the highest relative contribution, by utilizing less textual content and more images in creating survey gateway websites. The current implementation of the system is less likely to detect these scenarios. In fact, while the nature of online survey scams is not as diverse as other web-based social engineering attacks (e.g. phishing websites), adversaries are still able to utilize a set of tricks to evade some of the specific features used in the learning process of SURVEYLANCE. However, we have not explored how these evasions may impact the practicality of the attacks, or how the corresponding websites are perceived by normal users in order to draw a concrete conclusion about their responses to such cases.

Second, note that our initial seed selection approach to obtain survey gateways relies on the Google Trends service. In our analysis, we selected a set of popular terms that real users usually incorporate in their search queries everyday. However, our seed selection was done during a limited time period, and included only a subset of major categories in the Google Trend service. Therefore, our results are based on a sampling of web pages from different categories, which is not necessarily representative of the *entire* web. In fact, we tried to explore what we believe are interesting parts of the web from a scammer's perspective. However, we cannot prove that this is a comprehensive sample of everything that a real-user would encounter on the web. Moreover, in this paper, we did not explore the possible effects of different sets of initial seeds on the detection results. At a high level, the effectiveness of our approach depends on the quantity and diversity of search items that we used to generate queries. In a real-world deployment, the seed selection strategy should be improved by considering a larger and more diverse set of items for which a user would typically search for. Furthermore, recall that our approach to collect the labeled dataset of survey gateways relies on search engine results. However, search engines try to comply with copyright violation laws, and hence might not index all potentially interesting candidates. These cases can occur in scenarios where the survey gateways use well-known logos or trademarks to convince visitors to participate in completing a survey. While our approach to collect the labeled survey gateways may be extended by incorporating other sources, such as popular social network websites that have millions of active users, we are not aware of their exact copyright violation strategy, nor have we investigated enough other sources to collect a set of survey gateway pages.

Fourth, note that providing an estimate on scammers' monetary gain is out of the scope of this paper. In this work, we mainly focused on identifying the entities and operations in survey scam ecosystem as well as analyzing the malicious practices used by adversaries to monetize those operations. However, we do not have enough data to quantify a reliable estimate of the adversaries' financial gain through survey scams. Note that this is a multi-variate problem, and a scientific approach to estimate the financial revenue requires a deeper understanding on how adversaries make revenue by delivering PUPs, collecting user information, and delivering malicious extensions.

Finally, another limitation of our study is that SURVEYLANCE cannot provide thorough information on the interaction between survey gateways and survey publishers. While adversaries behind survey scams, similar to other scammers, have significant freedom to use the collected user information, we cannot provide any accurate insights on how they may utilize the collected data. In fact, the information about the users can be collected by survey gateways, survey publishers, and third-party scripts. We have not investigated enough the interaction of these parties to make statistically significant claims about particular types of personal information misuse.

VIII. Related Work

Analogous to other scam-based attacks, social engineering techniques are a popular means to trick and recruit victims into participating in survey scams. There is a large body of work studying social engineering and deceptive attacks [1], [6], [19], [47], [16], [7], [50]. In most of this work, the focus is on the technical mechanisms used by attackers to spread malware. In our work, we also observed deceptive techniques used by attackers to lure victims into downloading malware. However, in contrast to existing work, we do not solely explore the infection phase in a malware-related attack. Rather, we study in depth how online survey scams are launched, what the attackers aim to achieve, and how victims are tricked and recruited. An additional important contribution of our work is the description and implementation of a detection approach that can identify online survey scam websites.

There is also a significant amount of prior work on the security and privacy of Internet users on the web. Similar to this existing work, the primary focus of our research is the analysis of the specific techniques scammers use in the wild to attack the security and privacy of Internet users. Therefore, the goals of our research are in-line with recent work that has investigated other types of online scams [22], scareware [17], [15], [52], PUPs [18], [45], [23], and the identification of risky websites [48], [12]. Furthermore, existing research that focuses on analyzing deceptive techniques in malicious advertisements, online fraud [20], [11], [37], [4], ad injection [51], [44], [3], and adversarial actions for monetary gains [42], [24] are also related to our work. These studies, however, target a different class of attacks. While parts of our work incorporate some of the features (e.g. extracting redirection chains) used in the context of online advertisements and malware detection, we note that our work differs significantly from existing work as survey scams have not been explored in depth before.

The most relevant work to ours in the literature is presented by Clark et al. [8], an independent work which provided the first analysis on survey scams by looking into Facebook spam URLs. The authors identified 283 survey scam URLs and manually interacted with those pages. They concluded that ad networks are active participants in this ecosystem. Compared to this work, we introduced SURVEYLANCE, an automated tool that is specifically designed to detect such malicious actions. With the use of SURVEYLANCE, we were able to studying more than 8,600 survey gateways and 318,000 survey publishers, and thereby perform a more comprehensive analysis on this ecosystem. Furthermore, we provide insights into several malicious practices that scammers use to monetize their operations.

IX. Conclusion

This paper presented SURVEYLANCE, a novel approach for detecting online survey scam websites. We implemented a prototype of SURVEYLANCE and performed a large-scale analysis of online survey scam websites. Our analysis confirms the existing anecdotal evidence that online survey scams are popular among attackers, showing that attackers aim to steal sensitive information from victims as well as deliver malware and PUPs. Our results show that SURVEYLANCE is able to successfully detect a significant number of scam websites, and potentially disrupt the malicious operations of survey scammers. We hope that the approach we present in this paper will be useful for blacklist operators, search engine providers, and endpoint security vendors in protecting their users against online survey scams.

X. Acknowledgments

This work was supported by the National Science Foundation (NSF) under grant CNS-1703454 award, and Secure Business Austria. We would like to thank our shepherd, Manos Antonakakis, and the anonymous reviewers for their helpful comments.

References

[1] ABRAHAM, S., AND CHENGALUR-SMITH, I. An overview of social engineering malware: Trends, tactics, and implications. *Technology in Society 32*, 3 (2010), 183–196.

[2] ALEXA INTERNET INC. Free stuff sub category. http://www.alexa.com /topsites/category/Computers/Internet/On_the_Web/Free_Stuff, 2017.

[3] ARSHAD, S., KHARRAZ, A., AND ROBERTSON, W. Identifying extension-based ad injection via fine-grained web content provenance. In *International Symposium on Research in Attacks, Intrusions, and Defenses* (2016), Springer, pp. 415–436.

[4] ARSHAD, S., KHARRAZ, A., AND ROBERTSON, W. Include me out: In-browser detection of malicious third-party content inclusions. In *Proceedings of the 20th International Conference on Financial Cryptography and Data Security (FC)* (2 2016).

[5] Breiman, L. Random forests. *Machine learning 45*, 1 (2001), 5–32.

[6] Caballero, J., Grier, C., Kreibich, C., and Paxson, V. Measuring pay-per-install: The commoditization of malware distribution. In *Usenix security symposium* (2011), p. 15.

[7] Cialdini, R. B. *Influence: Science and Practice*. Writers of the Round Table Press, Boston, 2009.

[8] Clark, J. W., and McCoy, D. There are no free ipads: An analysis of survey scams as a business. In *6th USENIX Workshop on Large-Scale Exploits and Emergent Threats, LEET '13, Washington, D.C., USA, August 12, 2013* (2013).

[9] Facebook Bussiness. How to target Facebook Ads. https://www.facebook.com/business/a/online-sales/ad-targeting-details, 2017.

[10] Google Development. Extending DevTools. https://developer.chrome.com/extensions/devtools, 2017.

[11] Haddadi, H. Fighting online click-fraud using bluff ads. *ACM SIGCOMM Computer Communication Review 40*, 2 (2010), 21–25.

[12] Invernizzi, L., and Comparetti, P. M. Evilseed: A guided approach to finding malicious web pages. In *Security and Privacy (SP), 2012 IEEE Symposium on* (2012), IEEE, pp. 428–442.

[13] Jashua, B. How qualtrics turned online surveys into a 1 billion bussiness. https://www.bloomberg.com/news/articles/2014-09-24/how-qualtrics-turned-online-surveys-into-a-1-billion-business, 2014.

[14] Jocelyn, B. Customer Service Survey Scams: Dont Fall for Them! http://www.huffingtonpost.com/nextadvisorcom/customer-service-survey-s_b_7844408.html, 2015.

[15] Kharraz, A., Arshad, S., Mulliner, C., Robertson, W., and Kirda, E. Unveil: A large-scale, automated approach to detecting ransomware. In *25th USENIX Security Symposium (USENIX Security 16)* (2016), USENIX Association, pp. 757–772.

[16] Kharraz, A., Kirda, E., Robertson, W., Balzarotti, D., and Francillon, A. Optical Delusions: A Study of Malicious QR Codes in the Wild. In *Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)* (06 2014).

[17] Kharraz, A., Robertson, W., Balzarotti, D., Bilge, L., and Kirda, E. Cutting the gordian knot: A look under the hood of ransomware attacks. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment* (2015), Springer, pp. 3–24.

[18] Kotzias, P., Bilge, L., and Caballero, J. Measuring pup prevalence and pup distribution through pay-per-install services. In *25th USENIX Security Symposium (USENIX Security 16)* (Austin, TX, 2016), USENIX Association, pp. 739–756.

[19] Kwon, B. J., Mondal, J., Jang, J., Bilge, L., and Dumitras, T. The dropper effect: Insights into malware distribution with downloader graph analytics. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (2015), ACM.

[20] Li, Z., Zhang, K., Xie, Y., Yu, F., and Wang, X. Knowing your enemy: understanding and detecting malicious web advertising. In *Proceedings of the 2012 ACM conference on Computer and communications security* (2012), ACM, pp. 674–686.

[21] Microsoft Corporation. Cognitive Services Pricing – Bing Search API. https://azure.microsoft.com/en-us/pricing/details/cognitive-services/search-api/web/, 2017.

[22] Miramirkhani, N., Starov, O., and Nikiforakis, N. Dial one for scam: Analyzing and detecting technical support scams. In *22nd Annual Network and Distributed System Security Symposium (NDSS 16* (2016), NDSS.

[23] Nelms, T., Perdisci, R., Antonakakis, M., and Ahamad, M. Towards measuring and mitigating social engineering software download attacks. In *25th USENIX Security Symposium (USENIX Security 16)* (Austin, TX, 2016), USENIX Association, pp. 773–789.

[24] Nikiforakis, N., Maggi, F., Stringhini, G., Rafique, M. Z., Joosen, W., Kruegel, C., Piessens, F., Vigna, G., and Zanero, S. Stranger danger: exploring the ecosystem of ad-based url shortening services. In *Proceedings of the 23rd international conference on World wide web* (2014), ACM, pp. 51–62.

[25] Nishant, D. Survey Scammers Moving to Pinterests. https://www.symantec.com/connect/blogs/survey-scammers-moving-pinterest, 2014.

[26] NLTK 3.2.3 documentation. Natural Language Toolkit. http://www.nltk.org/, 2017.

[27] Opinion Miles Club. Earn award miles for sharing your opinions. https://www.opinionmilesclub.com/, 2017.

[28] Oscar, A. Survey Scams Aimed at Social Networking Netizens. https://www.trendmicro.com/vinfo/us/threat-encyclopedia/web-attack/109/survey-scams-aimed-at-social-networking-netizens, 2012.

[29] Pasgrimaud, G. Pyquery: a jquery-like library for python, 2017. https://pythonhosted.org/pyquery/.

[30] Polettini, N. The vector space model in information retrieval- term weighting problem, 2004.

[31] Provos, N., Mavrommatis, P., Rajab, M. A., and Monrose, F. All your iframes point to us. In *Proceedings of the 17th Conference on Security Symposium* (Berkeley, CA, USA, 2008), SS'08, USENIX Association, pp. 1–15.

[32] Salton, G., and McGill, M. J. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.

[33] Salton, G., Wong, A., and Yang, C. S. A vector space model for automatic indexing. In *ACM* (New York, NY, USA, November 1975), ACM, pp. 613–620.

[34] Satnam, N. Instascam: Instagram for PC Leads to Survey Scam. https://www.symantec.com/connect/blogs/instascam-instagram-pc-leads-survey-scam, 2013.

[35] SciKit Learn. Random Forest Algorithm. http://scikit-learn.org/stable/modules/ensemble.htmlrandom-forests, 2017.

[36] Secrects, M. M. Earn free united miles if you have a lot of spare time, that is!, 2016. http://millionmilesecrets.com/2014/02/27/earn-free-united-miles-if-you-have-a-lot-of-spare-time-that-is/.

[37] Springborn, K., and Barford, P. Impression fraud in on-line advertising via pay-per-view networks. In *USENIX Security* (2013), pp. 211–226.

[38] Stanford University. The vector space model for scoring. https://nlp.stanford.edu/IR-book/html/htmledition/the-vector-space-model-for-scoring-1.html, 2009.

[39] Starov, O., Gill, P., and Nikiforakis, N. Are you sure you want to contact us? quantifying the leakage of pii via website contact forms. *Proceedings on Privacy Enhancing Technologies 2016*, 1 (2016), 20–33.

[40] Stelian, P. Remove 2017 Annual Visitor Survey pop-up ads (Virus Removal Guide). https://malwaretips.com/blogs/remove-2017-annual-visitor-survey-popups/, 2017.

[41] Stelian, P. Remove Chrome Opinion Survey pop-ups (Virus Removal Guide). https://malwaretips.com/blogs/remove-chrome-opinion-survey-popup/, 2017.

[42] Stone-Gross, B., Abman, R., Kemmerer, R. A., Kruegel, C., Steigerwald, D. G., and Vigna, G. The underground economy of fake antivirus software. In *Economics of Information Security and Privacy III*. Springer, 2013, pp. 55–78.

[43] The World Wide Web Consortium (W3C). Http archive (har) format, 2012. https://dvcs.w3.org/hg/webperf/raw-file/tip/specs/HAR/Overview.html.

[44] Thomas, K., Bursztein, E., Grier, C., Ho, G., Jagpal, N., Kapravelos, A., McCoy, D., Nappa, A., Paxson, V., Pearce, P., et al. Ad injection at scale: Assessing deceptive advertisement modifications. In *Security and Privacy (SP), 2015 IEEE Symposium on* (2015), IEEE, pp. 151–167.

[45] Thomas, K., Crespo, J. A. E., Rasti, R., Picod, J.-M., Phillips, C., Decoste, M.-A., Sharp, C., Tirelo, F., Tofigh, A., Courteau, M.-A., Ballard, L., Shield, R., Jagpal, N., Rajab, M. A., Mavrommatis, P., Provos, N., Bursztein, E., and McCoy, D. Investigating commercial pay-per-install and the distribution of unwanted software. In *25th USENIX Security Symposium (USENIX Security 16)* (Austin, TX, 2016), USENIX Association, pp. 721–739.

[46] Thomas, K., Crespo, J. A. E., Rasti, R., Picod, J. M., Phillips, C., Decoste, M.-A., Sharp, C., Tirelo, F., Tofigh, A., Courteau, M.-A., et al. Investigating commercial pay-per-install and the distribution of unwanted software. In *USENIX Security Symposium* (2016), pp. 721–739.

[47] Vadrevu, P., Rahbarinia, B., Perdisci, R., Li, K., and Antonakakis, M. Measuring and detecting malware downloads in live

network traffic. In *European Symposium on Research in Computer Security* (2013), Springer, pp. 556–573.

[48] Vissers, T., Joosen, W., and Nikiforakis, N. Parking sensors: Analyzing and detecting parked domains. In *Annual Network and Distributed System Security Symposium* (2015), The Internet Society.

[49] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on 13*, 4 (2004), 600–612.

[50] Whaley, B. Toward a general theory of deception. *The Journal of Strategic Studies 5*, 1 (1982), 178–192.

[51] Xing, X., Meng, W., Lee, B., Weinsberg, U., Sheth, A., Perdisci, R., and Lee, W. Understanding malvertising through ad-injecting browser extensions. In *Proceedings of the 24th International Conference on World Wide Web* (2015), WWW '15.

[52] Zarras, A., Kapravelos, A., Stringhini, G., Holz, T., Kruegel, C., and Vigna, G. The dark alleys of madison avenue: Understanding malicious advertisements. In *Proceedings of the 2014 Conference on Internet Measurement Conference* (2014), ACM, pp. 373–380.

[53] Zauner, C. Implementation and benchmarking of perceptual image hash functions. http://www.phash.org/, 2010.

Appendix

A. Third-party Inclusions in Survey Gateways

To motivate our use of third-party script incidence as a feature, we compared the usage of third-party scripts (e.g., advertisements) in survey gateways to benign survey pages. Figure 8 shows the number of unique third-party scripts used as advertisement on survey gateways versus the number of unique third-party scripts referenced by the baseline benign survey pages. The plot clearly shows that survey gateways include significantly more third-party scripts to benign survey pages.

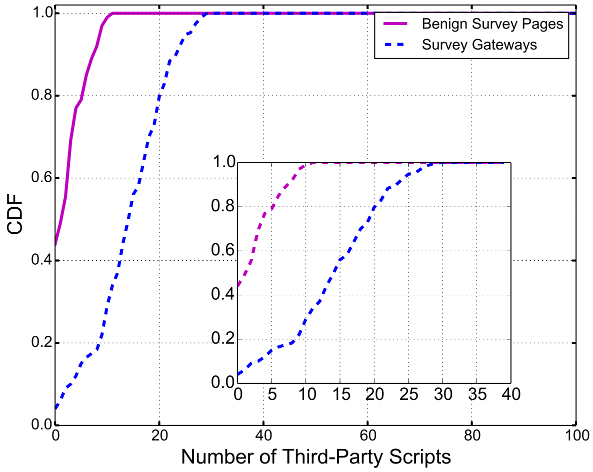


Fig. 8: Number of included third-party scripts in the survey scam pages.

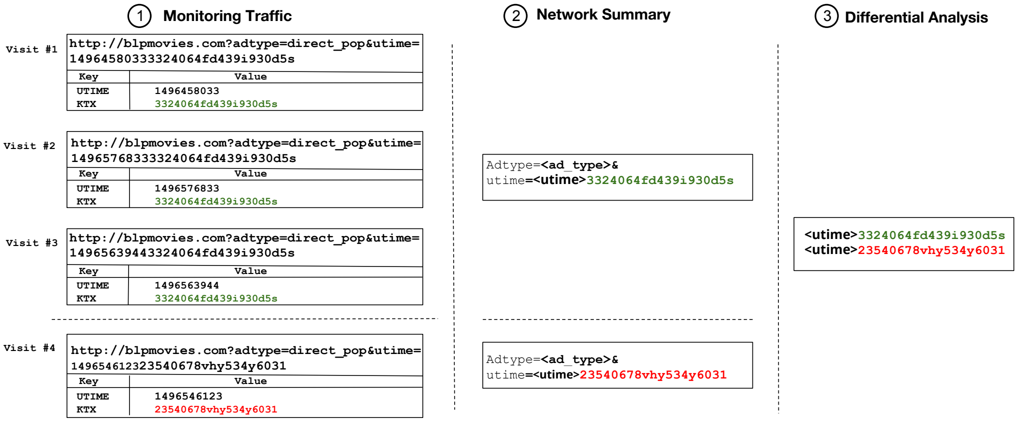
B. Interacting with Survey Gateways

Our preliminary experiment on 10 survey gateways showed that the transmitted data to servers varies if different browser configurations and IP addresses are used, albeit providing identical responses to the initial set of questions (e.g. age, gender). To better explore this, we conducted a larger scale study on 200 randomly selected survey gateways to infer potential information flow into survey gateways.

Our analysis consists of two phases. In the first phase, we visited each survey gateway multiple times with an identical browser profile (i.e., same IP address and browser user-agent) and collected raw network traces. The goal of this phase is to construct a complete picture of the network behavior of a given survey gateway. Note that determining the number of times to visit a gateway website in order to draw a comprehensive view of its network behavior strongly depends on the complexity of network traces between the survey gateway and the browser. For example, the number and sources of non-deterministic parameters usually vary from website to website and can have significant impacts on our analysis in this experiment. By performing a differential analysis on the collected traces, we empirically observed that running the first phase of the experiment three times is sufficient to reach convergence and identify potential discrepancies in the traces for a large number of websites in our dataset.

To this end, we collected the raw HTTP traffic sent to survey gateways and also monitored their interactions with browser features such as WebStorage APIs (i.e., LocalStorage, SessionStorage). We then checked the raw HTTP traffic and searched for values by string comparison to find any potential sources of non-determinism. We used similar techniques that prior work [39] employed to extract specific values in the network traffic. In fact, we labeled any parameters that varied during the first phase in which we did not alter any source of information. We then combined all the traces and defined the *behavior summary* of the given survey gateway, which was then used in the second phase of the experiment. In the second phase, after visiting the survey gateway with a different browser user-agent, we monitored the information sent to the server to identify potential sources of non-determinism. Figure 9 illustrates a simplified version of our experiment on one of the survey gateways.

From the 200 randomly selected survey gateways, we observed that 144 (72%) of them were interacting with browser WebStorage APIs by calling `set` and `get` functions. However, 112 (56%) of all the websites did not reach convergence during the first phase of our analysis, or we were not able to extract any particular patterns due to multiple levels of encoding. Our empirical analysis on the network traces shows that among all the remaining 88 (44%) cases, the most common sources of non-determinism were timestamps, dates, cookies or session identifiers assigned by gateways. This experiment clearly implies that the interaction with survey gateways relies on creating and maintaining unique IDs for visiting users. In fact, survey gateways differentiate among visiting users and redirect them to survey publishers based on their browser configuration and responses to the initial set of questions. However, we cannot claim that the interaction between survey gateways and publishers relies solely on the set of unique IDs that we observed by interacting with survey gateways. Furthermore, we do not know whether adversaries use any specific technique besides the user-agent analysis, nor have we investigated enough to make statistically significant claims about any particular types of fingerprinting techniques.



Visit #1

```
http://blpmovies.com?adtype=direct_pop&utime=
14964580333324064fd439i930d5s
```

Key	Value
UTIME	1496458033
KTX	3324064fd439i930d5s

Visit #2

```
http://blpmovies.com?adtype=direct_pop&utime=
14965768333324064fd439i930d5s
```

Key	Value
UTIME	1496576833
KTX	3324064fd439i930d5s

Visit #3

```
http://blpmovies.com?adtype=direct_pop&utime=
14965639443324064fd439i930d5s
```

Key	Value
UTIME	1496563944
KTX	3324064fd439i930d5s

Visit #4

```
http://blpmovies.com?adtype=direct_pop&utime=
149654612323540678vhy534y6031
```

Key	Value
UTIME	1496546123
KTX	23540678vhy534y6031

```
Adtype=<ad_type>&
utime=<utime>3324064fd439i930d5s
```

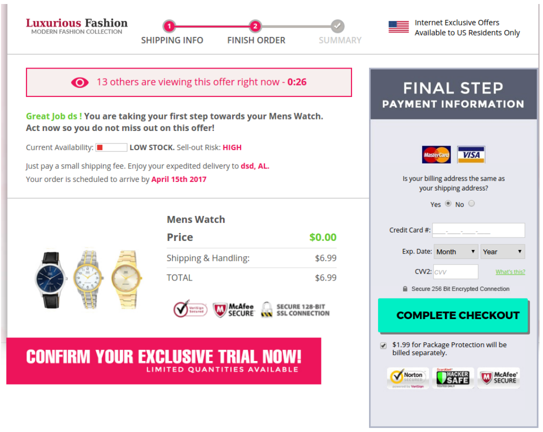
```
Adtype=<ad_type>&
utime=<utime>23540678vhy534y6031
```

```
<utime>3324064fd439i930d5s
<utime>23540678vhy534y6031
```

Fig. 9: An example of how we analyzed the browser interaction with survey gateways. In the first phase, we create a behavior summary of a given survey gateway by visiting the website n times, and locating non-deterministic parameters. In the second phase, we perform a differential analysis by changing the browser setting and IP address to identify differences in HTTP traffic.

C. Exposing Victims to Scam Pages After Filling Out a Survey Page

After completing a survey, victims are redirected to a new page, and are usually asked to enter sensitive information, such as a credit card number, along with other information to receive rewards. Figure 10 shows an example of a scam page after completing a survey.



Internet Exclusive Offers
Available to US Residents Only

13 others are viewing this offer right now - **0:26**

Great Job ds ! You are taking your first step towards your Mens Watch. Act now so you do not miss out on this offer!

Current Availability: **LOW STOCK.** Sell-out Risk: **HIGH**

Just pay a small shipping fee. Enjoy your expedited delivery to **dsd, AL.**
Your order is scheduled to arrive by **April 15th 2017**

Mens Watch

Price	**$0.00**
Shipping & Handling:	$6.99
TOTAL	$6.99

CONFIRM YOUR EXCLUSIVE TRIAL NOW!
LIMITED QUANTITIES AVAILABLE

FINAL STEP
PAYMENT INFORMATION

Is your billing address the same as your shipping address?

Yes ● No ○

Credit Card #: ____-____-____-____

Exp. Date: Month ▾ Year ▾

CVV2: CVV What's this?

Secure 256 Bit Encrypted Connection

☑ $1.99 for Package Protection will be billed separately.

Fig. 10: An example of a scam page that a user is exposed to after filling out a survey.