Security Analysis of Top Visited Arabic Web Sites

Abdulrahman Alarifi, Mansour Alsaleh
Computer Research Institute,
King Abdulaziz City for Science and Technology, Riyadh, KSA
{aarifi, maalsaleh}@kacst.edu.sa

AbdulMalik Al-Salman
Computer Science Department,
King Saud University, Riyadh, KSA
salman@ksu.edu.sa

***Abstract*—The richness and effectiveness of client-side vulnerabilities contributed to an accelerated shift toward client-side Web attacks. In order to understand the volume and nature of such malicious Web pages, we perform a detailed analysis of a subset of top visited Web sites using Google Trends. Our study is limited to the Arabic content in the Web and thus only the top Arabic searching terms are considered. To carry out this study, we analyze more than 7,000 distinct domain names by traversing all the visible pages within each domain. To identify different types of suspected phishing and malware pages, we use the API of Sucuri SiteCheck, McAfee SiteAdvisor, Google Safe Browsing, Norton, and AVG website scanners. The study shows the existence of malicious contents across a variety of types of Web pages. The results indicate that a significant number of these sites carry some known malware, are in a blacklisting status, or have some out-of-date software. Throughout our analysis, we characterize the impact of the detected malware families and speculate as to how the reported positive Web servers got infected.**

***Index Terms*—Malware, Malicious links, Web spam, Search engine spam, Web vulnerabilities**

I. Introduction

As the Web is increasingly becoming a crucial medium for industries and businesses such as banking and government, there is a drastic evolution in the for-profit Web-based malware. A typical attack starts with the user visiting a malicious Web page that either tempts the user to execute some driveby downloads or exploits a vulnerability in the Web browsers or a browser plug-in (e.g., Flash, Java and PDF viewers). The exploit could then execute a malicious binary to turn the host into a bot controlled by the adversary. Various types of monetization vectors can be utilized by adversaries such as spamming, clickfraud, dropping, browser hijacking, information stealing and fake software. Prior studies show that driveby downloads that target browser and browser plug-in vulnerabilities to install malware now represent the largest threat to end users [10]. The success of driveby downloads led to the spread of exploit kits that facilitate compromising the visiting hosts of a malicious page by providing a set of browser exploits [4].

It is important to note that most malicious Web servers are not owned by adversaries but are rather exploited by adversaries who either: (i) scan the Web servers over the Internet for some particular vulnerabilities that are then exploited to host some malicious pages [6]; (ii) embed their malicious code into some third-party Web controls (e.g., ad rotator, calender, and hit counters) that are utilized by some Web developers; or (iii) use online advertisements for propagating malware (e.g., scamming and click frauds) [8]. While it is expected that developers should only use JavaScript from trustworthy vendors, not just that this is not usually the case, but it is also possible that some trustworthy JavaScript libraries get compromised [9].

For simplicity, adversaries usually link or redirect their compromised pages to a fewer number of malicious pages. Also, given that adversaries usually utilize some known Web exploit toolkits and use automation to accelerate their work process, there are some similarities between malicious pages on different sites [5].

For detecting malicious Web pages, low- and high-interaction honeyclients (e.g., Capture-HPC [11]) can be used to detect malicious content [3, 12]. To focus the examination of Web pages in the wild for malicious content, a number of heuristics can guide the search so that only the pages that are more likely to contain malicious content are analyzed [2]. New approaches include leveraging a seed of known, malicious Web pages to extract the similarities that these pages share. These similarities are then leveraged using search engines to find similar malicious characteristics in other pages [5].

Given the recent spike in the malicious Web activity in the middle east region [7] and that literature lacks any quantitative study pertaining the status of malicious Arabic Web content, we perform an analysis study of a subset of top visited Arabic Web sites. In this work, we analyze more than 7,000 distinct domain names by traversing all the visible pages within each domain. We use the API of Sucuri SiteCheck, McAfee SiteAdvisor, Google Safe Browsing, Norton, and Sophos (using Yandex service) website scanners to detect malicious Web pages in the inspected domains.

Our results show that there are a variety of malicious Web pages across many inspected domains of Arabic content. In addition to the existence of some known malicious binaries, many inspected Web site were found blacklisted by AV vendors. We also characterize the impact of the detected malware families and speculate as to how the reported positive Web servers got infected.

Organization. Section II describes the collection process and the malware classification methodology. Section III presents the results and analysis of the examined Web pages. Section IV provides further discussion and concluding remarks.

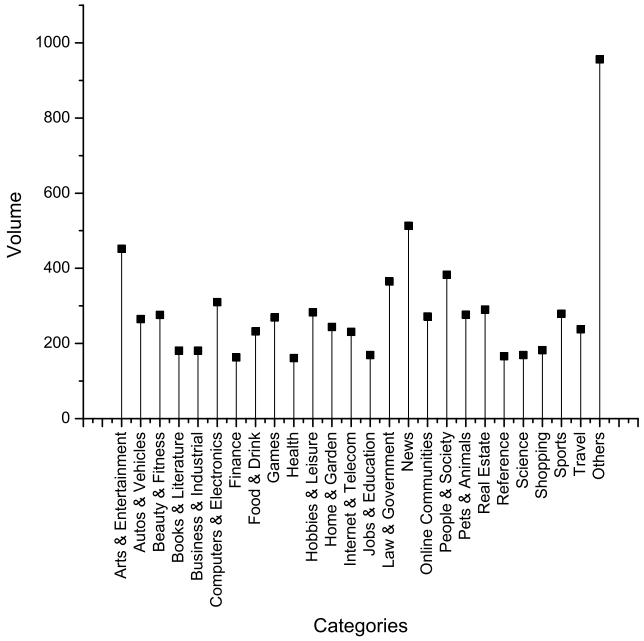

Fig. 1. The number of URLs per category.

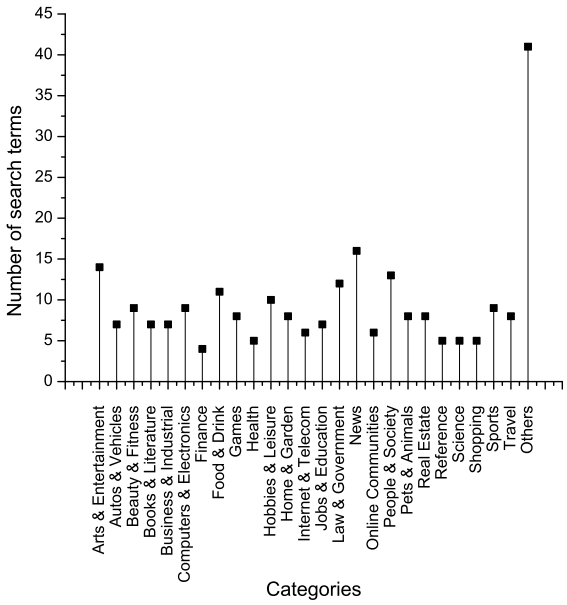

Fig. 2. The number of search terms per category.

II. Setup and Methodology

For locating and examining malicious Web pages in the Arabic domains, a two-step process is followed. First, we collect the top Arabic search terms from Google Trends for the period between January 2004 and October 2012. Google Trends classifies the search terms into 16 categories[1]. We have also created a new category (called "others") of some random Arabic words, to not be limited to Google Trends top search terms. Figure 1 shows the distribution of the Arabic search terms among the 16 categories. Figure 2 shows the number of search terms per category. The collected search terms (more than 7,507 distinct domain names) are then used to query Google search engine and gather the top 50 pages of search results (only the URLs of the results are stored).

In the second phase, every URL is scanned against six website scanners of some known AV vendors: (1) Sucuri SiteCheck; (2) McAfee SiteAdvisor; (3) Google Safe Browsing; (4) Norton; and (5) Sophos (using Yandex ranking). The website scanners examine every visible page in the whole domain of an URL. The scanning results of every URL is then stored into a MySQL database for both the reported positive and negative samples. Given the low number of detected malicious pages using (3), we omit the detection results of Google Safe Browsing. Figure 3 shows our process pipeline for locating and examining Web pages in the Arabic domains.

[1]Category refers to verticals; i.e., a classification of industries or markets [1]. The 16 categories are: arts and entertainment, autos and vehicles, beauty and fitness, books and literature, business and industrial, computers and electronics, finance, food and drink, games, health, hobbies and leisure, home and garden, Internet and telecom, jobs and education, law and government, news, online communities, people and society, pets and animals, real estate, reference, science, shopping, sports, and travel.

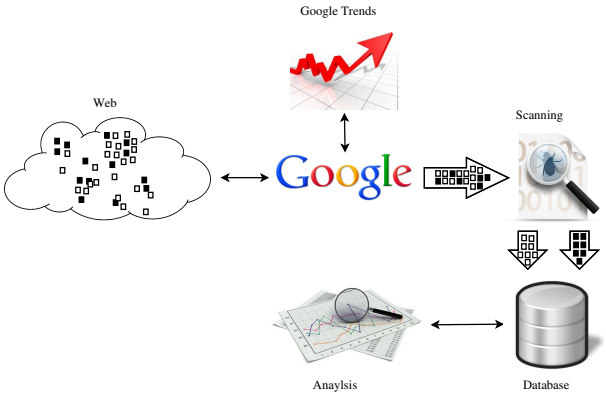

Fig. 3. The experiment process flow.

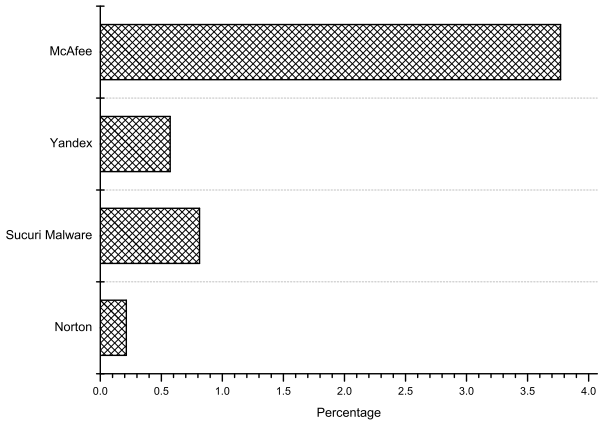

Fig. 4. The detection percentage of reported positive URLs for each Web scanner.

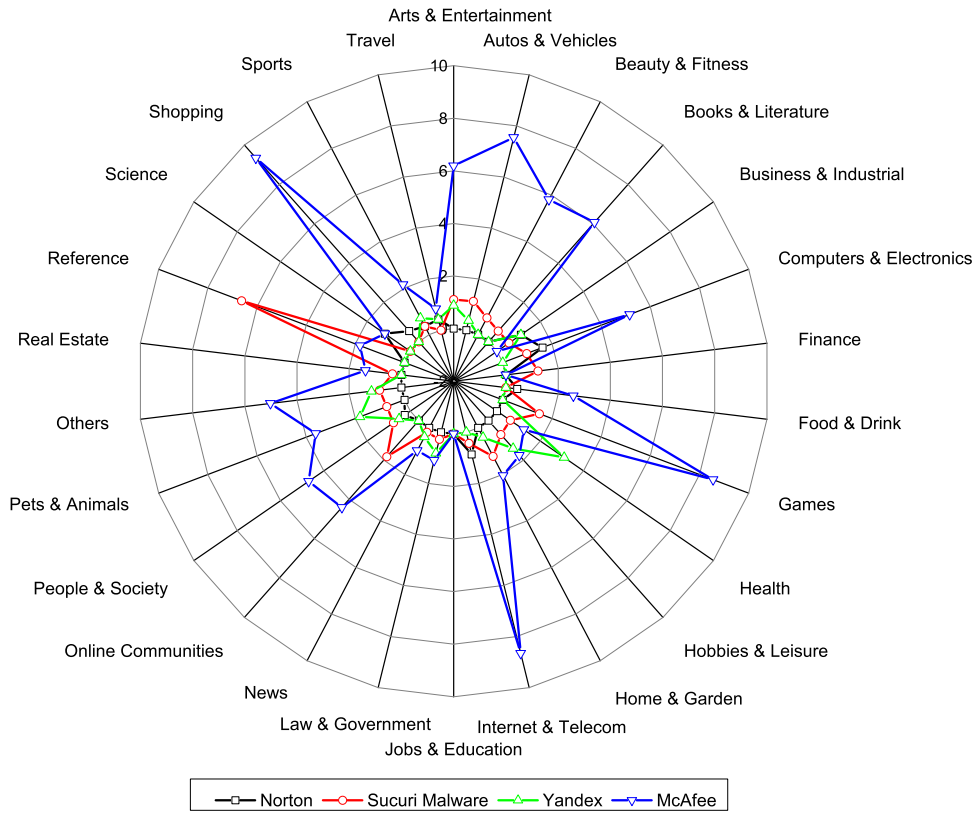

Fig. 5. The distribution of reported positive URLs among Google Trends categories.

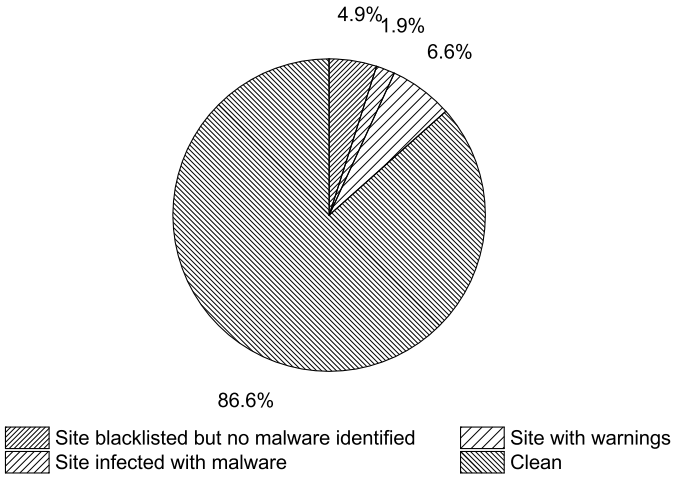

Fig. 6. Classifications of the flagged Web links.

III. Results and Analysis

As discussed in Section II, the second phase of the process is to examine each Web site using different Web scanners. In this section, we analyse the detection rate of each Web scanner and correlate between them.

Figure 4 shows the detection percentage of reported positive URLs for each Web scanner (note that the URL is classified as malicious if any other page within the URL domain is classified as malicious). We note that McAfee SiteAdvisor scored the highest detection rate among the other Web scanners.

To understand the distribution of reported positive URLs among Google Trends categories, in Figure 5, we show the distribution in a radar chart. Note that shopping, games, and Internet and telecom categories have more than 8% of blacklisted Web pages, perhaps because these categories are expected to represent a larger sector of users (e.g., relatively young audience) and thus are targeted by adversaries.

Figure 6 divides the flagged Web links into one of three classes: (i) sites that carry some known malware; (ii) sites that are in a blacklisting status; (iii) sites that have some out-of-date software, and (iv) sites that have no known malicious content. In Figure 7, we see that news, shopping, and travel categories score the highest for classes (i), (ii), and (iii), respectively. We note here that class (i) is directly proportional to class (iii), as once an unpatched version of a vulnerable software exists in a host, the host is vulnerable to a variety of malware that can be installed. Also, we note that class (ii) in Figure 6 is similar to the number of reported positive by Norton in the radar chart in Figure 5, which shows the type of flagged URLs by Norton Web scanner.

Figure 8 shows the percentage of clean URLs (i.e., sites that have no known malicious content) for every category. While job and education, home and garden, and health are the highest three categories (i.e., most clean), reference, shopping, and

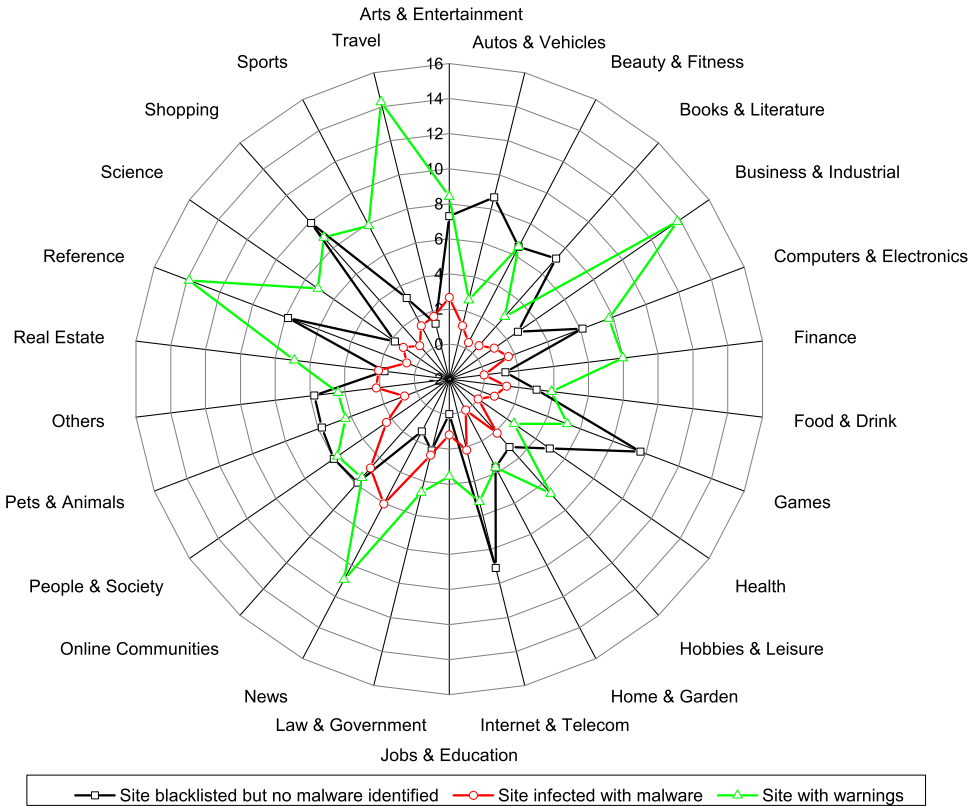

Fig. 7. Classifications of the flagged Web links among the different categories.

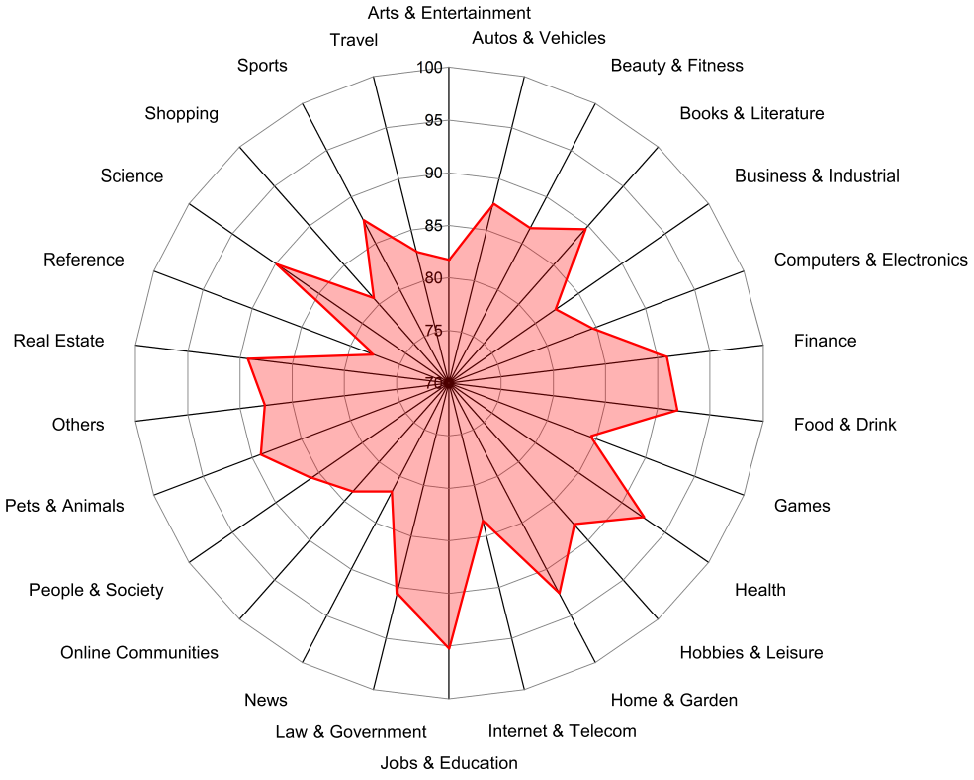

Fig. 8. The percentage of clean URL for every category.

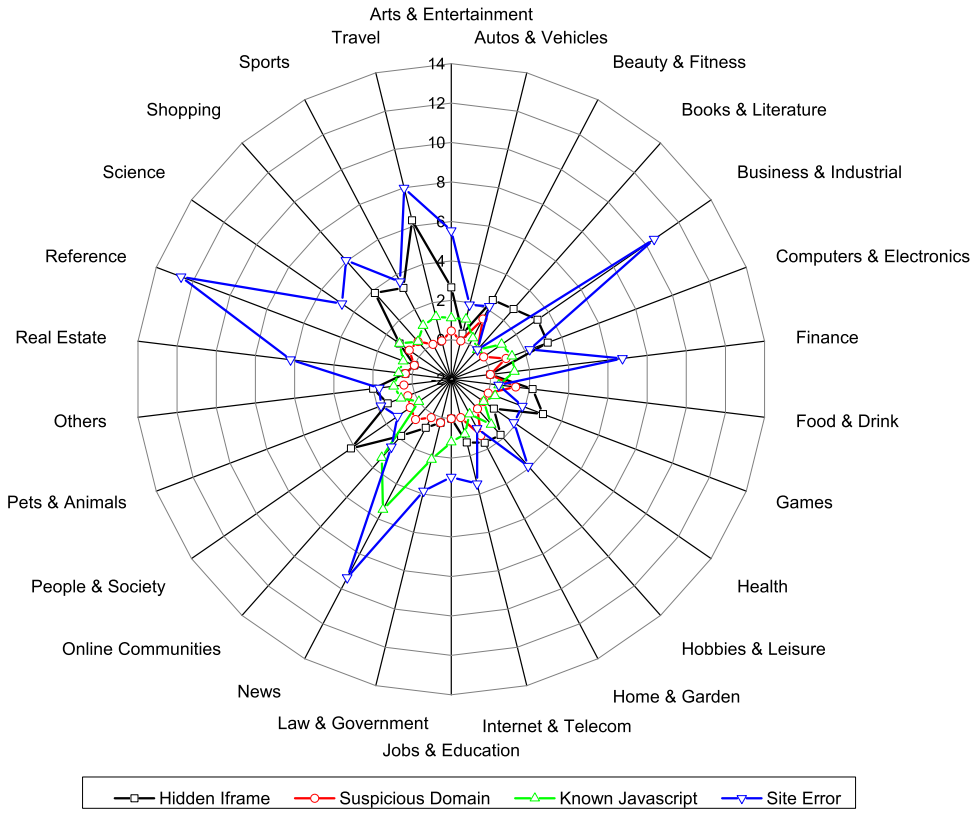

Fig. 9. The distribution among the different categories (see Figure 10).

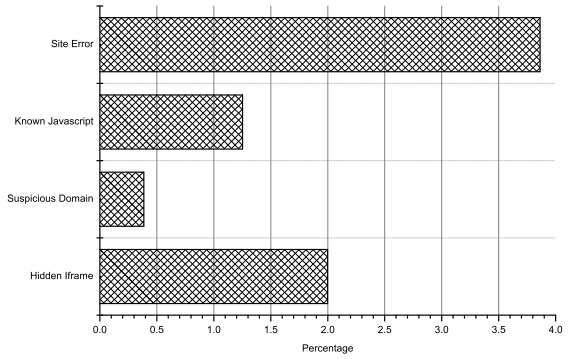

Fig. 10. Classifying URLs according to 4 classes: site error, known malicious JavaScript, suspicious domain, and hidden iframe.

news are the lowest three categories.

In Figure 10 shows different criteria for classifying URLs: (i) hidden iframe is founded; (ii) known malicious JavaScript is found; (iii) the domain is found suspicious; (iv) some errors are found in the site. One observation is that the sites with known malicious JavaScript represent 66% of the sites that carry some known malware (see Figure 7). In Figure 9, we show the distribution of the above criteria among the different categories.

IV. Conclusion

The fact that Google removes malicious Web sites from top search results explains why Google Safe Browsing reports only few malicious links in our datasets. However, this preliminary study shows that Google search engine removes only a portion of reported malicious Web sites by the Web scanners of other AV vendors. Also, we find out that the distribution of the blacklisted sites and the sites with malware vary according to the subject of the Web site content. We emphasize that the study in this paper is preliminary as it only gives a snapshot of the current status of the Arabic content in the Web, and that further work in this area is required.

Avenues for future work include: (i) studying a larger sample space using different methods for gathering search terms; (ii) repeating the study using other search engines; and (iii) characterizing the impact of the detected malware families in further details and understanding how the malicious Web servers got infected. Given that Arabic Web spam is another area that lacks any quantitative studies, we also plan to collect a dataset of Arabic Web spam pages, study the used spam techniques, and examine the effectiveness of search engines in filtering out these spam pages.

Acknowledgment

We thank Abdulmajeeds Alsuwayed and Ahmad Alkhalidi, especially for their implementation work related to the dataset gathering.

REFERENCES

[1] Google Trends Categories. Accessed: Oct 2012. http://support.google.com/trends/bin/topic.py?hl=en&topic=19357&parent=15089&ctx=topic.

[2] D. Canali, M. Cova, G. Vigna, and C. Kruegel. Prophiler: A fast filter for the large-scale detection of malicious web pages. In *Proceedings of the 20th international conference on World wide web*, pages 197–206. ACM, 2011.

[3] B. Feinstein and D. Peck. Caffeine monkey: Automated collection, detection and analysis of malicious javascript. *Black Hat USA*, 2007, 2007.

[4] C. Grier, L. Ballard, J. Caballero, N. Chachra, C. Dietrich, K. Levchenko, P. Mavrommatis, D. McCoy, A. Nappa, A. Pitsillidis, et al. Manufacturing compromise: The emergence of exploit-as-a-service. 2012.

[5] L. Invernizzi, P. Comparetti, S. Benvenuti, C. Kruegel, M. Cova, and G. Vigna. Evilseed: A guided approach to finding malicious web pages. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 428–442. IEEE, 2012.

[6] J. John, F. Yu, Y. Xie, M. Abadi, and A. Krishnamurthy. Searching the searchers with searchaudit. In *Usenix Security Symposium*, 2010.

[7] M. Labs. Mcafee threats report: Third quarter 2012. Accessed: Nov 2012. http://www.mcafee.com/au/resources/reports/rp-quarterly-threat-q3-2012.pdf.

[8] Z. Li, K. Zhang, Y. Xie, F. Yu, and X. Wang. Knowing your enemy: understanding and detecting malicious web advertising. In *Proceedings of the 2012 ACM conference on Computer and communications security CCS'12*, pages 674–686. ACM, 2012.

[9] N. Nikiforakis, L. Invernizzi, A. Kapravelos, S. Van Acker, W. Joosen, C. Kruegel, F. Piessens, and G. Vigna. You are what you include: Large-scale evaluation of remote javascript inclusions. In *Proceedings of the ACM Conference on Computer and Communications Security*, 2012.

[10] M. Rajab, L. Ballard, N. Jagpal, P. Mavrommatis, D. Nojiri, N. Provos, and L. Schmidt. Trends in circumventing web-malware detection. *Google, Google Technical Report*, 2011.

[11] C. Seifert and R. Steenson. Capture-HPC. Accessed: Jun 2012. https://projects.honeynet.org/capture-hpc.

[12] Y. Wang, D. Beck, X. Jiang, R. Roussev, C. Verbowski, S. Chen, and S. King. Automated web patrol with strider honeymonkeys. In *Proceedings of the 2006 Network and Distributed System Security Symposium*, pages 35–49, 2006.