Automatic Detection and Analysis of the "Game Hack" Scam

Emad Badawi[1,*], Guy-Vincent Jourdan[1], Gregor Bochmann[1] and Iosif-Viorel Onut[2]

[1]*Faculty of Engineering, University of Ottawa, Ottawa, Canada*
[2]*IBM Centre for Advanced Studies, Ottawa, Canada*
E-mail: ebadawi@uottawa.ca; GuyVincent.Jourdan@uottawa.ca; Bochmann@uottawa.ca; vioonut@ca.ibm.com
[*]*Corresponding Author*

Received 30 August 2019; Accepted 03 December 2019; Publication 16 December 2019

Abstract

The "Game Hack" Scam (GHS) is a mostly unreported cyberattack in which attackers attempt to convince victims that they will be provided with free, unlimited "resources" or other advantages for their favorite game. The endgame of the scammers ranges from monetizing for themselves the victims time and resources by having them click through endless "surveys", filing out "market research" forms, etc., to collecting personal information, getting the victims to subscribe to questionable services, up to installing questionable executable files on their machines. Other scams such as the "Technical Support Scam", the "Survey Scam", and the "Romance Scam" have been analyzed before but to the best of our knowledge, GHS has not been well studied so far and is indeed mostly unknown.

In this paper, our aim is to investigate and gain more knowledge on this type of scam by following a data-driven approach; we formulate GHS-related search queries, and used multiple search engines to collect data about the websites to which GHS victims are directed when they search online for various game hacks and tricks. We analyze the collected data to provide new insight into GHS and research the extent of this scam. We show that despite

doi: 10.13052/jwe1540-9589.1881

its low profile, the click traffic generated by the scam is in the hundreds of millions. We also show that GHS attackers use social media, streaming sites, blogs, and even unrelated sites such as *change.org* or *jeuxvideo.com* to carry out their attacks and reach a large number of victims.

Our data collection spans a year; in that time, we uncovered 65,905 different GHS URLs, mapped onto over 5,900 unique domains. We were able to link attacks to attackers and found that they routinely target a vast array of games. Furthermore, we find that GHS instances are on the rise, and so is the number of victims. Our low-end estimation is that these attacks have been clicked at least 150 million times in the last five years. Finally, in keeping with similar large-scale scam studies, we find that the current public blacklists are inadequate and suggest that our method is more effective at detecting these attacks.

Keywords: Game scam, scam analysis, fraud detection, cyberattack.

1 Introduction

Game developers depend mostly on the purchase of in-game resources as well as in-game advertisements to make a profit [17, 18], Figures 1 and 2 are examples of in-game resources. To obtain these resources, some players are willing to bypass the regular route and use “cracks”, game-modifying software, or any other means of hacking. For these reasons, games are a rich ground for hackers.

In this paper, we study an understudied social engineering attack targeting games players. We call this attack the Game Hack Scam (GHS). Usually, GHS starts when a victim searches for cheats and hacks for their game using search engines, social media, streaming sites, blogs, or any other site. The returned search results may directly contain GHS instances such as Figure 3. In other cases, the search results link to pages that have links to GHS instances such as Figure 4.

In GHS, the attackers claim that they can hack a specific game and provide the victim with free, unlimited resources or other advantages for their favorite game. To obtain these claimed advantages, the victims are asked to complete one or more tasks, called “offers”. These so-called offers represents the final payload and include, but are not limited to, clicking through endless “surveys”, filing out “market research” forms, collecting personal information, getting the victims to subscribe to questionable services, installing questionable executable files on their machines, etc.

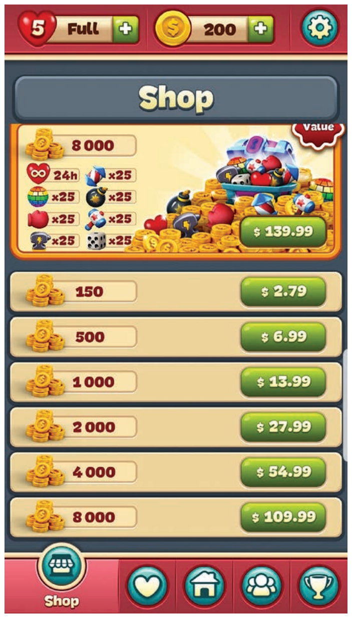


Figure 1 In-game resources for the game Toon Blast.

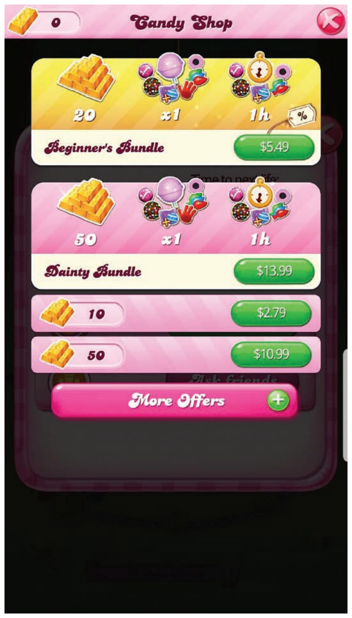
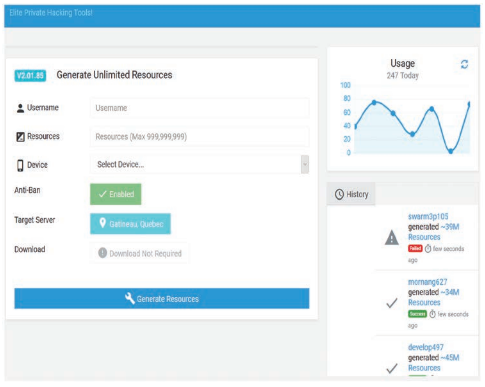


Figure 2 In-game resources for the game Candy Crush.



Generate Unlimited Resources

Username: Username

Resources: Resources (Max 999,999,999)

Device: Select Device...

Anti-Ban: ✓ Enabled

Target Server: Gatineau, Quebec

Download: Download Not Required

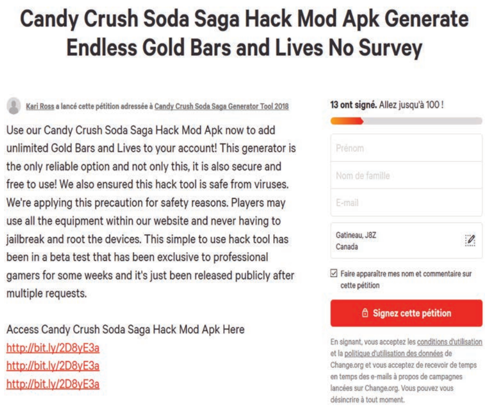
247 Today

swarm3p105 generated ~39M Resources Failed few seconds ago

mornang627 generated ~34M Resources Success few seconds ago

develop497 generated ~45M Resources

Figure 3 GHS instance.


Candy Crush Soda Saga Hack Mod Apk Generate Endless Gold Bars and Lives No Survey

Kari Ross a lancé cette pétition adressée à Candy Crush Soda Saga Generator Tool 2018

Use our Candy Crush Soda Saga Hack Mod Apk now to add unlimited Gold Bars and Lives to your account! This generator is the only reliable option and not only this, it is also secure and free to use! We also ensured this hack tool is safe from viruses. We're applying this precaution for safety reasons. Players may use all the equipment within our website and never having to jailbreak and root the devices. This simple to use hack tool has been in a beta test that has been exclusive to professional gamers for some weeks and it's just been released publicly after multiple requests.

Access Candy Crush Soda Saga Hack Mod Apk Here
http://bit.ly/2D8yE3a
http://bit.ly/2D8yE3a
http://bit.ly/2D8yE3a

13 ont signé. Allez jusqu'à 100 !

Prénom

Nom de famille

E-mail

Gatineau, J8Z
Canada

☑ Faire apparaître mes nom et commentaire sur cette pétition

Signez cette pétition

En signant, vous acceptez les conditions d'utilisation et la politique d'utilisation des données de Change.org et vous acceptez de recevoir de temps en temps des e-mails à propos de campagnes lancées sur Change.org. Vous pouvez vous désincrire à tout moment.

Figure 4 Clean page with a link leads to GHS.

Usually, attackers carry out GHS attacks by creating an online website targeting their victims. We call these websites “generators”. These generators are carefully designed web pages which attempt to convey to the victim the advanced technical abilities of the scammer and a large, satisfied user base for the GHS instance. GHS instances tend to use a variety of similar templates. Many of these templates ask for the victim’s identifier on the game and the resources that the victim wants. Other templates attempt to be more convincing by asking for additional information such as the game platform, the region they live in, and the ability to use a proxy. Also, these advanced templates could display a fake chat box, and a pop-up showing claimed current users and the number of resources they supposedly gained.

Once the information is provided, the generator page pretends to perform some hacking process. After that, a pop-up appears claiming that the hack was successful and the victim then is invited to a "verification" step. During this verification process, some screen is shown to the user, asking to complete one or more "offers". This type of screens is called a "content-locker" (CL) by the creators of these scams. The "CL" with its set of offers is what the scammer ultimately wants the victim to see, as they lead to the payload. In some cases, the generator is bypassed and the victim is directly presented with the CL, the payload, or a modified version of the game APK.

In this paper, we extend and update our previous work in [4]. We have updated our classifier to include more GHS samples. This increased our classification accuracy. Also, we incorporated new search queries using google trend [43] services. We have updated our results using the newly discovered GHS instances. Moreover, we have collected 59 executable files that were reported as harmful by virus total [41], and many of them were reported by locally-installed anti-virus scanners as well. Finally, we have collected more than 400 modified Android games APKs and compared them to their respective APKs from google play.

Our main contributions are the following:

- We gave insight into a new type of scam that targets games players.
- We uncovered more than 5,900 GHS-related second-level domains, and 375 offers second-level domains.
- We show that the attackers routinely target a vast array of games.
- We show that the existing public blacklists (PBLs) are ineffective against this type of scam.
- By analyzing the GHS URLs that are shortened by Bitly, we estimate that these attacks have been clicked at least 150 million times since mid-2014.

All the data used in our study is available at http://bit.ly/GHSJWE.

2 Methodology

In our work, we have developed a data-driven approach to collect, detect and analyze GHS. We started our work by utilizing our previous research results in [4] to prepare a representative data-set for our model. This helped us get a large data set with a variety of samples, which we used to collect and identify more GHS instances. Figure 5 describes our system. It includes

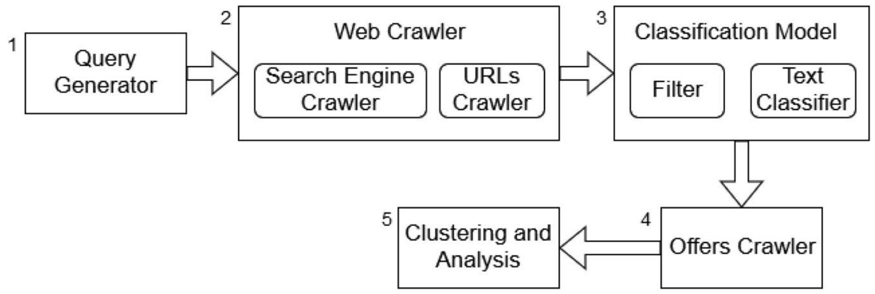


Figure 5 Games scam detection and analysis model.

five modules: Search query generator, Web Crawler, Classifier Model, GHS instances triggering, and Analysis.

2.1 Search Query Generator

Choosing good search queries that have a high likelihood to lead to the scam pages is an important task. Kharraz et al. [19] used Google Trends service to generate such queries. Srinivasan et al. [49] used a probabilistic analysis. In our work, we have used a combination of all of these techniques to cover as many GHS instances as possible. We started from our initial corpus of pages leading to GHS instances as well as the GHS instance pages themselves. We extracted the bag of words from our corpus. We found 1,964 words that have a frequency greater than ten. We selected manually 39 of these words based on their direct connection to GHS and added the stop words "without" and "no". We then generated our queries using the Markov assumption [16] to approximate n-gram probabilities. We generated our n-grams for $n = 3$ to 7[1]. That gave us 795 n-grams, and we manually selected 410 search queries from them. The full details are available at http://bit.ly/GHSJWE.

We then created a list of 966 game names by extracting Facebook, Google, and iTunes top games, and we combined each of these game names with 9 of our n-grams, thus getting 8,694 new queries, for a total of 9,104 queries.

Finally, we have used the Google trend API [43] to generate GHS related queries. Google trend reflects the popularity of search queries as it used by normal web users. We have crawled Google trend API twice to generate the search queries. In the first crawling, we used the 9,104 queries generated

[1]Our experiments showed that 8-grams and up did not improve our results.

above as search terms. After filtering the non GHS related queries, we acquired 972 new GHS related queries. In the second crawling, we used the 972 newly generated queries as a seed, and generated 872 more queries.

Our final query list contains 10,708 search queries.

2.2 Web Crawler

Our search crawler uses the previously identified search queries as a seed to search daily for GHS pages using *Google.com*, *Bing.com*, *search.yahoo.com*, and *search.1and1.com*. For each query, we only consider the first and second pages (that is, 20 search results) returned by each engine. The crawler is based on ChromeDriver [37] and Python Selenium [40]. Using Python beautifulsoup [34] and the CSS selectors, we extract and crawl the URLs resulting from our searches. For the crawling process, we use a lightweight scripted headless browser built using python by integrating Selenium [40], ChromeDriver [37] and BeautifulSoup [34]. We collect data about the crawled URLs including URL redirections, HTML contents, a screen-shot of the landing page and its resources (scripts, CSS files etc.).

2.3 Classification Module

The majority of the pages that our crawler collects are not directly GHS instances. Instead, they are often either pages hosted on benign sites with URL links that lead to a GHS instance or completely benign pages related to games. Therefore, we need a classifier to automatically identify actual GHS instances in our results.

In this paper, we have used two steps classification model to detect GHS instance. Our model achieved a very high accuracy with 99.95% True Positive Rate and only 0.16% False Positive Rate. The full analysis is conducted in Section 3.

2.4 Interacting with GHS Instances

To collect information about the GHS instances, we need to interact with them, provide the necessary inputs and follow each GHS instances instructions in order to reach the final stage, at which point the list of "offers" is provided. Following these offers, the victim is asked to provide personal information, subscribe to fraudulent services or install malware. We collect for analysis the set of offers that are provided by the GHS instances we have found. We did interact with these pages manually to extract the CLs URLs, this is planned to be automated in our future research.

2.5 Clustering and Analysis

We then conduct four different analyses of the different pages and domains that our model identified as scam. Our first analysis is done on the GHS instances themselves. We use the identifiers found in the pages to detect similarities and infer common ownership of the GHS instances. Our second analysis is done on the "offers", the final step in the scam life-cycle. We classify the different types of offers and show the convergences into smaller set of offers. Our third analysis is done on the domain names hosting the GHS instances and the offers. We also study the effectiveness of the current PBLs against GS. Our fourth and last analysis is done on these GHS instances that use the URL shortener Bitly. Bitly provides publicly available statistics for its URLs, which in turns gives us a unique insight into the effectiveness and the trends of the scam.

Our analyses are presented in Sections 4 and 6.

3 Classifier

In our crawling process, the majority of the URLs we collect are either benign pages having nothing to do with GHS, or benign pages with links to GHS instances. To filter out GHS instances, we have developed a two-step classification model. In the first step, we use features that we extracted from the GHS instances DOM. These features are used to filter out the crawled pages as GHS instances, benign pages, or unidentified. In the second step, we feed the unidentified pages to a text-based SVC classifier. These pages are then flagged GHS instances or benign. Our classification model achieved a very high accuracy, with True Positive Rate (TPR) above 99.9% and False Positive Rate (FPR) lower than 0.2%.

In our analysis, True Positive (TP) refers to the number of scam pages actually classified as scam, True Negative (TN) refers to the number of benign pages classified as benign, False Positive (FP) refers to the number of benign pages wrongly classified as scam and finally False Negative (FN) refers to the number of scam instances wrongly classified as benign. From these basic measure, the F1 score is derived as follows:

$$F1 = 2 * (Precision * Recall)/(Precision + Recall) \tag{1}$$

Where,

$$Precision = TP/(TP + FP) \tag{2}$$

$$Recall = TP/(TP + FN) \tag{3}$$

The higher F1, the better.

3.1 Data-Set Construction

In our previous study [4], we have manually selected GHS instances and benign pages to train a SVC classifier. Our classifier had a TPR of 96.7% and a FPR 2.1%. We have used the classifier to filter out the URLs we have collected over five months from May to September 2018. Through that period, we have collected over 33k GHS instances, but this list contains some noise. In this paper, we use these GHS instances as a seed to prepare a representative GHS data set to train our new model.

Our previous study showed that the GHS instances are often based on similar templates that the scammers create and distribute. These templates usually have a similar DOM structure as well as similar text. The scammers only need to change the game name and the name of the in-game resources to create a new scam instance. We used this finding to filter out some of the noise in our initial data set. We have used the clustering method used by Cui et al. [12] cluster together pages with similar DOM structure. We then manually inspected two to three pages selected randomly from each cluster, to flag the clusters as true or false positive. This allowed us to identify and remove hundreds of pages wrongly classified as GHS instances. We then divided our data into two different data-sets; A and B. Data-set A contains one GHS instance for each TP cluster. The instance representing the cluster is randomly selected from within that cluster. This process resulted in a data-set of 835 GHS instances, which are meant to represent the entire data-set. We used that data-set for training. The remaining 31,095 GHS instances, which we call Data-set B, were not used for training nor for testing. We still validated these approach by showing that training on data-set A was sufficient to classify data-set B, as explained in Appendix A.

The same approach was applied to a set of 8 k pages classified as clean to prepare a benign training data set. We have randomly picked 1,079 clean pages for data-set A, and 7,939 clean pages for data-set B.

Table 1 Results of a 10-Fold cross-validation on the five classifiers

Classifier	Page Type	Classified Clean	Classified GHS	F1
SVC	clean	1075	4	99.57
	gen	3	832	
NB	clean	1060	19	98.26
	gen	10	825	
Kneighbors	clean	1017	62	95.26
	gen	19	816	
Random Forest	clean	1068	11	97.94
	gen	23	812	
MLP	clean	1072	7	99.21
	gen	6	829	

3.2 Text Classifier

To evaluate our classifiers, we used 10-fold cross-validation on the labeled data-set A we prepared in Section 3.1. We ran our experiments using five different classifiers: Vector Classifier (SVC), Naive Bayes (NB), Kneighbors, Random Forest, and Multi-layer Perceptron (MLP) classifiers. We used these five classifiers to classify the crawled pages based on the text as seen by the end-user.

Table 1 present the results obtained with the five classifiers. As we can see, the SVC text classifier achieved the highest results with 99.57 F1 score followed by MLP with 99.21 F1 score. The other classifiers also performed fairly well with Kneighbors having the lowest F1 score value equals to 95.26. Based on these results, we used the SVC classifier throughout our experiments in this paper.

3.3 Filters

A further improvement on the classifier can be obtained by directly flagging the pages that are easily recognized as GSH or as clean, and only using the classifier on the other pages. There are two benefits to this additional step: first it is much faster, and second, we can reduce both FPR and FNR.

Through manual inspection of more than 100 randomly selected GHS instances from our training set, we have identified two distinguishing features:

- "Content Locker": Many of the GHS instances contain the template provider identifier or English terms related to generators. The presence of such terms can be a good indicator of GHS instances. To employ "Content Locker" as a feature, we search for the presence of the template

Table 2 The effect of applying the filters on the training data-set A

Classifier	#FP	#FP Detected by Filter	#FN	#FN Detected by Filter
SVC	4	2	3	1
NB	19	6	10	5
Kneighbors	62	9	19	8
Random Forest	11	3	23	7
MLP	7	3	6	2

provided identifier or the generator terms in the targeted page text. We report the value of this feature as a boolean value where true means that "Content Locker" exists.

- "Hack button": GHS instances usually contain a button meant initiate the fake hacking process, usually alongside a text such as "generate" or "detect device". For this feature, we simply count the number of tags related to buttons. We include the tags <button>, the tag <input> when the type is "button", and any other tag with "class" or "id" related to buttons.

These features are used to classify the crawled pages as either GHS instances, benign pages, or unidentified. By setting the values of "content locker" to true and "hack button" threshold to two, we were able to filter 415 (49.7%) of the GHS instances without introducing any false positive. We did find the threshold for "hack button" by trial and error. Additionally, negating the values of "content locker" and "hack button" threshold filtered out 10% of the clean pages without introducing any false negative. Using these filters reduces the detection execution time: the filter feature's extraction requires 67 microseconds on average, while, for example, the SVC classifier requires 25,793 microseconds on average.

Table 2 presents the result of applying the filters on our training data-set A. As shown in the table, using the filtering step improves the performance of all the classifiers used in Section 3.2. Our filter detects many of the FN and FP pages before applying the classifier. Thus these pages will not be classified erroneously.

4 Results and Analysis

We used our university's server as well as Compute Canada dedicated servers [8] to deploy the model mentioned above to collect the possible GHS pages. The results reported in this paper come from data collected over a year from May 2018 to May 2019.

In this section, we discuss the results obtained from the analysis of the data collected. We first present the basic numbers we got. We then shed some light over the GHS instances, their similarities, and the games they targets. We also show that scammer relies on pre-built templates to create new attacks without any technical knowledge. After that, we study the offers reached when interacting with the GHS instances. Finally, we look at the domain names used by servers hosting the GHS instances and the offers. We show that public blacklists are mostly ineffective against GHS.

4.1 Classification Result

Our system identified 65,905 different GHS instances URLs, mapped onto 5,930 unique second-level domains. 3,193 (53.8%) of these domains have only 1 GHS URL. On the other hand, 739 (12.46%) of these domains host more than 9 GHS URLs. Moreover, there are many domains with a large number of GHS instances URLs. The largest three domains hosted 4,664, 2,762, and 2,439 URLs, respectively. Almost 50% of the GHS instances where identified during the filtering process. Our previous studies [4] showed that none of the top 1 K Alexa domains contains actual GHS instances, only links to GHS instances. Based on these findings, we only report results for the URLs hosted on domains outside Alexa top 1 k.

In our analysis, we have trained our classifier on pages with English text only. Thus, we focus our research on pages with English text and ignore the other crawled pages.

4.1.1 Search URLs Classification

Throughout our crawling period, we have collected 775,961 different pages, 679,514 of which are in English. Our classifier identified 41,383 of these pages as GHS instances, which means our search queries yield an instance of the scam 6.09% of the time. The number of GHS instances identified per month is presented in Figure 6. On average, our model detected 2,009 GHS instances per month in the period from July 2018 until Jan 2019. This number has increased to 5,788 GHS instance per month in the last four months of crawling, after incorporating Google Trend services to generate new search queries and crawling the second page of each search engine. Figure 7 present the percentage of the GHS instances found in the second page of the search engines over the full found GHS instances. As seen in the figure, most of the GHS instances were found on the second page, this in turns explains the increase of our model performance in detecting GHS instances.

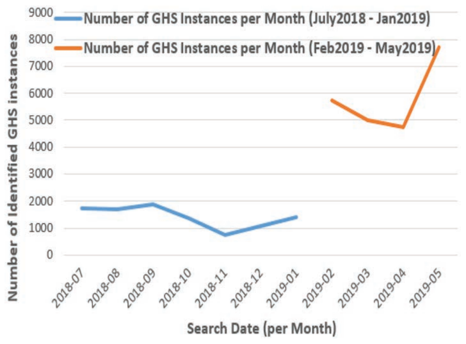


Figure 6 Number of GHS instances found by search engines per month.

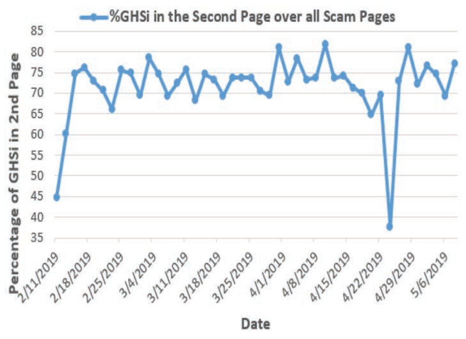


Figure 7 Percentage of GHS instances found in the second page over all scam pages.

4.1.2 Extracted URLs Classification

We now look at the pages with links yielding to GHS instances. Overall, we have crawled 3 M URLs that we extracted from the benign pages returned by the search engine. Out of these, we were able to reach and save 999,573 pages with English text. Our classifier identified 24,522 of these pages as GHS instances.

Our analysis shows that some of the domains that contain URLs yielding to GHS are blogs and domains with high traffic. This suggests that attackers target these domains to reach more victims. We found links leading to GHS instances in posts hosted in *Jeuxvideo.com*, *Groups.Google.com*, *Pinterest.com*, *change.org*, *Youtube.com*, and *npm.runkit.com*.

4.2 GHS Analysis

In this section, we present our analysis of the GHS instances we collected. Here we provide an insight into the relationship between the different

GHS instances. We first cluster GHS instances into groups based on unique identifiers that we have found in the GHS instances. We then look at the set of games that are targeted by related scammers.

4.2.1 GHS Groups

This analysis was conducted based on the finding that many GHS pages are built using similar templates. We found at least two different online advertisement websites that either provide GHS instance templates or provide tutorials on how to copy existing templates and deploy them in the scam. An example of GHS templates is presented in Figure 8.

We manually inspected the DOM of several GHS pages in search of identifiers that can be used to map the scam instances to the attacker publishing them, and identified eight such identifiers. Some of these identifiers relate to analytic collections. For example, we found links to the site *histats.com* in 26,756 of our GHS instances, about a third of them and *statcounter.com* in 5,244 of the pages. It does not mean that either *histats.com* or *statcounter.com* have any part in the scam, merely that scammers tend to use these sites for their analytic. Other identifiers commonly found in the DOM of the GHS instances relate to the sites that provide the GHS templates and offers at the end of the scam. Identifiers for these two websites appear in approximately a third of our GHS instances, with 21,989 occurrences. To confirm our findings, we have created our own attack[2]. The attack can be reached at https://dwnlds.co/3396a94.

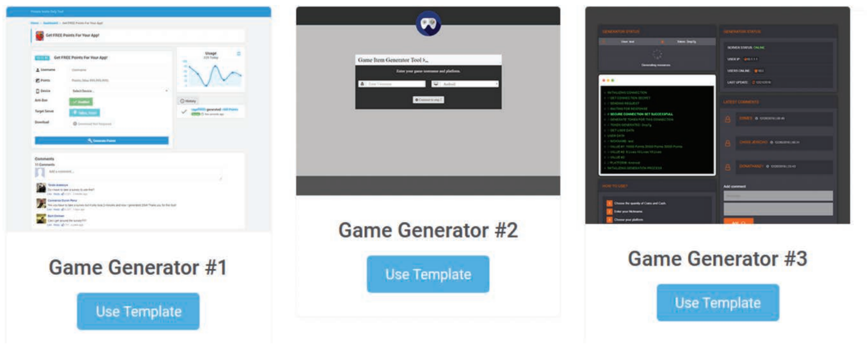


Figure 8 Examples of GHS templates.

[2]Of course; we did not deploy these attack, so no one was victimized by our tests.

Some of the identifiers have unique ID values for each account, we assumed that each ID belongs to a different attacker, as suggested by our experiments. Some of the IDs appear in more than one GHS pages, which suggests that the set of pages containing the same ID belong to the same attacker. Overall, we have identified 8,450 unique ID values for the eight identifiers. 8,009 of them (94.7%) span less than five pages; thus, we excluded them to reduce the skew in our analysis. In the subsequent analysis, we used the 441 IDs that spans at least five pages. The breakdown of these IDs is shown Figure 10.

4.2.2 Targeted Games

Having identified clusters of attacks belonging to the same attackers, we then turn out attention to the targets of these related attacks. In particular, we wanted to understand why a given attacker would carry several attacks: was it to avoid detection, or was it to cast a wider net?

To answer this question, we looked at the actual games targeted by related GHS instances. We have extracted around 40 k different game titles from our database of GHS instances. Some titles have a great number of occurrences. These are typically "generic" titles with no particular targeted game. We have identified 14 such titles. The top three are **"Generate Resources For Your Game!"** with 1,287 occurrences,**"Resource Generator"** with 590 occurrences and **"Generate Points For Your App!"** with 203 occurrences. We removed these pages since they provide no added value in this analysis. In this analysis, we only consider the attacker's IDs that span at least five pages in our database, since we are interested in trends among the attackers publishing several attacks.

Figure 9 shows our results. The x-axis represents the number of unique game title over the number of related pages, and the y-axis represents the fraction of unique attacker's IDs. We can see that around 2/3 of the attackers have at least 50% diversity in the game title they target, i.e., 50% of the games they target are unique. Moreover, 20% of the attackers target each game title only once. This clearly suggests that the attackers generate new attacks primarily to cover new games and increase the spread of their scams.

4.3 Offers

The offers are the last stage in the GHS. Usually, they appear after the victim provides their game credentials and the fake hacking process starts. At this

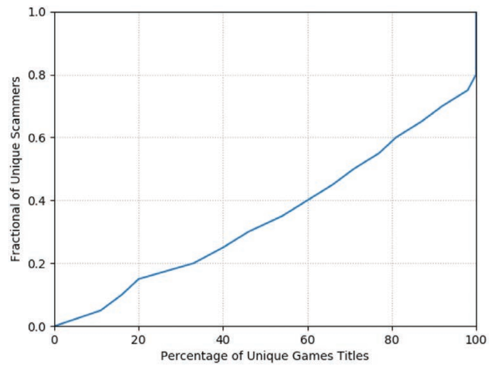


Figure 9 Number of games each scammer spans.

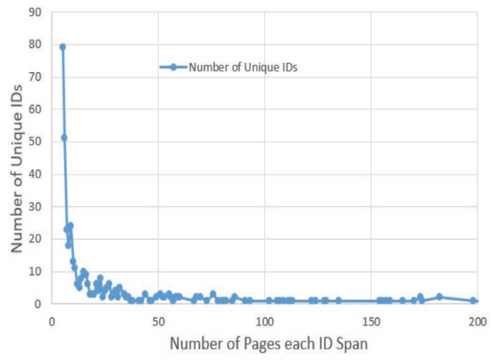


Figure 10 Breakdown of the number of GHS instances per ID.

stage, a pop-up appears claiming that the hack was successful and the victim then invited to a “verification” step. During this verification process, some screen is shown to the user, asking to complete one or more tasks, called “offers”. These offers are a dynamically loaded list of several tasks for the victim to complete. Our previous analysis in [4] showed that this list of tasks usually leads to a small set of offers pages, and there is a significant overlap between the offers provided by the different scammers.

In this research, we were able to identify and collect 375 different offers website. Many of the identified offers are subscriptions for services advertising online libraries and video/music streaming. All of these domains use very similar site templates and similar sign-up forms. Moreover, their second-level domain names tend to be created following similar patterns; the books sites contain “book” in the domain name and the streaming sites contain "music"/"play" in the domain name. In general, these sites claim to have a free trial period, but a valid credit card must be provided to enroll. It is very doubtful that any of these sites would provide any service at all. Other users reported their inability to get through these sites customer services [35, 51]. As an example, subscription scam is the sixth-highest scam causing money loss in Canada with $2.9 M in 2015 [21].

Other websites ask the victim to download and install executable files. Unsurprisingly, these executable files are flagged as malware by sites such as virus total [41].

Finally, some of the offers are sites that promise free vouchers, gift cards, and free products in exchange for completing surveys. These websites are part of the survey scam which has recently been explored in prior work [10, 19]. For example, prize scams are the third-highest scam that caused money loss in Canada with $6.5 M in 2015 [21].

4.4 Domains Analysis

In this section, we analyze the domains names of the servers hosting generators and offers. We first present the most abused TLDs in the collected domains. We then compare these domains with popular Blacklists as well as Google safe browsing [38] and virus total [41].

4.4.1 Most Abused TLDs

For the generator domains, the most common TLD in our database is *.com* which appeared in 39.37% of the final-landing scam domain names. The second most common TLD is *.club* appearing in 7.96% of the domains names. *.xyz*, *.online* and *.net* each represent more than 4% of the domain names. Table 3 shows the details. In the case of the offer domains, we find that *.com* and *.net* are by far the most common TLDs, used in 71.2% and 17.87% of the time respectively.

Table 3 Most common top-level domains (TLDs) for the final URLs of GHS instances

TLD	%	Num Domains	TLD	%	Num Domains
com	39.37	2,284	org	3.86	224
club	7.96	462	us	3.05	177
xyz	5.84	339	top	2.98	173
online	4.71	273	win	2.88	167
net	4.58	266	pro	2.64	153

4.4.2 Overlap with Blacklists

We checked if the domains of the final URLs of our scam domains are flagged by some of the popular public blacklists (PBL), including malware-domains [26], SANS [30], abuse.ch [31], Malc0de database [28], malware-domainlist [29], and hpHosts [27]. For each domain, we check if blacklisted and if so, when it was first added to the list. Only 110 of the 5,930 domains hosting GHS instances are blacklisted by at least one PBL (1.85% of the domains).

Moreover, we have scanned our domains against Google Safe Browsing [38] and virus total [41]. We found that 336 (5.66%) of the domains are flagged by virus total, and 8 (0.13%) by Google Safe Browsing. Cumulatively, we have only 398 (6.71%) domains identified as a scam.

The PBLs fare better when it comes to the offer domains. Although 3 of the PBLs do not flag any of the offer domains, hpHosts [27] flagged 189 (50.4%) of the offers domains. However, these domains were black-listed long after their registration date. On average, they were black-listed 918 days after the domain registration, and the earliest black-listing time was 34 days. However, we should note that we do not know when the domain started to host scams actively.

Similar to the generators domains, we scanned the offers domains using Safe Browsing [38] and virus total [41]. We found that 96 (25.6%) of the domains are flagged by virus total, and 5 (1.3%) by Google Safe Browsing. Cumulatively, we have only 233 (62.1%) domains identified as a scam.

These results suggest that the current PBLs are ineffective against GHS attacks, as they are against other scams such as the Technical Support Scam [49]. A system such as ours is much more effective at protecting end-users.

5 Executable Files and Modified APKs

In some cases, the generator is bypassed and the victim is directly presented with an alternate way to supposedly hack their favorite game. In this case, the attacker either provides an executable to download: either a modified version of the wanted game (an APK executable for Android), or an executable MS Windows file. These executable files are also sometimes provided as a payload by generators. We have collected 59 Windows executable files and 325 unique modified Android games APKs.

We scanned the 59 executable files using virus total [41]. Virus total scans any file or URL with over 70 antivirus scanners and URL/domain blacklisting services. All of the 59 files were flagged by at least two anti-virus scanners, and 54 (91.5%) of the files were flagged by at least 5 scanners. Moreover, many of these files were flagged by avast [32], avg [33], bitDefender [36], and kaspersky [39]. Traces for Trojan, Malware, Bitcoin miner, Coin miner, Dropper, and Adware were reported.

We were able to scan 40 of the 325 APKs using the free API of virus total[3]. 26 (65%) of the APKs were flagged at least once, and 19 (47.5%) of the files were flagged by at least 5 scanners. Traces for Trojan, Coin miner, Coin hive, Bitcoin miner, Malware, Adware, and Dropper were reported. Furthermore, we randomly selected 10 of the APKs for which we could find the original game on Google play. We were not able to run 3 of these APKs. Another one turned out to not be the game at all, but simply an instance of a GHS wrapped into an app. 5 of the remaining APKs seemed to be working instance of the original game, in which the identifier used to displays advertisement in the game had been modified, probably providing income to the hacker instead of to the genuine game developer. The last APK downloads and installs another APK, which is another game store.

6 Bitly Links Analysis

In our corpus, 2,708 of the GHS URLs were shortened using Bitly before being published. As pointed out in [22], Bitly provides a public API that can be used to collect metrics related to its URLs. In this section, we utilize this Bitly API to gain some insights about how successful GHS attacks are. We look at the lifespan of the links and at the number of clicks each link received. Then, we look at click through over time. Finally, we analyze the traffic, to find out the most common country of origin and referrer for the victims.

6.1 Click Through Analysis

Looking at the click-through activity seen on the GHS links, we see that 2,694 (99.48%) of the URLs received at least two clicks and 30% of the URLs receive at least 630 clicks. On average, we see an astonishing average of 2,274.68 clicks per link, accumulating a total of 6,127,995 clicks in our database of links. Our click-count analysis is presented in Figure 11.

[3]The maximum file size allowed by the free API of virus total is 32 MB, which is smaller than most of our APKs.

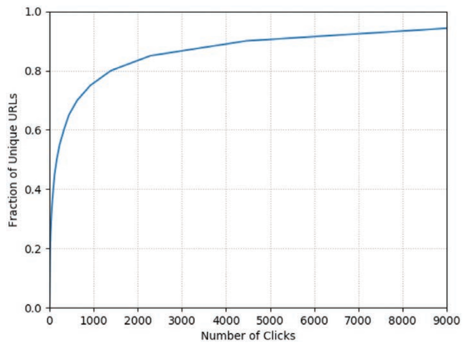


Figure 11 GHS click through analysis.

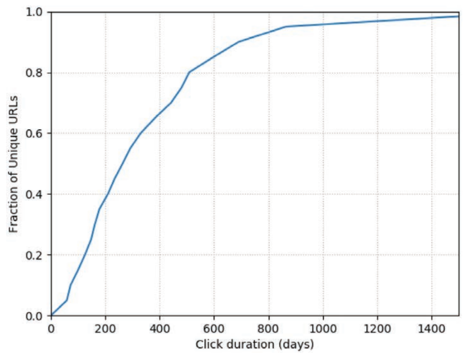


Figure 12 GHS click duration analysis.

This shows that the scam attracts a large number of people. If we assume that in our database, the links that go through Bitly are reasonably representative of the other links, it suggests that our 65,905 URLs have generated around 150 million clicks. What is more, our method is certainly not exhaustive, and we are probably missing many GHS URLs, so the number of people clicking through the scam is perhaps even higher still.

As for the link click duration, our analysis shows that the links have a relatively long lifespan, and 40% of the links register clicks over a year or more. Moreover, around 10% of the URLs registered clicks over two years. This suggests that the links remain effective for a long time. Click-through-duration analysis is presented in Figure 12.

6.2 Monthly URL Clicks and Creation Analysis

In this analysis, we look at when the scam was most active. We also show that the URLs discovered in our previous analysis are still not blocked, and still

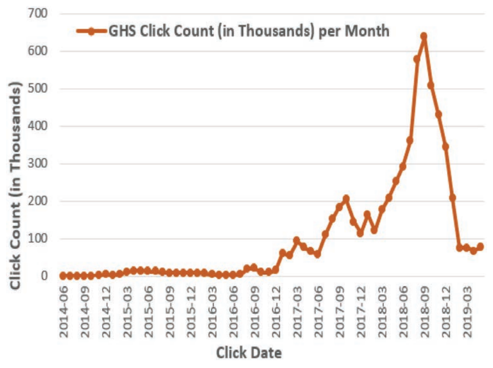


Figure 13 GHS clicks count per month.

have a high click rate. Perhaps the most telling metrics is shown Figure 13. In this figure, we show the number of clicks received each month by our Bitly URLs since 2014. The line represents clicks count for the shortened URLs we collected as of May-2019. As shown in the figure, the number of clicks was on the rise, with a very sharp increase throughout 2018, reaching its maximum with more than 637 k clicks in September-2018. We do not know the cause of this peak.

The analysis also shows that the URLs collected for our previous analysis [4] are still active, and no URL was blocked. Moreover, over the last eight months following the first analysis, these URLs got around 1.35 M clicks, 626.5 new clicks per URL on average. These results indicate that this scam is very active, and the number of victims is growing. Besides, this analysis suggests that no real actions are taken to stop this type of fraud. Awareness of GHS must be increased, and some suitable protection mechanisms are needed to stop it.

6.3 Country and HTTP Referrer Clicks

If we look at the countries from which the links have been clicked, we find a total of 245 countries, out of 254 possible country codes [1]. It shows that GHS attracts victims from nearly everywhere on earth. In terms of volume, victims in the US and India have generated the largest number of clicks, with 21.3% and 10.7% respectively. Figure 14 shows a break down of the number of clicks per country. If we consider the world population [42] to normalize the number of clicks per country per citizen; Singapore, Malaysia, and New Zealand have the highest number of clicks. To mitigate any bias

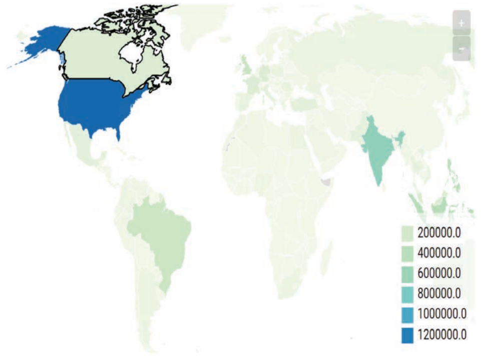


Figure 14 GHS clicks per country.

Table 4 Referrers and countries with the highest number of clicks (countries analysis is normalized using the clicks-population ratio)

	Top Countries	Top Referrals	
Rank	Countries	Referrer	% Clicks
1	Singapore	direct	71.4
2	Malaysia	piktochart.com	4.6
3	New Zealand	jeuxvideo.com	3.33
4	Greece	google.com	1.48
5	United Kingdom	fliphtml5.com	1.43
6	United States	change.org	1.32
7	Canada	megatut.com	1.2
8	Australia	kabam.com	1
9	Philippines	t-adbar1.com	0.8
10	Romania	flasygames.com	0.8

that may occur from countries with low hits and a low number of citizens, we ignored any country with a population less than 1 M or got less than 20,000 hits. Please refer to Table 4 for the top ten countries.

As for the URLs referrers, we find that GHS URLs were accessed from 1,532 domains. "Direct" access is the most common source with 71.4%. Direct access includes sources like email clients, instant messages, and dedicated applications [4, 22]. Table 4 gives the breakdown of the top ten origin countries and referrers.

7 Related Work

Although recent researches have provided important insights into different types of scam, to our knowledge, GHS is not fully studied yet. The works most closely related to ours are studies about the so-called "Technical Support Scam" (TSS) as well as studies on online survey scams. In TSS, scammers combine online scam and telephone fraud activities to convince their victims that their machines are infected with malware, and offer a fake technical support service. Miramirkhani et al. presented the first systematic TSS study [24], which was continued and improved by Srinivasan et al. [49]. TSS was also studied in [47, 48], and several reports were published about the scam [44].

In the case of survey scam, victims are tricked into providing sensitive information and installing malware and unwanted programs. Usually, this happens while asking the victims to complete some surveys in exchange of some expected award. A variety of awards are advertised, for example, free software's, gifts, as well as gift cards for different stores such as Amazon and Costco [10, 19]. Several security companies have published reports about survey scams [9, 13, 25, 45, 46, 54].

Another type of scam is the Internal Revenue Service (IRS) scam. In the IRS scam, the scammer impersonates an IRS official to trick the victim into sending money [5, 11].

Furthermore, there is the "romance scam", which can cause considerable emotional damage in addition to financial losses. In this case, a false relationship is initiated by the scammer using chat services, social media, and dating sites. The victim is then asked to provide some financial support to the scammer. This scam and its serious emotional consequences has been studied in [6, 7, 20, 52, 53].

Finally, Many researchers have studied malware detection in mobile applications. The majority of these researches focused on extracting a set of features from the application APKs to be used in a classifier. An example of the extracted features are the application permissions, the permissions used within the app, and the API calls in the application code. Drebin [2] uses SVC classifier to distinguish between benign and malicious APKs based on a set of features extracted from AndroidManifest.xml and disassembled codes. In [14] Idrees and Rajarajan proposed a detection method that utilizes classes.dex and manifest file to extract a feature set from the permissions and API calls. Similar approach was followed by Yang and Wen in [23]. Other researchers focused entirely on APKs permissions or code behaviour to decide if the application is harmless or not. The reader can refer to [3, 15, 50, 55] for more information. To our knowledge, no one have studied the modified version of an actual game APK.

8 Limitations and Future Work

One of the main limitations of our study is that we only look for GHS instances based on the ones we have already found. Thus, some of our current results may be biased by the type of GHS instances we are looking for, and a more systematic search would shed new lights to the situation (for example, other template providers might come to light).

Another limitation is that we are studying the URLs distributed using social media and blogs. Bitly analysis suggests that 71% of the generated traffic is direct through emails and instant messages. This, in turn, suggests that we are missing a big source of URLs distribution. On the other hand, this 71% generated traffic comes from URLs we collected using websites crawling. Thus, although the web traffic is only 29%, it helps in discovering traffic from bigger sources of URLs distribution.

Finally, we would like to study in more details the offer side of the scam and pinpoint the template providers who provide these offers. The difficulty here is to be able to gain insight without contributing financially to the scam. We are in discussion with our ethics board to find the best solution to achieve this.

9 Conclusions

In this paper, we investigated what we call the "Game Hack" Scam (GHS). we formulated GHS-related search queries, and used multiple search engines to collect data about the websites to which GHS victims are directed when they search online for various game hacks and tricks. We looked at the pages returned directly by the search engines, as well as the pages linked from these pages. We also investigated the modified APKs, and the executable files collected when searching online for the game hack.

Our data collection spanned a year; in that time, we uncovered 65,905 different GHS URLs, mapped onto over 5,900 unique domains. We were able to link several attacks to attackers and found that they routinely target a vast array of games. Furthermore, we find that GHS instances are on the rise, and so is the number of victims. Our low-end estimation is that these attacks have been clicked at least 150 million times in the last five years. Additionally, in keeping with similar large-scale scam studies, we find that the current public blacklists are inadequate and suggest that our method is more effective at detecting these attacks.

Finally, we found that more than 90% of the GHS related executable files are flagged by at least five antivirus scanners in virus total. For the modified Android games APKs, 47.5% are flagged by at least five antivirus scanners in virus total. Furthermore, some of these games are not working, some of them have changed the in-game advertisements, and some of them have changed the game completely.

All the data used in our study is available at http://bit.ly/GHSJWE.

Appendix

Appendix A: Further Testing on Data-set B

As explained in Section 3.1, we have trained our model on Data-set A, a subset of our complete database, where each cluster only contributes one element to the training set. Data-set B contains the other instances, that is, the complete database minus data-set A. We did not used data-set B for our model testing because it was unclear how different that data-set was from the training set. However, to be thorough, We wanted to see how well the trained model would perform on the other instances that were discarded for training, the data-set B. We report here the results of that experiment.

Here, we present a further testing of our final classification model, which is a two-step model combining the two filters constructed in 3.3 and the SVC text-based classifier of 3.2. In this experiment, we used our classification model to classify instances that have not been observed in the training phase. For this purpose, we used data-set B we prepared in Section 3.1.

Table 5 presents the performance of our model with and without the two filters constructed in 3.3. We use the same SVC text-based classifier described in Section 3.2, trained on data-set A.

Table 5 SVC text classifier with and without the filtering step. This test is conducted on our testing data set B

	Filtered Clean	Classified Clean	Filtered GHS	Classified GHS	%
	SVC Text Classifier				
clean	NA	7,887	NA	52	0.65% FPR
gen	NA	68	NA	31,027	99.78% TPR
	Filters+SVC Text Classifier				
clean	2,815	5,111	0	13	0.16% FPR
gen	1	13	21,462	9,619	99.95% TPR

Without using the filters, the SVC classifier already performs very well, with a TPR of 99.78% and a FPR of 0.65%. These results are further improved when the filters are added as a preprocessing steps: the filter alone are able to detect 69% of the generator pages and 35% of the clean pages, and the over result is improved with a TPR reaching 99.95% and a FPR down to 0.16%.

As a conclusion, our classification model achieved a very high accuracy on data-set B. These results proves that a single instance from each true positive cluster is sufficient to detect all the duplicates instance belongs to the same cluster.

References

[1] List: The two-letter country code/country abbreviation. bit.ly/2ROvg8N, 2018.

[2] Daniel Arp, Spreitzenbarth Michael, Hubner Malte, Gascon Hugo, Rieck Konrad, and Siemens C. E. R. T. Drebin: Effective and explainable detection of android malware in your pocket. *Ndss*, 14:23–26, 2014.

[3] A. M. Aswini and P. Vinod. Droid permission miner: Mining prominent permissions for android malware analysis. In *The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014)*, pages 81–86, Feb. 2014.

[4] Emad Badawi, Guy-Vincent Jourdan, Gregor Bochmann, Iosif-Viorel Onut, and Jason Flood. The "game hack" scam. In *International Conference on Web Engineering*, pages 280–295. Springer, 2019.

[5] Morvareed Bidgoli and Jens Grossklags. "hello. this is the irs calling.": A case study on scams, extortion, impersonation, and phone spoofing. In *Electronic Crime Research (eCrime), 2017 APWG Symposium on*, pages 57–69. IEEE, 2017.

[6] Tom Buchanan and Monica T. Whitty. The online dating romance scam: causes and consequences of victimhood. *Psychology, Crime & Law*, 20(3):261–283, 2014.

[7] Carolyn Budd and Jessica Anderson. *Consumer Fraud in Australasia: Results of the Australasian Consumer Fraud Taskforce Online Australia Surveys 2008 and 2009.* Australian Institute of Criminology, 2011.

[8] Compute Canada. Research portal home – compute canada. https://www.computecanada.ca/research-portal/, 2019.

[9] Oscar Celestino. Survey scams aimed at social networking netizens. bit.ly/2Jr9UXK, 2012.

[10] Jason W. Clark and Damon McCoy. There are no free ipads: An analysis of survey scams as a business. In *Presented as part of the 6th USENIX Workshop on Large-Scale Exploits and Emergent Threats*, Washington, D.C., 2013. USENIX.

[11] Cassandra Cross, Kelly Richards, and Russell G. Smith. The reporting experiences and support needs of victims of online fraud. *Trends and Issues in Crime and Criminal Justice*, 518:1–14, 2016.

[12] Qian Cui, Guy-Vincent Jourdan, Gregor V. Bochmann, Russell Couturier, and Iosif-Viorel Onut. Tracking phishing attacks over time. In *International World Wide Web Conferences Steering Committee*, pages 667–676, 2017.

[13] Nishant Doshi. Survey scammers moving to pinterest. symc.ly/2SwIfbZ, 2012.

[14] F. Idrees and M. Rajarajan. Investigating the android intents and permissions for malware detection. In *2014 IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 354–358, Oct. 2014.

[15] L. Jing. Mobile internet malicious application detection method based on support vector machine. In *2017 International Conference on Smart Grid and Electrical Automation (ICSGEA)*, pages 260–263, May 2017.

[16] Daniel Jurafsky and James H. Martin. Markov assumption. stanford.io/29zsjAy, 2014.

[17] Daniel Kaszor. How free-to-play games make money. bit.ly/2QgHpPc, 2012.

[18] Kate Kershner. How do free-to-play games make money? bit.ly/2yN3huU, 2018.

[19] Amin Kharraz, William Robertson, and Engin Kirda. Surveylance: Automatically detecting online survey scams. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 70–86. IEEE, 2018.

[20] Christian Kopp, James Sillitoe, Iqbal Gondal, and Robert Layton. THE ONLINE ROMANCE SCAM: A COMPLEX TWO-LAYER SCAM. *Journal of Psychological & Educational Research*, 24(2):144–161, 2016.

[21] Mike Laanela. Canada's top 10 scams earned crooks $1.2 b last year, say bbb | cbc news. bit.ly/2P6r2IC, 2016.

[22] Sophie Le Page, Guy-Vincent Jourdan, Gregor V. Bochmann, Jason Flood, and Iosif-Viorel Onut. Using url shorteners to compare phishing

and malware attacks. In *In APWG Symposium on Electronic Crime Research (eCrime)*, pages 1–13. IEEE, 2018.

[23] Manzhi Yang and QiaoYan Wen. Detecting android malware with intensive feature engineering. In *2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pages 157–161, Aug. 2016.

[24] Najmeh Miramirkhani, Oleksii Starov, and Nick Nikiforakis. Dial one for scam: A large-scale analysis of technical support scams. *arXiv preprint arXiv:1607.06891*, 2016.

[25] Satnam Narang. Instascam: Instagram for pc leads to survey scam. symc.ly/2ESLmbC, 2013.

[26] Online. Dns-bh malware domains. http://www.malwaredomains.com/, 2017.

[27] Online. hphosts online, simple, searchable & free. https://hosts-file.net/, 2017.

[28] Online. Malcode database. http://malc0de.com/database/, 2017.

[29] Online. Mdl: Malware domain list. https://www.malwaredomainlist.com/, 2017.

[30] Online. Sans: Suspicious domains. bit.ly/2FNCzHv, 2017.

[31] Online. The swiss security blog. bit.ly/2EE7HK1, 2017.

[32] Online. Avast download free antivirus for pc, mac and android. https://bit.ly/2XaviWv, 2018.

[33] Online. Avg 2019 free antivirus, vpn and tuneup for all your devices. https://bit.ly/2RKgsVE, 2018.

[34] Online. Beautifulsoup. https://pypi.org/project/beautifulsoup4/, 2018.

[35] Online. Beware of music g8 at musicg8.com – it is a fraudulent website. http://bit.ly/2XDi4pG, 2018.

[36] Online. Bitdefender antivirus – discover the complete security solution. https://bit.ly/2NmsXs2, 2018.

[37] Online. Chromedriver – webdriver for chrome. bit.ly/2CMwVBG, 2018.

[38] Online. Google safe browsing api. https://goo.gl/4yAFyQ, 2018.

[39] Online. Kaspersky lab antivirus protection and internet security software. https://bit.ly/3038R7H, 2018.

[40] Online. Selenium with python, selenium python bindings. bit.ly/2LNldJn, 2018.

[41] Online. Virustotal. https://www.virustotal.com/, 2018.

[42] Online. Country codes, phone codes, dialing codes, telephone codes, iso country codes. https://countrycode.org/, 2019.

[43] Online. Google trends. https://trends.google.com/trends/?geo=US, 2019.

[44] Orla. Technical support phone scam. symc.ly/2OdDyR3, 2010.

[45] Stelian Pilici. How to remove "2017 annual visitor survey" adware (virus help guide). bit.ly/2yGeLjU, 2017.

[46] Stelian Pilici. How to remove "chrome opinion survey" pop-ups (survey scam). bit.ly/2ziF5A6, 2018.

[47] Sampsa Rauti and Ville Leppänen. "you have a potential hacker's infection": A study on technical support scams. In *2017 IEEE International Conference on Computer and Information Technology (CIT)*, pages 197–203. IEEE, 2017.

[48] Merve Sahin, Marc Relieu, and Aurélien Francillon. Using chatbots against voice spam: Analyzing lenny's effectiveness. In *Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017)*, pages 319–337, Santa Clara, CA, 2017. USENIX Association.

[49] Bharat Srinivasan, Athanasios Kountouras, Najmeh Miramirkhani, Monjur Alam, Nick Nikiforakis, Manos Antonakakis, and Mustaque Ahamad. Exposing search and advertisement abuse tactics and infrastructure of technical support scammers. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 319–328. International World Wide Web Conferences Steering Committee, 2018.

[50] P. Tiwari, G. Tere, and P. Singh. Malware detection in android application by rigorous analysis of decompiled source code. In *2016 International Conference on Computing Communication Control and Automation (ICCUBEA)*, pages 1–6, Aug. 2016.

[51] Vanessa. Detailed information about 888.980.9787 or 888.980.9787 phone number in free number 888 free 8xx us. bit.ly/2RMmbxv, 2018.

[52] Monica T. Whitty. Anatomy of the online dating romance scam. *Security Journal*, 28(4):443–455, 2015.

[53] Monica T. Whitty and Tom Buchanan. The online romance scam: A serious cybercrime. *CyberPsychology, Behavior, and Social Networking*, 15(3):181–183, 2012.

[54] Candid Wueest. Fast-flux facebook application scams. symc.ly/2ADviG F, 2011.

[55] Zhongyuan Qin, Yuqing Xu, Yuxing Di, Qunfang Zhang, and Jie Huang. Android malware detection based on permission and behavior analysis. In *International Conference on Cyberspace Technology (CCT 2014)*, pages 1–4, Nov. 2014.

Biographies

Emad Badawi earned his Bachelor's Degree in Computer Systems Engineering from the Arab American University in Jenin (AAUJ). After that, he moved to the United Arab Emirates in 2015 and joined the Master's program in Computer Engineering at the American University of Sharjah (AUS), from which he obtained a Master's Degree in Computer Engineering in 2017. His thesis research was about parallel implementations for Finite State Machines mutants elimination algorithms, based on OpenMP, MPI, and CUDA.

Currently, he is continuing his higher studies as a Ph.D. candidate at the Department of Electrical and Computer Engineering at the University of Ottawa, which he joined in September 2017. His current research is about software security, in particular, cyber-attacks detection.

Guy-Vincent Jourdan is a full professor at the Faculty of Engineering of the University of Ottawa. He joined the School of Electrical Engineering and Computer Science as an associate professor in June 2004, after 7 years in the private sector as C.T.O. and then C.E.O. of Ottawa-based Decision Academic Graphics. He received his PhD from l'université de Rennes/INRIA in France in 1995 in the area of distributed systems analysis. His research interests include distributed systems modeling and analysis, software security, cybercrime detection and prevention.

Gregor Bochmann was a full professor at the University of Ottawa from January 1998 to June 2016. Before, from 1972 to 1997, he was professor at the Université de Montréal. When he left, he was honored with the title of professeur émérite at the University of Ottawa. He received his PhD from McGill University in Canada in 1971. His research group works on methods for the development of communication protocols and distributed systems, on the use of formal methods for the analysis, design and implementation of communication protocols, and software development in general. Practical applications of these methods are pursued in relation with network protocols (e.g. Internet and optical networks), Web Services, workflow management, peer-to-peer systems, and analysis of Rich Internet Applications. In the past, they have also done much work in the areas of quality of service management for distributed multimedia applications and development of test suites with known fault coverage, diagnostics and testability.

Iosif-Viorel Onut is currently affiliated with IBM Security Systems and Center for Advanced Studies (CAS) at IBM. He also is Adjunct Professor at University of Ottawa. He completed his PhD degree at the Faculty of Computer Science, University of New Brunswick and specializes in topics related to network security, such as simulation, detection, prediction, and visualization of network attacks, and in-depth study of network features for attack detection. Currently, his main research focus is in the area of Web

2.0 application security, compliance and crawling, but also intelligent sensor technologies for context-aware security risk assessment.

Throughout his career, he was part of many Research and Development collaborations with institutions such as: DaimlerChrysler AG Research and Technology center in Berlin, Germany; National Research Council of Canada, Institute for Information Technology, Fredericton, Canada; City of Fredericton, Network Infrastructure, Fredericton, Canada; Q1Labs, Boston, US; and recently CAS-IBM, Ottawa, Canada. He authored more than 40 patents, journals, conference publications and technical reports at prestigious international journals and conferences such as Elsevier Computers and Security, Springer's Lecture Notes in Computer Science and Information Security Conference.