Exposing Search and Advertisement Abuse Tactics and Infrastructure of Technical Support Scammers

Bharat Srinivasan
Salesforce.com

Athanasios Kountouras
Georgia Institute of Technology

Najmeh Miramirkhani
Stony Brook University

Monjur Alam
Georgia Institute of Technology

Nick Nikiforakis
Stony Brook University

Manos Antonakakis
Georgia Institute of Technology

Mustaque Ahamad
Georgia Institute of Technology

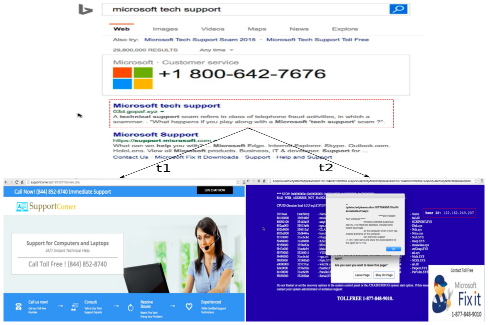
ABSTRACT

Technical Support Scams (TSS), which combine online abuse with social engineering over the phone channel, have persisted despite several law enforcement actions. Although recent research has provided important insights into TSS, these scams have now evolved to exploit ubiquitously used online services such as search and sponsored advertisements served in response to search queries. We use a data-driven approach to understand search-and-ad abuse by TSS to gain visibility into the online infrastructure that facilitates it. By carefully formulating tech support queries with multiple search engines, we collect data about both the support infrastructure and the websites to which TSS victims are directed when they search online for tech support resources. We augment this with a DNS-based amplification technique to further enhance visibility into this abuse infrastructure. By analyzing the collected data, we provide new insights into search-and-ad abuse by TSS and reinforce some of the findings of earlier research. Further, we demonstrate that tech support scammers are (1) successful in getting major as well as custom search engines to return links to websites controlled by them, and (2) they are able to get ad networks to serve malicious advertisements that lead to scam pages. Our study period of approximately eight months uncovered over 9,000 TSS domains, of both passive and aggressive types, with minimal overlap between sets that are reached via organic search results and sponsored ads. Also, we found over 2,400 support domains which aid the TSS domains in manipulating organic search results. Moreover, to our surprise, we found very little overlap with domains that are reached via abuse of domain parking and URL-shortening services which was investigated previously. Thus, investigation of search-and-ad abuse provides new insights into TSS tactics and helps detect previously unknown abuse infrastructure that facilitates these scams.

ACM Reference Format:
Bharat Srinivasan, Athanasios Kountouras, Najmeh Miramirkhani, Monjur Alam, Nick Nikiforakis, Manos Antonakakis, and Mustaque Ahamad. 2018. Exposing Search and Advertisement Abuse Tactics and Infrastructure of Technical Support Scammers. In *WWW 2018: The 2018 Web Conference, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3178876.3186098

WWW 2018, April 23–27, 2018, Lyon, France

ACM ISBN 978-1-4503-5639-8/18/04.
https://doi.org/10.1145/3178876.3186098



microsoft tech support

Also try: Microsoft Tech Support Scam 2015 · Microsoft Tech Support Toll Free

29,800,000 RESULTS Any time

Microsoft · Customer service
+1 800-642-7676

Microsoft tech support
03d.gopaf.xyz
A technical support scam refers to class of telephone fraud activities, in which a scammer. . "What happens if you play along with a Microsoft 'tech support' scam ?".

Microsoft Support
https://support.microsoft.com
What can we help you with? ... Microsoft Edge. Internet Explorer. Skype. Outlook.com. HoloLens. View all Microsoft products. Business, IT & developer. Support for ...
Contact Us · Microsoft Fix It Downloads · Support · Help and Support

Call Now! (844) 852-8740 Immediate Support

SupportCorner

Support for Computers and Laptops
24/7 Instant Technical Help
Call Toll Free ! (844) 852-8740

Call us now! Call our Toll free Number → Consult Talk to Our Tech Support Experts → Resolve Issues Watch The Tech Fixing Your Problem → Experienced 100% Certified Support Technicians.

Your Computer **** Windows Detected Suspicious Activity. The Infections detected, indicate some recent downloads on the computer which in turn has created problems on the computer. Call technical support +1-877-848-9010 and share this code B2957E to the Agent to Fix This.

Are you sure you want to leave this page?

Do not Restart or set the recovery options in the system control panel or the /CRASHDEBUG system start option. If this mes
contact your system administrator or technical support

TOLLFREE 1-877-848-9010.

Contact Toll Free
Microsoft Fix it
1-877-848-9010

Figure 1: The first search result on Bing.com on 02/02/2017 for 'microsoft tech support' points to domain 03d.gopaf.xyz which redirects to different types of TSS websites – passive (left) and aggressive (right) – depending on user context.

1 INTRODUCTION

The *Technical Support Scam* (TSS), in which scammers dupe their victims into sending hundreds of dollars for fake technical support services, is now almost a decade old. It started with scammers making cold calls to victims claiming to be a legitimate technology vendor but has now evolved into the use of sophisticated online abuse tactics to get customers to call phone numbers that are under the control of the scammers. In their research on TSS [53], Miramirkhani et. al. explored both the web infrastructure used by tech support scammers and the tactics used by them when a victim called a phone number advertised on a TSS website. They focused on TSS websites reached via malicious advertisements that are served by abusing domain parking and ad-based URL shortening services. Although their work provided important insights into how these services are abused by TSS, it has recently become clear that tech support scammers are diversifying their methods of reaching victims and convincing these victims to call them on their advertised phone numbers. Recent reports by the US Federal Trade Commission (FTC) and by search engines vendors suggest that scammers are turning to search engine results and the ads shown on search-results pages to reach their victim [5, 11, 30]. These new channels not only allow them to reach a wider audience but also help them diversify the ways in which they convince users to call them. Both government regulators and companies have taken action to stop TSS but these scams continue to adapt and evade their efforts [5, 7, 8, 12, 13, 30, 33–35].

In this paper, we perform the first systematic study of TSS abuse of search-and-ad channels. We develop a model for generating tech-support related queries and use the resulting 2,600 queries as daily searches in popular and less popular search engines. By crawling the organic search results and ads shown in response to our queries (note that we follow a methodology that allows us to visit the websites of ads while avoiding click-fraud), we discover thousands of domains and phone numbers associated with TSS. In addition to the traditional *aggressive* variety of TSS where visited webpages attempt to scare users into calling them, we observe a large number of *passive* TSS pages which appear to be professional, yet nevertheless are operated by technical support scammers. Figure 1 shows an example of such a scam. Using network-amplification techniques, we show how we can discover many more scam pages present on the same network infrastructure, and witness the co-location of aggressive with passive scam pages. This indicates that a fraction of these aggressive/passive scams are, in fact, controlled and operated by the same scammers. Our main contributions are the following:

- We design the first search-engine-based system for discovering TSS, and utilize it for eight months to uncover more than 9,000 TSS-related domains and 3,365 phone numbers operated by technical support scammers, present in both organic search results as well as ads located on search-results pages. We analyze the resulting data and provide details of the abused infrastructure, the SEO techniques that allow scammers to rank well on search engines, and the long-lived *support* domains which allow TSS domains to remain hidden from search engines.
- We find that scammers are complementing their aggressive TSS pages with passive ones, which both cater to different audiences and, due to their non-apparent malice, have a significantly longer lifetime. We show that well-known network amplification techniques allow us to not only discover more TSS domains but to also trace both aggressive and passive TSS back to the same actors.
- We compare our results with the ones from the recent TSS study of Miramirkhani et al. [53] and show that the vast majority of our discovered abusive infrastructure is not detected by prior work, allowing defenders to effectively *double* their coverage of TSS abuse infrastructure by incorporating our techniques into their existing TSS-discovering systems. Thus, our system reveals part of the TSS ecosystem that remained, up until now, unexplored.

2 METHODOLOGY

We utilize a data-driven methodology to explore TSS tactics and infrastructure used to support search-and-ad abuse. To do this, we search and crawl the web to collect a variety of data about TSS websites. Our system, which is shown in Figure 2, implements TSS data collection and analysis functions, and consists of the following six modules:

(1) The *Seed Generator* module generates phrases that are likely to be used in search queries to find tech support resources. It uses a known corpus of TSS webpages obtained from Malwarebytes [24] and a probabilistic language modeling technique to generate such search phrases.

n	# ngrams	Example English Phrase
1	74	virus
2	403	router support
3	1,082	microsoft tech support
4	720	microsoft online support chat
5	243	technical support for windows vista
6	72	hp printers technical support phone number
7	6	contact norton antivirus customer service phone number
Total	2,600 english phrases	

Table 1: Summary and examples of generated n-grams related to technical support scams.

(2) Using search phrases, the *Search Engine Crawler (SEC)* module mines popular search engines such as Google, Bing and Yahoo! for technical support related content appearing via search results (SRs) and sponsored advertisements (ADs). We also mine a few obscure ones such as goentry.com and search.1and1.com that we discovered are used by tech support scammers.

(3) The *Active Crawler Module (ACM)* then tracks and records the URI redirection events, HTML content, and DNS information associated with the URIs/domains appearing in the ADs and SRs crawled by the SEC module.

(4) *Categorization module* which includes a well-trained TSS website classifier, is used to identify TSS SRs and ADs using the retrieved content.

(5) The *Network Amplification Module (NAM)* uses DNS data to amplify signals obtained from the labeled TSS domains, such as the host IP, to expand the set of domains serving TSS, using an amplification algorithm.

(6) Lastly, using the information gathered about TSS domains, the *Clustering Module* groups together domains sharing similar attributes at the network and application level.

2.1 Search Phrase Seed Generator

We must generate search phrases that are highly likely to be associated with content shown or advertised in TSS webpages to feed to the search engine crawler module. Deriving relevant search queries from a context specific corpus has been used effectively in the past for measuring search-redirection attacks [50]. We use an approach based on joint probability of words in phrases in a given text corpus [52]. We start with a corpus of 500 known TSS websites from the Malwarebytes TSS domain blacklist (DBL) [24], whose webpage content was available. We were able to find 869 unigrams or single words after sanitizing the content in the corpus for stop words. We then rank these unigrams based on the TF-IDF weighting factor and pick the most important unigrams as an initial step. This leaves us with seventy four unique words. Using the raw counts of unigrams, we compute the raw bi-gram probabilities of eligible phrases with the chain rule of probability. We then use the Markov assumption to approximate n-gram probabilities [25]. Table 1 shows the total number of phrases found for different values of n and some examples of the phrases found. We restricted the value of n to 7, as the value of $n = 8$ does not yield any significant phrases. This way, we were able to identify 2600 English phrases that serve as search queries to the SEC module.

2.2 Search Engine Crawler (SEC) Module

The SEC module uses a variety of search engines and the search phrases generated from the TSS corpus to capture two types of listing: traditional search results, sometimes also referred to as organic

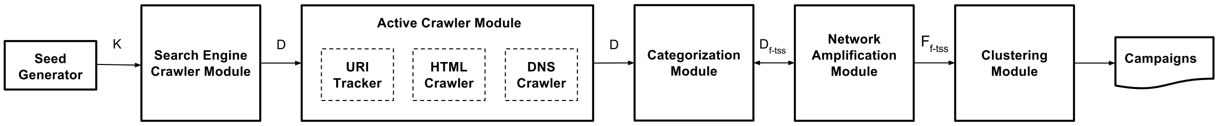


Figure 2: X-TSS threat collection and analysis system.

search results, and search advertisements, sometimes also referred to as paid/sponsored advertisements. Both Google [15] and Bing [9] provide APIs that can be used to get SRs. However, some of the search engines we considered did not have well documented APIs and vanilla crawlers are either blocked or not shown content such as ADs. In such cases, we automate the process using PhantomJS [26], a headless WebKit "scriptable" with a JavaScript API. It allows us to capture both SR and AD listings as it would be shown to a real user visiting the search engine from an actual browser.

Once we have the raw page p from the search engine in response to a query q, we use straighforward CSS selectors to separate the SRs from ADs. A SR object typically consists of basic components such as the the SR title, the SR URI, and a short snippet of the SR content. An AD object too, typically consists of these components, i.e. the AD title, the advertiser's URI/domain name, and a short descriptive text. The advertiser also provides the URI the user should be directed to when the AD is clicked. The SR/AD along with its components are logged into a database as a JSON object. The URI component of the ADs and SRs are then inserted into the ADC (AD crawling) and SRC (SR crawling) queues respectively, which then coordinate with the ACM to gather more information about them.

2.3 Active Crawler Module (ACM)

The ACM uses the ADC and SRC URI queues to gather more information relevant to an AD/SR. ACM has three submodules that keep track of the following information for each URI seen in the AD/SR: (i) URI tracking, (ii) HTML and Screenshot Capture, and (iii) DNS information.

URI Tracker: The purpose of the URI tracker is to follow and log the redirection events starting from the URI component seen in the AD/SR discussed in the previous module. Barring user clicks, our goal is to capture the sequence of events that a user on a real browser would experience when directed to technical support scams from SR/AD results, and *automate* this process. Our system uses a combination of python modules PhantomJS [26], Selenium [27] and BeautifulSoup [6] to script a light-weight headless browser. Finally, to ensure wide coverage, we configure our crawlers with different combinations of Referer headers and User-Agents.

Mimicking AD Clicks: When a user clicks on an AD, the click triggers a sequence of events in which the publisher, AD network and advertiser are involved, before the user lands on the intended webpage associated with the AD. Clearly, the intent of our automated crawlers is not to interfere with the AD monetization model by introducing extraneous clicks. One alternative to actually clicking on the ADs and a way to bypass the AD network is to visit the advertiser's domain name directly, while maintaining the *Referer* to be the search engine displaying the AD. In theory, any further redirections from the advertiser's domain should still be captured.

To validate if this was a viable option while maintaining accuracy of the data collection process, we conducted a controlled experiment in which we compared a small number of recorded URI resolution paths generated by real clicks to paths recorded while visiting the advertiser's domain name directly. We did this for the same set of technical support ADs while keeping the same browser and IP settings. For a set of 50 fake technical support ADs from different search engines identified manually and at random, these paths were found to be identical giving us confidence in this approach.

HTML Crawler: The HTML crawler works in conjunction with the URI Tracker and captures both the raw HTML as well as visual screenshots of webpages shown after following the ADs and SRs. For each domain d and webpage p, in the path from an AD/SR to the final-landing webpage, the crawler stores the full source HTML and an image of the webpage as it would have appeared in a browser, into a database.

Active DNS Crawler: For each domain, d, in the path from an AD/SR to the final-landing domain, the active DNS crawler logs the IP address, ip, associated with the domain to form a (d, ip, t) triplet, based on the DNS resolution process at the time of crawling, t. This information is valuable for unearthing new technical support scam domains (Section 2.5) and in studying the network infrastructure associated with TSS (Section 4).

2.4 Categorization Module

Although we input technical support phrases to search engines with the aim of finding fake technical support websites, it is possible and even likely that some SRs and ADs lead to websites that are legitimate technical support or even completely unrelated to technical support. To categorize all search engine listings obtained during the period of data collection, we first divide the URIs collected from both ADs and SRs into two high-level categories: TSS and Non-TSS, (i.e. those URIs that lead to technical support scam pages and those that lead to benign or unrelated pages). Within each category, we have subcategories: TSS URIs are further separated into those leading to aggressive TSS websites and those leading to passive TSS websites.

TSS Website Classifier: We determine an AD/SR as technical support scam or not primarily based on the webpage content shown in the final-landing domain corresponding to an AD/SR. We leverage the observation that a lot of fake technical support websites host highly similar content, language and words to present themselves [53]. This can be represented as a feature vector where features are the words and values are the frequency counts of those words. Thus, for a collection of labeled TSS and Non-TSS websites, we extract the bag of words after sanitization (such as removing stop words), and create a matrix of feature vectors where the rows are the final-landing domains and the columns are the text features. We can then train a classifier on these features which can be used to automatically label future websites.

To that effect, we built a model using the Naive Bayes classification algorithm with 10-fold cross validation on a set comprising of 500 technical support scam and 500 non-technical support scam websites identified from the first few weeks of ADs/SRs data. The

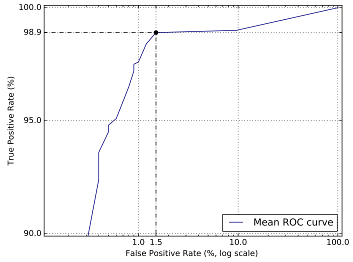


Figure 3: ROC Curve of the TSS Website Classifier on the training set.

training set is randomly selected and manually labeled. The selection consists of representative samples of different kinds of TSS webpages, both passive and aggressive types, along with Non-TSS webpages that were found among the search listings including benign or unrelated webpages. The performance of the classifier is captured in the ROC Curve shown in Figure 3. We see that a threshold of 0.6 yields to an acceptable true positive rate (TPR) of 98.9% and a false positive rate (FPR) of 1.5%. Moreover the area under the curve (AUC), which is a measure of the overall accuracy of the trained model, is 99.33% which gives us confidence that the technical vs. non-technical support webpage classification works well. To make sure, we are not including, genuine, popular and high reputation technical support service websites in our TSS dataset, eg. Best Buy's Geek Squad [14], we drop domain names (if any), appearing in the Alexa top 10,000 websites list [4].

Next, to separate TSS URIs into those leading to passive/aggressive websites, we use the presence of features extracted from the HTML of the landing TSS website. Aggressive TSS websites exhibit behavior that contributes to a false sense of urgency and panic through a combination of audio messages describing the problem and continuous pop-up messages/dialogue loops which can be detected using tags such as <audio>, window.alert(), window.confirm(), window.prompt() etc. On the other hand, passive TSS websites adopt the approach of seeming genuine. This is accomplished by using simple textual content, certifications, seals, and other brand-based images. They often present themselves as official tech support representatives of large companies and, because of their non-apparent malice, pose new challenges for the detection of TSS [5].

To evaluate the performance of this TSS classifier, we sample data from the test AD/SR dataset. To verify actual TSS websites, we use Malwarebytes [24] TSS blacklist data as an independent source of ground truth. The blacklist consists of domain names and phone numbers that serve both passive and aggressive TSS. However, certain websites from the test set that are marked as TSS may not be listed in Malwarebytes. For these, we use a combination of manual analysis of the website content, IP co-location indicators, WHOIS giveaways and relevant online complaints associated with the advertised phone number to verify that the website is indeed associated with TSS. While aggressive TSS websites are easy to verify using characteristics of the website content itself, passive TSS websites require additional work for verification. Instead of calling the phone numbers listed on websites classified as passive TSS, we use clues mentioned previously to create TSS ground truth with reasonable confidence. For instance, in Section 3.3, we show that indeed, some of the passive scams are operated out of the same IP infrastructure that runs the aggressive ones, giving us confidence in creating ground truth on passive TSS websites based partially on this feature. Using this strategy, we were able to evaluate the performance of the classifier on a ground truth dataset consisting of 200 TSS websites and 200 Non-TSS websites, sampled randomly from the test set. Among the TSS websites, there were 100 aggressive and 100 passive TSS websites in the ground truth set. 114/200 (76 aggressive and 38 passive) TSS websites were verified via Malwarebytes and the remaining 86 (24 aggressive and 62 passive) websites were verified via a mixture of aforementioned clues. We note that some of these clues are better used as indicators/heuristics rather than conventional classifier features due to the inconsistent nature of some of these records – eg. WHOIS records [51].

	Predicted TSS	Predicted non-TSS	Total
Actual TSS	196	4	200
Actual non-TSS	1	199	200
Total	197	203	

Table 2: Confusion matrix for the TSS classifier on the testing set.

Table 2 shows the confusion matrix related to this experiment. The TSS classifier was able to achieve a reasonable 98% TPR and low 0.5% FPR on the testing set, thus validating the TSS website classification methodology. Also, there was 100% accuracy in distinguishing passive from aggressive TSS websites using the aforementioned heuristics. In the future, we seek to add more distinguishing features to our classifier and scale our experiment using additional independent sources of ground truth data.

2.5 Network Amplification Module

Using search listings to identify active TSS websites works well for creating an initial level of intelligence around these scams. However, it may be possible to expand this intelligence to uncover more domains supporting TSS that may have been missed by our crawler. The give-away for these additional TSS domains could be the sharing of network-level infrastructure with already identified TSS domains. A DNS request results in a domain name, d, being resolved to an IP address, ip, at a particular time, t, forming a (d, ip, t) tuple. Let $\mathcal{D}_{f-tss}$ be a set of labeled final-landing TSS domains. For each domain, $d \in \mathcal{D}_{f-tss}$, we compute two sets: (i) $RHIP(d)$, which is a set of all IPs that have mapped to domain d as recorded by the DNS Crawler (Section 2.3) within time window T, and (ii) $RHDN(ip)$, which is the set of domains that have historically been linked with the ip or $ip/24$ subnet in the $RHIP$ set within time window $T \pm \Delta$, where Δ is also a unit of time (typically one week). Next, we compute $\mathcal{D}_{rhip-rhdn}(d)$, which represents all the domains related to d at the network level, as discovered by the $RHIP$-$RHDN$ expansion. Now, for each domain $d' \in \mathcal{D}_{rhip-rhdn}(d)$, we check if the webpage $w_{d'}$ associated with it is a TSS webpage using the classifier module, Section 2.4. Only if it is true, we add d' to an amplification set, $\mathcal{D}'_{f-tss}(d)$, associated with d since co-location can sometimes be misleading [63]. The cardinality of the eventual amplification set gives us the amplification factor, $\mathcal{A}(d)$. Finally, we define the expanded set of TSS domains, $\mathcal{E}_{f-tss}$, as the union of all amplification sets. Combining the initial set of domains, $\mathcal{D}_{f-tss}$, with the expanded set, $\mathcal{E}_{f-tss}$, gives us the final set of fake-technical support domains $\mathcal{F}_{f-tss}$. The data pertaining to historic DNS resolutions comes from the ActiveDNS Project [3].

2.6 Clustering Module

The purpose of the clustering module is to identify different TSS campaigns. We identify the campaigns by finding clusters of related domain names associated with abuse in a given time period or epoch t. A two step hierarchical clustering process is used. In the first level, referred to as Network CLustering (NCL), we cluster together domain names based on the network infrastructure properties. In the second level, referred to as Application CLustering (ACL), we further separate the network level clusters based on the application level web content associated with the domains in them.

In order to execute these two different clustering steps, we employ the most common statistical features from the areas of DNS [38, 65] and HTML [61, 65] modeling to build our feature vector. In NCL, we use Singular Value Decomposition (SVD) [69] to reduce the dimensionality of the sparse feature matrix, and then use the X-Means clustering algorithm [58] to cluster domains having similar network-level properties. To further refine the clusters with ACL, we use features extracted from the full HTML source of the web pages associated with domains in $\mathcal{F}_{f-tss}$. We compute TF-IDF statistical vector on the bag of words on each cluster c [61]. Once we have the reduced application based feature vectors representing corresponding domains with SVD, this module too uses the X-Means clustering algorithm to cluster domains hosting similar content.

Campaign Labels: This submodule is used to label clusters with keywords that are representative of a campaign's theme. Let C be a cluster produced after NCL and ACL, and let D_C be the set of domains in the cluster. For each domain $d \in D_C$, we create a set $U(d, T)$ that consists of all the parts of the domain name d except the effective top level domain (eTLD) and all parts of the corresponding webpage title T. Next, we compute the set of words $W(U(d))$ using the Viterbi algorithm [43]. Using W, we increment the frequency counter for the words in a cluster specific dictionary. In this manner, after iterating over all domains in the cluster, we get a keyword to frequency mapping from which we pick the top most frequent word(s) to attribute to the cluster.

3 RESULTS

We built and deployed the system described in Section 2 to collect and analyze SR and AD domains for TSS. Although the system continues to be in operation, the results discussed in this section are based on data that was collected over a period of 8 months in two distinct time windows, April 1 to August 31, 2016 initially, and again between Jan 1 - Mar 31, 2017, to study the long running nature of TSS. We crawled 5 search engines for both ADs and SRs, which include Google.com, Bing.com, Yahoo.com, Goentry.com and search.1and1.com. Each day, the SEC module automatically sends 2,600 different queries, as discussed in Section 2.1 for technical support-related terms to the various search engines. We consider the top 100 SR URIs (unless there are fewer) while recording all the AD URIs displayed for each query.

3.1 Dataset Summary

In total we collected 14,346 distinct AD URIs and 109,657 distinct SR URIs. Table 3 presents the breakdown of all the search listings into the different categories. The AD URIs mapped to 4,954 unique Fully Qualified Domain Names (FQDNs), while the SR URIs mapped to 20,463 unique FQDNs. Among the AD URIs, 10,299 (71.79%) were observed as leading to TSS websites. This is a significant portion and shows that ADs related to technical support queries are dominated by those that lead to scams. It also means that the technical support scammers are actively bidding in the AD ecosystem to flood the AD networks with rogue technical support ADs, especially in response to technical support queries. Such prevalence of TSS ADs is the reason why Bing announced a blanket ban on online tech support ADs on its platform [7, 8] in mid-May, 2016. The TSS AD URIs mapped to 2132 FQDNs. Among the TSS AD URIs and corresponding FQDNs, we found the presence of both aggressive and passive websites. More than two thirds of the URIs were seen to lead to aggressive websites. The ratio between aggressive and passive websites was closer to 4:3 when considering just the TSS AD FQDNs. Past research has only investigated aggressive TSS websites, but our results show that passive websites are also a serious problem. We did observe legitimate technical support service AD URIs and FQDNs (13.19% of all AD URIs and 29.10% of all AD FQDNs).

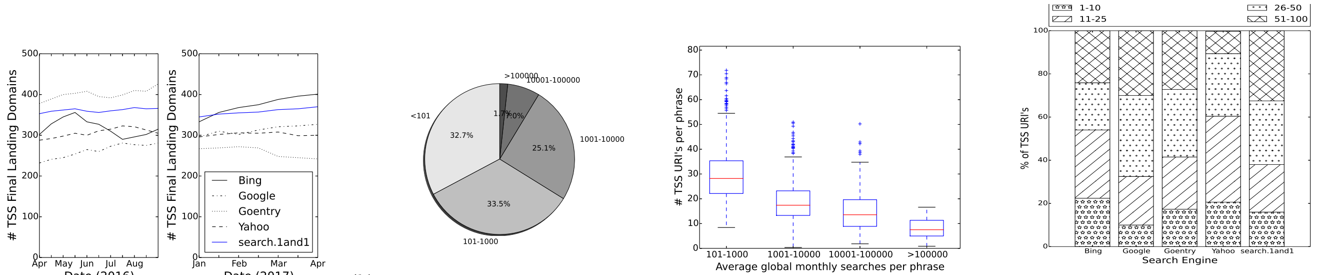
Among the SR URIs, 59,500 (54.26%) were observed leading to TSS websites. The URIs mapped to 3,583 (17.51%) FQDNs. Among the TSS SR URIs, we again found the presence of those leading to both aggressive and passive TSS varieties. The sheer number of such URIs is surprising as, unlike ADs, it is harder to manipulate popular search engine algorithms to make rogue websites appear in search results. However, as we discuss later, we observe that using black hat SEO techniques, TSS actors are able to trick the search engine ranking algorithms. Compared to ADs, we found that almost 76% TSS SR URIs lead to aggressive TSS websites while the remaining lead to passive TSS websites, again pointing to the prevalence of the common tactic of scare and sell [29]. Although TSS SR URIs were frequently seen interspersed in search results, SR URIs also consisted of non-TSS ones. Among these we observed 3.39% legitimate technical support service URIs, 9.13% blog/forum URIs, 9.12% URIs linked to complaint websites and 11.05% URIs pointing to news articles (mostly on TSS). The remaining 13.05% URIs were uncategorized.

We also report aggregate statistics for FQDNs after combining ADs and SRs data. We see that in total there were 5134 TSS FQDNs found, with URIs corresponding to 3166 FQDNs leading to aggressive websites and 1968 leading to passive websites. These together comprise of about 22.1% of the total number, 23,195 FQDNs retrieved from the entire dataset. One interesting observation is that majority of the FQDNs seen in ADs were not seen in the SRs and vice versa, with only a small amount of overlap in the TSS AD FQDNs and TSS SR FQDNs, consisting of 581 FQDNs. It suggests that the resources deployed for TSS ADs are different from those appearing in TSS SRs.

Support and Final-landing TSS domains: The purpose of *support* domains is to conduct black hat SEO and redirect victims to TSS domains but not host TSS content directly. We found 61.7% of the TSS search listing URIs redirected to a domain different from the one in the initial URI, while the remaining 38.3% did not redirect to a different domain. There were an additional, 2,435 *support* domains found. Moreover, popular URL shortening and redirection services such as bit.ly or goo.gl were noticeably missing.

	Advertisements (AD)				Search Results (SR)				AD+SR	
	URIs		Domains		URIs		Domains		Domains	
	#	%	#	%	#	%	#	%	#	%
TSS	**10,299**	**71.79**	**2,132**	**43.04**	**59,500**	**54.26**	**3,583**	**17.51**	**5,134**	**22.13**
Aggressive*	7,423	51.74	1,224	24.71	45,567	41.55	2,281	11.15	3,166	13.65
Passive	2,876	20.05	908	18.33	13,933	12.71	1,302	6.36	1,968	8.48
Non-TSS	**4,047**	**28.21**	**2,822**	**56.96**	**50,157**	**45.74**	**16,880**	**82.49**	**18,061**	**77.87**
Legitimate	1,892	13.19	1,442	29.10	3,726	3.39	3,499	17.09	3,790	16.34
Blogs/Forums	0	0.00	0	0.00	10,012	9.13	3,001	14.67	3,001	12.94
Complaint Websites	0	0.00	0	0.00	9,998	9.12	202	0.99	202	0.87
News	0	0.00	0	0.00	12,113	11.05	1,208	5.90	1,208	5.21
Uncategorized	2,155	15.02	1,380	27.86	14,308	13.05	8,970	43.84	9,860	42.51
Total	**14,346**	**100.00**	**4,954**	**100.00**	**109,657**	**100.00**	**20,463**	**100.00**	**23,195**	**100.00**

Table 3: Categorization of Search Results. * *Includes FakeCall, FakeBSOD, TechBrolo etc.*



(a) Bi-weekly trend of the number of final-landing TSS domains found.

(b) Fraction of technical support phrases with the corresponding average global monthly searches on Google during the months of threat data collection.

(c) Relationship between popularity of a search phrase and the TSS URI pollution levels in the search listings.

(d) Distribution of TSS SR URIs based on the position in search listings for different search engines.

Figure 4: Measurements related to AD and SR listings

When a TSS URI appearing in the search listings is clicked, it leads to the webpage that lures the victim into the scam. This webpage could be hosted on the same domain as the domain of the URI, or on a different domain. We refer to this final domain name associated with the TSS webpage as the *final-landing* TSS domain. Furthermore, it is possible that the path from the initial SR/AD URI to the final-landing TSS domain consists of other intermediate domains, which are mainly used for the purpose of redirecting the victim's browser. This is discussed in Section 2.3. Figure 4a plots the number of final-landing TSS domains discovered by our system over time across the various search engines. A bi-weekly trend shows that, across all search engines, we are able to consistently find hundreds of final-landing TSS domains and webpages. Bing, Google, Goentry, Yahoo and search.1and1.com, all act as origination points to TSS webpages. Starting mid-May 2016, we see a sudden dip in the number of TSS domains found on Bing. We suspect that this is most likely correlated to Bing's blanket ban on technical support advertisements [7, 8]. However, as we can see, activity, contributing mainly to SR based TSS, picked up again during July, 2016, continuing an upward trend in Jan to Mar 2017. Goentry, which was a major source of technical support ADs leading to final-landing TSS domains during our initial period of data collection saw a significant dip during the second time window. We suspect this may be due to our data collection infrastructure being detected (refer Section 5) or law enforcement actions against technical support scammers in India [18, 19], which is where the website is registered. In total we were able to discover 1,626 unique AD originated final-landing TSS FQDNs, and 2,682 unique SR originated final-landing TSS FQDNs. Together, we were able to account for 3,996 unique final-landing TSS FQDNs that mapped to 3,878 unique final-landing TSS TLD+1 domain names.

3.2 Search Phrases Popularity and SR Rankings

Since we use search queries to retrieve SRs and ADs, one may question the popularity of search phrases used in these queries. We use popularity level derived from Google's keyword planner tool [21] that is offered as part of its AdWords program. Figure 4b shows the distribution of technical support search phrases based on their popularity. We can see that out of the 2600 phrases associated with TSS, about one third (32.7%) were of very low popularity, e.g. *'kaspersky phone support'* with less than 100 average global monthly searches, one third (33.5%) were of low popularity, e.g. *'norton antivirus technical support'* with 101-1,000 hits per month on average, while there were 25.1% phrases that had medium levels of popularity, e.g. *'hp tech support phone number'* with 1,001-10,000 average hits. At the higher end, 7% of the technical support phrases had moderately high levels of popularity, e.g. *'dell tech support'*, *'microsoft support number'* with 10,001-100,000 hits per month on average, and 1.7% of the technical support search phrases were highly searched for, e.g. *'lenovo support'* with greater than 100,000 hits per month globally.

One may expect that less popular search terms are prone to manipulation in the context of both ADs and SRs, while more popular ones are harder to manipulate due to competition via bidding (in the case of ADs) or SEO (in the case of SRs). To validate this, we measure the number of total TSS URIs found per search phrase (referred to as *pollution* level), as a function of the popularity of the phrase. Since the popularity levels of phrases are gathered from Google, we only consider the TSS URIs (both AD and SR as seen on Google) to make a fair assessment. Figure 4c depicts a box plot that captures the pollution levels for all search phrases grouped

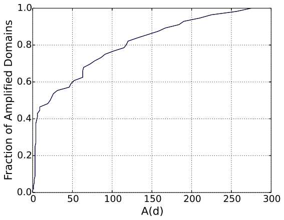


Figure 5: CDF of the network amplification factor, $\mathcal{A}$, of final-landing TSS domains discovered using search listings.

by the popularity levels except the ones with very low popularity. By comparing the median number of TSS URIs (depicted by the red line(s)) from different popularity bands, we witness that as the popularity level of a search term increases, the pollution level, decreases. We can make several additional observations: (i) there is definite pollution irrespective of the popularity level: in other words, more than a single TSS URI appeared in almost all of the technical support search queries we considered, as can be seen from the floor of the first quartile in every band; (ii) while many (~50%) low popularity search terms (e.g. those with 101-1000 hits per month) yielded 28 or more TSS URIs, there were outliers even among the high popularity search terms that accounted for the same or even more number of TSS URIs; and lastly, (iii) the range in the number of TSS URIs discovered per query varied more widely in the case of low popularity terms as compared to higher popularity terms.

To effectively target victims, it is not merely enough to make TSS URIs appear among the search results. It is also important to make them appear high in the search rankings. To measure this, we show the distribution of TSS SR URIs based on their ranking/position among the search results for different search engines. We use four brackets to classify the TSS SR URIs based on its actual position: 1-25 position (high rank), 26-50 position, 51-75 position and 76-100 position (low rank). If the same URI appears in multiple search positions, for example on different days, we pick and associate the higher of the positions with the URI. We do this to reflect the worst-case impact of a TSS SR URI. Thus, each unique URI is eventually counted only once. Figure 4d summarizes our findings. We see that all 5 search engines return TSS URIs that are crowding out legitimate technical support websites by appearing high in the rankings. For a more fine grained analysis of the rankings and its potential impact, out of the top 25 positions, we measured the fraction of TSS SR URIs appearing in the top three as well as the top ten positions. We found that Bing had the highest percentage, 8% of TSS SR URIs appearing among the top three positions and 17% TSS SR URIs appearing in a top ten spot. Even the other search engines had their top three and top ten search positions polluted regularly by TSS URIs. This makes it hard to trust a high ranking URI as legitimate.

3.3 Network Amplification Efficacy

The Network-level amplification helps us discover additional TSS domains. Dropping any domains having amplification factor $\mathcal{A}(d) < 1$, we are conservatively left with only 2,623 domains in the $\mathcal{D}_{f-tss}$ set that contributed to the *rhip-rhdn* expansion set, $\mathcal{E}_{f-tss}$. Figure 5 plots the cumulative distribution of the amplification factor of these domains. As we can see, around 60% domains had $\mathcal{A}(d) \leq 50$ while the remaining 40% domains had $\mathcal{A}(d) > 50$, with the maximum $\mathcal{A}(d)$ value equal to 275. In all, the total number of unique FQDNs hosting TSS content, $|\mathcal{F}_{f-tss}| = 9{,}221$, with 3,996 TSS FQDNs coming from the final-landing websites in search listings and 5,225 additional TSS FQDNs discovered as a result of network-level amplification. These 9,221 FQDNs mapped to 8,104 TLD+1 domains. Thus, even though amplification is non-uniform, it helps in discovering domains that may not be visible by search listings alone. The network amplification process also allowed us to identify 840 passive-type TSS domains co-located with one or more aggressive TSS domains. This indicates that some of the passive scams are operated by the same scammers who operate the aggressive ones.

TLD	%	TLD	%	TLD	%
com	25.56	org	4.86	co	1.89
xyz	16.21	in	4.44	tf	1.67
info	7.62	website	4.10	support	1.44
online	6.78	site	3.69	others	5.34
us	6.34	tk	2.03		
net	5.91	tech	2.12	Total	100

Table 4: Most abused top-level domains (TLDs) used in final-landing TSS websites.

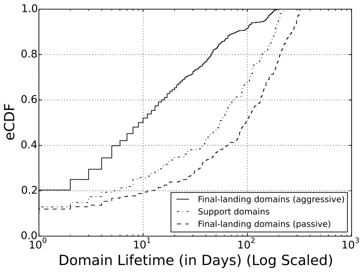


Figure 6: Lifetime of different types of TSS domains

3.4 Domain Infrastructure Analysis

In this section, we analyze all the domain names associated with TSS discovered by our system. This includes the final-landing domains that actually host TSS content as well as support domains, whose purpose is to participate in black hat SEO or serve as the redirection infrastructure.

Most abused TLDs: First, we analyze the final-landing TSS domain names. Table 4 shows the most abused TLDs in this category. The *.com* TLD appeared in 25.56% final-landing TSS domain names, making it the most abused TLD. Next, 16.21% domain names had *.xyz* as the TLD, making it the second most abused TLD. *.info*, *.online* and *.us* each had greater than 6% domain names registered to them completing the top five in this category. Other popular gTLDs included *.website, .site, .tech, .support*, while the ccTLDs included *.in, .tk, .co* and *.tf.* Among the *support* domains, the top three most popular TLDs were *.xyz, .win* and *.space.*

Domains Lifetimes: The lifetime of a final-landing TSS domain is derived by computing the difference between the earliest and most recent date that the domain was seen hosting TSS content. The lifetime of a support domain is derived based on earliest and the most recent date that the domain was seen redirecting to a final-landing TSS domain. Figure 6 plots the lifetimes of these two categories of domains with the final-landing domains split up into the passive and aggressive types. Final-landing TSS domains of the aggressive type had a median lifetime of ~9 days with close

Blacklist Name	Coverage (in %)		Type
	FQDN	TLD+1	
Malwarebytes TSS List	18.1%	n/a	Telephony BL
Google Safe Browsing	9.6%	5.2%	Traditional DBL
800notes.com	14.2%	n/a	Telephony BL
VirusTotal	22.6%	10.8%	Traditional DBL
Others$^+$	5.3%	3.4%	Traditional DBL
Cumulative	26.8%	12.5%	

Table 5: Overlap between final-landing TSS domains with popular public blacklists. $^+$includes Malware Domains List, sans, Spamhaus, itmate, sagadc, hphosts, abuse.ch and Malc0de DB.

Final-landing domains	Support domains	IPs	Phone Numbers	Clustering Label(s)	Sample Domains
662	452	216	521	microsoft virus windows	call-po-1-877-884-6922.xzz0082-global-wind0ws.website, virusinfection0x225.site
232	0	38	112	amazon kindle phone	kindlesupport.xyz
199	172	112	199	microsoft technician vista windows	talktoyour-technician.xyz
91	43	134	46	error microsoft threat	error-go-pack-akdam-0x00009873.website, suspiciousactivitydetectedpleasecallon18778489010tollfree.*.lacosta.cf
82	0	21	43	key office product	officesetupzone.xyz
76	0	36	38	antivirus norton	nortonsetup.online
75	0	18	28	browser firefox	firefoxtechnicalsupport.com
68	0	23	36	gmail login	gmailsupportphonenumber.org
55	0	41	51	chrome google	chromesupportphonenumber.com
48	22	42	47	apple risk	apple-at-risk.com, apple-atrisk.com
42	0	10	2	code error network	networkservicespaused.site, 04cve76nterrorcode.site
36	0	12	15	customer facebook service	facebooksupportphonenumber.com

Table 6: Selected large campaigns, as measured by the number of final-landing TSS domains, identified by the clustering module.

to 40% domains having a lifetime between 10-100 days, and the remaining ∼10% domains having a lifetime greater than a 100 days. In comparison, final-landing TSS domains of the passive type had a much longer median lifetime of ∼100 days. Some of the domains in this category had a lifetime of over 300 days. Clearly, passive TSS domains outlast those of the aggressive type. In comparison, support domains had a median lifetime of ∼60 days, with ∼33% domains having a lifetime greater than 100 days. Generally, this is a longer lifetime relative to final-landing TSS domains of the aggressive type. To provide context, phishing websites have a median lifetime of only 20 hours [54]. As we discuss later, in addition to blacklisting the final-landing domains, take down/blacklisting of these *support* domains would lead to a more effective defense in breaking parts of the TSS abuse infrastructure.

Overlap with Blacklists: Using domains and phone numbers from a large number of public blacklists (PBL) [1, 2, 16, 17, 20, 22–24, 28, 31, 32, 36], we verify if and when a TSS resource appeared in any of the PBLs. We collected data from these lists beginning Jan 2014 up until April 2017, encompassing the AD/SR data collection period, which allows us to make fair comparisons. Table 5 shows the overlap with several blacklists. Cumulatively, these lists cover only 26.8% FQDNs, that were found to be involved in TSS by our system. Moreover, out of the 26.8% blacklisted FQDNs, 8.2% were already present in one of the lists when our system detected them, while the remaining 18.6% were detected by our system ∼26 days in advance, on average. Moreover, when we cross-listed the *support* domains against these lists, we found that <1% of those were present in any of these lists. This analysis suggests that while exclusive TSS blacklists are a good idea alongside traditional PBLs, there is much scope for improvement by detecting these domains using an automated system such as ours.

3.5 Campaigns

The Clustering module (Section 2.6) produces clusters consisting of final-landing domains that share similar network and application features. Table 6 lists some of the major campaigns attributed by our system and the resources associated with them. First, although TSS are notoriously synonymous with Microsoft and its products, we found that many other brands are also targets of TSS campaigns. These brands include Apple, Amazon, Google and Facebook among others. Microsoft, however, remains on top of the most abused brands with 4 out of the top 5 TSS campaigns targeting Microsoft and its products. Second, we observed that TSS campaigns tend to advertise services targeted at particular brands and its line of products. This behavior is likely because the call center agents are trained to specialize in technical aspects associated with a particular type of product/service which could be a device (e.g. kindle), software (e.g. browser) or OS (e.g. Windows Vista) rather than generic technical support.

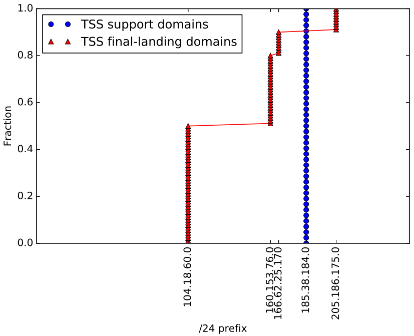


Figure 7: Fraction of Domains as a function of the IP address space.

4 CASE STUDY: BLACK HAT SEO TSS CAMPAIGN

In this case study, we analyze the largest TSS campaign from Table 6 to highlight the technique used to promote the TSS websites and the infrastructure used to grow and sustain the campaign over time. The campaign primarily targeted `Bing.com` users. It consisted of 452 support domains, 662 final-landing domains which mapped to 216 IPs over time and advertised 521 unique phone numbers. The campaign was first detected on 04/16/2016 and was active as recently as 03/30/2017. *SEO Technique*: The support domains use black hat SEO techniques sometimes referred to as *spamdexing* to manipulate the SRs. The support domains seen on the search page act as *doorway* pages to final-landing TSS domains. However, they use cloaking techniques such as text stuffing and link stuffing, consisting of technical support related keywords and links, to hide their real intent from search engine crawlers and get promoted up the SR rankings. It is surprising that these standard techniques still work.

IP Infrastructure Insights: We found that IP space used by support domains is quite different and decoupled from where final-landing TSS domains are located. Also, while the address space for fake technical support domains is fragmented, the entire set of support domains are concentrated in a single subnet, 185.38.184.0/24 as evident from Figure 7. IP to AS mapping for the subnet points to AS# 13213 under the name UK2NET-AS, GB. The ASN has country code listed as ME, Montenegro. The IP-Geo location data too points to an ISP in Montenegro, Budva. In contrast, IP's associated with final-landing TSS domains pointed to different AS#'s

31815, Media Temple, Inc, AS# 13335, Cloudflare and AS# 26496 GO-DADDY-COM-LLC based on IP to AS mapping data. They were geographically located in the US based on IP-Geo data. The fragmentation in the hosting infrastructure for the final-landing TSS domains gives the technical support scammers a reliable way to spread their assets. The decoupling of the infrastructure between support domains and final-landing TSS domains suggests that the technical support scammers are using the support domains as a "service" to offload the work of SEO.

5 DISCUSSION AND LIMITATIONS

Comparison with Past TSS studies: In this section, we compare our results with the findings of a previous study [53]. For a direct comparison, we were able to obtain data from Miramirkhani et al. for the period Jan-Mar 2017, which overlaps with the second time window of data collection in our work. Specifically, we received a list of 2,768 FQDNs discovered by their tool (2441 second-level domains), 882 toll-free phone numbers and 1,994 IP addresses. Upon intersecting these sets with our own data, we found 0/2,768 FQDNs and 0/2,441 second-level domains that were common. Moreover, in terms of server and telephony infrastructure, we discovered that the two datasets had 92/1,994 common IP addresses of servers hosting TSS and 5/882 common toll-free phone numbers. We also discovered frequent use of "noindex" [10] meta tags in the HTML source of webpages associated with domains in Miramirkhani et. al. dataset which was noticeably missing from webpages in our dataset. This indicates that TSS domains circulated via malvertising channels such ad-based url shorteners and typosquatting do not wish to be discovered by search engine crawlers, quite the opposite of search-and-ad TSS domains that are the focus of this work. Given this near-zero intersection of the two datasets and observations made above, we argue that our approach is discovering TSS infrastructure that ROBOVIC [53] is unable to find. An important contribution of our work is focusing on "passive" TSS which manifest mostly as organic search results. These pages are unlikely to be circulated over malvertising channels: a benign-looking tech support page is unlikely to capture the attention of users who were never searching for technical support in the first place.

Limitations: Like all real-world systems, our work is not without its limitations. Our choice of using PhantomJS for crawling search results and ads can, in principle, be detected by scammers who can use this knowledge to evade our monitors. We argue that replacing PhantomJS with a real browser is a relatively straightforward task which merely requires more hardware resources. Similarly, our choice of keeping our crawler stateless could lead to evasions which would again be avoided if one used a real, Selenium-driven, browser. Finally, while we conduct reviews for cluster quality, we expect to formally evaluate the efficacy of clustering with more ground truth data.

6 RELATED WORK

As mentioned throughout this paper, Miramirkhani et al. [53] performed the first analysis of technical support scams (TSS) by focusing on scams delivered via malvertising channels and interacting with scammers to identify their modus operandi. In recent work, Sahin et al. [60] investigated the effectiveness of chatbots in conversing with phone scammers. Researchers have identified the evolving role of telephony and how phone numbers play a central role in a wide range of scams, including Nigerian scams, phishing, phone-verified accounts, and SMS spam [40, 41, 45–47, 56, 65, 68].

In addition to telephony-specific work, researchers have analyzed a range of underground ecosystems detailing their infrastructure and identifying the parties involved, in addition to potential pressure points [42, 50, 55, 57, 64]. Since TSS is a type of underground ecosystem, we borrowed ideas found in prior work, such as, the appropriate setting of *User Agent* and *Referrer* crawler parameters used by Leontiadis et al. during their analysis of drug scams [50] to make requests appear as if they originated from a real user clicking on a search result. Also, search-redirection based drug scams discovered by them rely on compromising high-reputation websites while the TSS scams discovered by our system rely on black hat SEO and malicious advertisement tactics.

Finally, there have been numerous studies that cluster abuse/spam infrastructure and campaigns based on URLs [67], IP infrastructure [38, 39] and content [37]. Similar hierarchical clustering techniques too have been shown effective in multiple contexts [48, 49, 62, 65, 66]. In terms of countermeasures, prior work has shown the ineffectiveness of traditional blacklists in protecting services, such as instant messaging (IM) [59], and social media [44, 67]. Unfortunately, until blacklist curators adopt systems such as our own, blacklists will also be ineffective against TSS scams.

7 CONCLUSIONS

In this paper, we analyzed TSS by focusing on two new sources of scams: organic search results and ads shown next to these results. Using carefully constructed search queries and network amplification techniques, we developed a system that was able to find thousands of active TSS. We identify the presence of long-lived support domains which shield the final scam domains from search engines and shed light on the SEO tactics of scammers. In addition to aggressive scams, our system allowed us to discover thousands of passive TSS pages which appear professional, and yet display phone numbers which lead to scammers. We showed that our system discovers thousands of TSS-related domains, IP addresses, and phone numbers that are missed by prior work, and would therefore offer a marked increase of protection when incorporated into systems generating blacklists of malicious infrastructure.

Acknowledgments: This material is based upon work supported in part by the US Department of Commerce grants no. 2106DEK and 2106DZD, National Science Foundation (NSF) grant no. CNS-1514035, and Air Force Research Laboratory/Defense Advanced Research Projects Agency (DARPA) grant no. 2106DTX. For Stony Brook University, this work was supported by the Office of Naval Research (ONR) under grant N00014-16-1-2264 and by NSF under grants CNS-1617902 and CNS-1617593. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the US Department of Commerce, National Science Foundation, Office of Naval Research, Air Force Research Laboratory, or Defense Advanced Research Projects Agency.

REFERENCES

[1] Online. 800notes - Directory of UNKNOWN Callers. http://800notes.com/. (Online).

[2] Online. abuse.ch - the swiss security blog. https://www.abuse.ch/. (Online).

[3] Online. Active DNS Project. https://www.activednsproject.org/. (Online).

[4] Online. Alexa Topsites. http://www.alexa.com/topsites. (Online).

[5] Online. Bad Ads Trend Alert: Shining a Light on Tech Support Advertising Scams. http://bit.ly/2y2rbnq. (Online).

[6] Online. BeautifulSoup. https://pypi.python.org/pypi/beautifulsoup4. (Online).

[7] Online. Bing Ads bans ads from third-party tech support services. http://searchengineland.com/bing-bans-third-party-tech-support-ads-249356. (Online).

[8] Online. Bing brings in blanket ban on online tech support ads. https://goo.gl/6bgPFF. (Online).

[9] Online. Bing Search API. http://datamarket.azure.com/dataset/bing/search. (Online).

[10] Online. Block search indexing with ‘noindex’. https://support.google.com/webmasters/answer/93710?hl=en. (Online).

[11] Online. FTC - Tech Support Scams. http://bit.ly/1XIF9RV. (Online).

[12] Online. FTC Charges Tech Support Companies With Using Deceptive Pop-Up Ads to Scare Consumers Into Purchasing Unneeded Services. https://www.ftc.gov/news-events/press-releases/2016/10/ftc-charges-tech-support-companies-using-deceptive-pop-ads-scare. (Online).

[13] Online. FTC Obtains Settlements from Operators of Tech Support Scams. https://www.ftc.gov/news-events/press-releases/2017/10/ftc-obtains-settlements-operators-tech-support-scams. (Online).

[14] Online. Geek Squad Services - Best Buy. https://goo.gl/s7lWlq. (Online).

[15] Online. Google Custom Search. https://goo.gl/GyU7zP. (Online).

[16] Online. Google Safe Browsing. https://goo.gl/d1spJ. (Online).

[17] Online. hphosts. http://www.hosts-file.net/. (Online).

[18] Online. Indian police arrest alleged ringleader of IRS scam. http://money.cnn.com/2017/04/09/news/tax-scam-india-arrest-ringleader/. (Online).

[19] Online. India’s Call-Center Talents Put to a Criminal Use: Swindling Americans. http://nyti.ms/2xpFv8C. (Online).

[20] Online. I.T. Mate Product Support. http://support.it-mate.co.uk/. (Online).

[21] Online. Keyword Planner. https://adwords.google.com/KeywordPlanner. (Online).

[22] Online. Malc0de database. http://malc0de.com/database/. (Online).

[23] Online. Malware Domain List. https://www.malwaredomainlist.com/. (Online).

[24] Online. Malwarebytes Lab. https://blog.malwarebytes.com/tech-support-scams/. (Online).

[25] Online. N-Grams. http://stanford.io/29zsjAy. (Online).

[26] Online. PhantomJS. http://phantomjs.org/. (Online).

[27] Online. Python language bindings for Selenium WebDriver. https://pypi.python.org/pypi/selenium. (Online).

[28] Online. sagadc summary. http://dns-bh.sagadc.org/. (Online).

[29] Online. Scare and sell: Here’s how an Indian call centre cheated foreign computer owners. http://bit.ly/2oj2Rpz. (Online).

[30] Online. Searching For ‘Facebook Customer Service’ Can Lead To A Scam. http://n.pr/2kex6vU. (Online).

[31] Online. SPAMHaus Blocklist. https://www.spamhaus.org/lookup/. (Online).

[32] Online. Suspicious domains - sans internet storm center. https://isc.sans.edu/suspicious_domains.html. (Online).

[33] Online. Tech support scams persist with increasingly crafty techniques. https://goo.gl/cHHPDI. (Online).

[34] Online. Tech support scams remain at the top of the list of bad actors that search engines have to keep fighting. http://selnd.com/24jskRr. (Online).

[35] Online. Two Massive Tech Support Scams Shutdown by State of Florida, FTC. https://trustinads.org/2014/11/two-massive-tech-support-scams-shutdown-by-state-of-florida-ftc/. (Online).

[36] Online. VirusTotal. https://www.virustotal.com/. (Online).

[37] David S Anderson, Chris Fleizach, Stefan Savage, and Geoffrey M Voelker. Spamscatter: Characterizing internet scam hosting infrastructure.

[38] Manos Antonakakis, Roberto Perdisci, David Dagon, Wenke Lee, and Nick Feamster. USENIX Security 2010. Building a Dynamic Reputation System for DNS.

[39] Leyla Bilge, Engin Kirda, Christopher Kruegel, and Marco Balduzzi. NDSS 2011. EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis.

[40] Nathaniel Boggs, Wei Wang, Suhas Mathur, Baris Coskun, and Carol Pincock. ACSAC 2013. Discovery of Emergent Malicious Campaigns in Cellular Networks.

[41] Andrei Costin, Jelena Isacenkova, Marco Balduzzi, Aurélien Francillon, and Davide Balzarotti. International Conference on Privacy, Security and Trust (PST) 2013. The role of phone numbers in understanding cyber-crime schemes.

[42] Joe DeBlasio, Saikat Guha, Geoffrey M. Voelker, and Alex C. Snoeren. 2017. Exploring the Dynamics of Search Advertiser Fraud. In *Proceedings of the 2017 Internet Measurement Conference (IMC ’17)*. ACM, New York, NY, USA, 157–170. https://doi.org/10.1145/3131365.3131393

[43] Jr. Forney, G.D. 1973. The viterbi algorithm. *Proc. IEEE* 61, 3 (March 1973), 268–278. https://doi.org/10.1109/PROC.1973.9030

[44] Chris Grier, Kurt Thomas, Vern Paxson, and Chao Michael Zhang. ACM CCS 2010. @spam: the underground on 140 characters or less.

[45] Payas Gupta, Bharat Srinivasan, Vijay Balasubramaniyan, and Mustaque Ahamad. 2015. Phoneypot: Data-driven Understanding of Telephony Threats. In *22nd Annual Network and Distributed System Security Symposium, NDSS 2015, San Diego, California, USA, February 8-11, 2015.* The Internet Society. http://bit.ly/2wM1jff

[46] Jelena Isacenkova, Olivier Thonnard, Andrei Costin, Aurélien Francillon, and Davide Balzarotti. EURASIP J. Information Security 2014. Inside the SCAM Jungle: A Closer Look at 419 Scam Email Operations. (EURASIP J. Information Security 2014).

[47] Nan Jiang, Yu Jin, Ann Skudlark, and Zhi-Li Zhang. USENIX Security 2013. Greystar: Fast and Accurate Detection of SMS Spam Numbers in Large Cellular Networks Using Grey Phone Space.

[48] Shalini Kapoor, Shachi Sharma, and Bharat Srinivasan. 2014. Clustering devices in an internet of things (‘IoT’). (March 11 2014). US Patent 8,671,099.

[49] Shalini Kapoor, Shachi Sharma, and Bharat Ramakrishnan Srinivasan. 2013. Attribute-based identification schemes for objects in internet of things. (July 23 2013). US Patent 8,495,072.

[50] Nektarios Leontiadis, Tyler Moore, and Nicolas Christin. USENIX Security 2011. Measuring and Analyzing Search-redirection Attacks in the Illicit Online Prescription Drug Trade.

[51] Suqi Liu, Ian Foster, Stefan Savage, Geoffrey M. Voelker, and Lawrence K. Saul. 2015. Who is .Com?: Learning to Parse WHOIS Records. In *Proceedings of the 2015 Internet Measurement Conference (IMC ’15)*. ACM.

[52] Christopher D Manning and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*. Vol. 999. MIT Press.

[53] Najmeh Miramirkhani, Oleksii Starov, and Nick Nikiforakis. NDSS 2017. Dial One for Scam: A Large-Scale Analysis of Technical Support Scams.

[54] Tyler Moore and Richard Clayton. 2007. Examining the impact of website take-down on phishing. In *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*. ACM, 1–13.

[55] Marti Motoyama, Kirill Levchenko, Chris Kanich, Damon McCoy, Geoffrey M Voelker, and Stefan Savage. 2010. Re: CAPTCHAs-Understanding CAPTCHA-Solving Services in an Economic Context.. In *USENIX Security Symposium*.

[56] Ilona Murynets and Roger Piqueras Jover. IMC 2012. Crime scene investigation: SMS spam data analysis.

[57] Youngsam Park, Jackie Jones, Damon McCoy, Elaine Shi, and Markus Jakobsson. NDSS 2014. Scambaiter: Understanding targeted nigerian scams on craigslist.

[58] Dan Pelleg, Andrew W Moore, et al. ICML 2000. X-means: Extending K-means with Efficient Estimation of the Number of Clusters.

[59] Iasonas Polakis, Thanasis Petsas, Evangelos P. Markatos, and Spyros Antonatos. NDSS 2010. A Systematic Characterization of IM Threats using Honeypots.

[60] Merve Sahin, Marc Relieu, and Aurélien Francillon. SOUPS 2017. Using chatbots against voice spam: Analyzing LennyâĂŹs effectiveness.

[61] Gerard Salton and Michael J. McGill. 1986. *Introduction to Modern Information Retrieval.* McGraw-Hill, Inc., New York, NY, USA.

[62] Shachi Sharma, Shalini Kapoor, Bharat R. Srinivasan, and Mayank S. Narula. 2011. HiCHO: Attributes Based Classification of Ubiquitous Devices. In *Mobile and Ubiquitous Systems: Computing, Networking, and Services - 8th International ICST Conference, MobiQuitous 2011, Copenhagen, Denmark, December 6-9, 2011, Revised Selected Papers (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering)*, Alessandro Puiatti and Tao Gu (Eds.), Vol. 104. Springer, 113–125. https://doi.org/10.1007/978-3-642-30973-1_10

[63] Craig A. Shue, Andrew J. Kalafut, and Minaxi Gupta. 2007. The Web is Smaller Than It Seems. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement (IMC ’07)*. ACM, New York, NY, USA, 123–128. https://doi.org/10.1145/1298306.1298324

[64] Kyle Soska and Nicolas Christin. USENIX Security 2015. Measuring the Longitudinal Evolution of the Online Anonymous Marketplace Ecosystem.

[65] Bharat Srinivasan, Payas Gupta, Manos Antonakakis, and Mustaque Ahamad. 2016. Understanding Cross-Channel Abuse with SMS-Spam Support Infrastructure Attribution. In *Computer Security - ESORICS 2016 - 21st European Symposium on Research in Computer Security, Heraklion, Greece, September 26-30, 2016, Proceedings, Part I (Lecture Notes in Computer Science)*, Ioannis G. Askoxylakis, Sotiris Ioannidis, Sokratis K. Katsikas, and Catherine A. Meadows (Eds.), Vol. 9878. Springer, 3–26. https://doi.org/10.1007/978-3-319-45744-4_1

[66] Bharat Ramakrishnan Srinivasan. 2017. *Exposing and Mitigating Cross-Channel Abuse that Exploits the Converged Communications Infrastructure.* Ph.D. Dissertation. Georgia Institute of Technology.

[67] Kurt Thomas, Chris Grier, Justin Ma, Vern Paxson, and Dawn Song. IEEE Symposium on Security and Privacy 2011. Design and Evaluation of a Real-Time URL Spam Filtering Service.

[68] Kurt Thomas, Dmytro Iatskiv, Elie Bursztein, Tadek Pietraszek, Chris Grier, and Damon McCoy. ACM CCS 2014. Dialing back abuse on phone verified accounts.

[69] Michael E Wall, Andreas Rechtsteiner, and Luis M Rocha. 2003. Singular value decomposition and principal component analysis. In *A practical approach to microarray data analysis*. Springer, 91–109.