GuidedTracker: Track the Victims with Access Logs to Finding Malicious Web Pages

Hongzhou Sha
School of Computer Science
Beijing University of Posts and Telecommunications
Beijing, China 100876
Email: shahongzhou@gmail.com

Qingyun Liu, Zhou Zhou*, Chao Zheng
Institute of Information Engineering
Chinese Academy of Sciences
Beijing, China 100093
*Corresponding Author: Email: zhouzhou@iie.ac.cn

***Abstract*—Malicious web pages have become a malignant tumour for the Internet, which spread malicious code, steal people's private information, and deliver spamming advertisements. And how to distinguish them from the huge number of normal web pages effectively remains a huge challenge in the era of big data. To detect malicious pages, one needs to first collect candidate web pages that are live on the web; then filter massive legitimate pages using fast filters and finally examine the remaining pages using precisely but slow analyzer. However, there are new challenges recently for these conventional techniques, including large scale, imbalance data and the usage of cloaking techniques. To cope with these challenges, the malicious URL detection system should perform more efficiently.**

In this paper, we propose a system, named GuidedTracker, to search for suspicious malicious pages. GuidedTracker starts from the seed set which includes known malicious pages. Then, it automatically figures out those victims based on the seed set and the visit relation database. Finally, the access records of these victims are used to identify other malicious pages. In this way, GuidedTracker increase the percentages of malicious URLs in the input URL stream submitted to the precisely analyzer. To our best knowledge, GuidedTracker is the first to introduce visit relations to tackle the malicious URL detection problem. The introduction of visit relations limits the scope of URL inspection and enables this approach to have the ability of self-learning. Experimental results show that the overall "toxicity" can be improved by 6.97% - 50.38% compared with full inspection of access logs.

I. Introduction

Malicious web pages have become the main tool for attackers to organize criminal activities such as spamming, phishing and driven-by-download. With the aid of these pages, attackers are able to spread malicious program [1], steal other's private information, and deliver spam advertisements. As reported by Kaspersky Lab [2], malicious web pages are the most active program and play a vital role in 87.36% web attacks. Among them, there is a kind of malicious pages that spread appealing but harmful information (e.g., *pornography and violence information*) to the victims. More specifically, these pages with harmful content lure victims to engage in criminal activities and often link to other malicious activities (e.g., trojans and phishing pages). Different from other malicious sites, these web pages provide victims with appealing content so that victims may *initiatively* visit it or related pages later. In order to avoid the potential risks aroused by visiting these pages, it is the first and most critical task to find them in time.

To address this issue, researchers have proposed many solutions [3] [4] [5] [6] [7] at present. And they usually divide the searching process [3] for malicious pages in three steps: collect candidate web pages that are live on the Internet; filter massive legitimate pages with fast-filters and finally examine the remaining web pages using specialized analyzer. More precisely, one has to first collect pointers to candidate pages (URLs) by using web crawlers or logs [4]. Given the candidate URLs, one requires a fast, but possibly imprecise pre-filter to discard massive legitimate pages. Such pre-filters examine static properties of URLs and decide the possibility that a page is malicious. In this way, a pre-filter can shrink the number of pages to be inspected fully by three orders of magnitude [5]. For the third step, the detection system should determine whether a given web page is malicious with high accuracy. To this end, some systems [6] [7] examine the HTML content as well as its active elements using static or dynamic analysis techniques. And the other detection systems [5] [8] [9] introduce honey-clients and monitor the changes of local systems (e.g., the creation of files or additional processes) once a page has been loaded.

However, there are several new challenges for this task, which are listed as below.

- *Large-scale*: The web is a fairly large place where new pages (both malicious and legitimate) are added at a daunting pace [3].
- *Imbalance data*: The number of malicious pages is just like a drop in the ocean as compared with that of legitimate ones. As reported in [10], Google Safe Browsing technology examines billions of URLs per day only to discover thousands of new unsafe sites.
- *Usage of cloaking techniques*: Network security can be an arms race [4] and many attackers have already used cloaking techniques to avoid detection from traditional methods. More precisely, malicious sites are sometimes not blacklisted [11] or even evaluated incorrectly due to the usage of cloaking techniques. For example, some malicious sites tend to be environment-specific [12] in order not to be detected by heuristic crawlers.

Besides, the time and computing resources to identify potential malicious pages are neither free nor infinite. Therefore,

it is essential to perform such searching process in an efficient way.

In this paper, we present a system that improves the efficiency of the first step of the process relying on the access log of the gateway. More precisely, we propose a system, named GuidedTracker, with a *guided track* for victims' access log of malicious URLs. GuidedTracker starts from a set of known URLs that are involved in malicious activities. With tracking these URLs, it focuses on the visit relations recorded in gateway logs and figure out those victims who frequently get access to these malicious URLs. In this article, visit relation refers to a kind of relation generated by the visitor accessing the web resources (e.g., URL, website and FTP server). In the next step, the visit relations corresponding to these victims are used to narrow the scope of suspicious URLs, avoiding the large scale of content inspection. We call these two steps as a target tracking process, because it searches for potential malicious pages by tracking the known malicious URLs and their associated victims. Of course, the tracking process is not guaranteed to return only real victims and malicious pages. Thus, it is necessary to analyze the track result by combining other techniques including fingerprinting techniques [13] and honey-clients [9].

However, the key advantage of our approach is that a result provided by GuidedTracker is *much more likely* to be malicious than other URLs recorded in the access log of the gateway. Thus, given a fixed amount of resources, GuidedTracker allows us to find more malicious pages in a relatively short period.

GuidedTracker can also be beneficial since it is a self-learning approach. Unlike traditional blacklist-based methods, once the initial URL blacklists is not empty, GuidedTracker will 'learn' the information of victims from visit relations related to blacklists. And the access intersection of victims will then 'teach' the system in return where to discover more suspicious URLs. In this way, GuidedTracker adapts itself to the change of victims and their access habits.

To our best knowledge, we are the first to introduce victim's visit relations in the URL classification problem. Different from existing works [11] [14] [15] [16], we classify URLs based on the hybrid usage of blacklists and visit relations. The main contributions of this paper are the following:

- We proposed a novel approach to find potential malicious sites based on an initial set of known malicious web pages and visit relations.
- We describe new techniques to identify more malicious web pages with a different angle. That is, we search for the potential malicious in the victims' access history.
- We implemented our techniques in a system and evaluated it on a large data set extracted from the real network traffic, demonstrating the approach is highly predictive and improves the state of the art.

II. Related Work

Finding malicious web pages on the Internet includes two main procedures: a detection process, which develop a specialized analyzer to decide whether a given page is malicious, and a searching process, which collect suspicious web pages to submit to the analyzer.

Detection process. Designing effective tools to detect malicious web pages (including phishing sites [17], spamming advertisements [18] and pages that perform driven-by-download attacks [5]) has attracted considerable attention. The proposed schemes can be classified into either static or dynamic detection systems. Some lightweight static detection systems focus on the lexical features of a URL [19], as well as DNS and WHOIS information [11]. And other static detection systems extract additional features from the HTML content and JavaScript codes [7]. As for the dynamic detection system, some approaches [20] [5] utilize high-interaction honeyclients to detect the unusual changes in the local system (e.g., the creation of new files or processes), while others [21] [22] detect malicious sites using low-interaction honeyclients or browsers with lightweight detectors. In this paper, we mainly make use of Google Safe Browsing blacklist [5], and take this tool as a black box. Therefore, it is possible to instrument GuidedTracker with any other available tool that focus on the detection of malicious web pages. Besides, honeyclients usually consume huge amount of resources to analyze a web page. So that a pre-filtering procedure [7] is often necessary because it is able to discard massive obviously benign pages.

Searching process. The searching process is to gather potential malicious web pages and submit them to the detection process. Previous work focused on web crawling, collecting suspicious pages based on several heuristic rules, and learning from the search logs which includes the attackers' behavior. Next, we discuss these schemes in detail.

Some researches [23] collect web pages using large web crawls. These massive crawls are able to produce a huge amount of web pages, and generate a "complete" view of the web [3]. However, the fact that these crawls need to consume substantial resources limits their practical use, so that they are available to just a few organizations. Besides, web crawls can be much smaller and targeted using some heuristic rules. These crawls [7] are available to most researchers since it requires a relatively low consumption of resources. However, they are not effective. For instance, Moshchuk et al. report a 0.4% detection rate by using a heuristic web crawler.

In addition to web crawls, some approaches [4] [24] focus on identifying potential malicious sites by analyzing clues left behind attack process. They argue that attackers often search for vulnerable web sites before they launch an attack. By identifying potential malicious queries from search engine logs, these approaches have proven to be effective pin the detection of compromised landing pages. However, this kind of detection schemes is not suitable to find malware distribution sites [25] since attackers do not need to search for their own sites.

Different from these schemes, our approach settled the problem from a different angle. Instead of searching clues attackers left to identify malicious queries, we turn to detect victims, and inspect their access log to find suspicious malicious web

sites. In this way, it mines malicious instances and access relationship to generate a URL stream with high toxicity. As our experiments point out, detecting victims and their access records improves the effectiveness of the detection process of malicious web pages.

III. GuidedTracker: A Guided Approach to Find Malicious Web Pages

In this section, we describe the motivation of our work and provide a detailed overview of the overall approach and the components in our system.

A. System Goal and Motivation

As mentioned previously, detecting malicious pages is a three-step process: collect candidate URLs, discard massive legitimate pages with a fast filter, and conduct a precise inspection for the remaining pages. In this paper, our goal is to improve the efficiency of the collecting phase. In other words, we aim to develop tools to gather URLs with a higher "density" than the URLs that can be discovered through (random) log inspection. The density discussed in this paper refers to the percentage of malicious URLs in a set.

Our tools are based on the idea of searching the potential malicious sites in the victims' access history. Intuitively, rather than randomly inspecting pages in the access log, GuidedTracker focus its searches on the access records of victims who once visited known malicious sites. More precisely, GuidedTracker first search for victims in terms of known malicious sites and visit relations; for those victims who frequently visit a known malicious site, their access to other web pages are also suspicious. Then, by regarding victims as guides, GuidedTracker perform a targeted search for potential malicious sites in the access log.

GuidedTracker is built on two key insights. The first one is that for victims who visit known malicious web sites, their access to other web sites is also suspicious. The reason is that some sites may offer victims with attractive content so that victims has the potential to visit it or related pages later. Besides, the adversaries may utilize automation tools in their campaigns. For instance, cybercriminals often link many malicious pages to a single site to simplify management [3]. When victims browse these pages, their access for other web pages is also worth detection.

The second insight is that there are techniques and tools [26] [13] that can identify different individual hosts, so that victims can be located precisely. A simple idea is to distinguish different users using IP address. But in a real network environment, the usage of Network Address Translation (NAT) technique offers people an opportunity to share the same IP address within an organization, which result in the mixture of these users' access log. To cope with this challenge, people have developed sophisticate tools to detect NAT and count NATted hosts by using OS fingerprinting [27], clock skews [13] and traffic features [28]. We leverage these techniques to distinguish different individual hosts.

B. Architecture

Fig. 1 presents the architecture of GuidedTracker. At a high level, GuidedTracker can be seen as having two phases. In the first phase, it extracts pointers of web pages (URLs) from every access record, analyzes each page with a set of conventional detection process (including a precise analyzer and optionally a pre-filter). In this way, it identifies malicious web pages slow but precisely and puts them into the seeds set. In the second phase, it searches for victims who once visit those known malicious sites, and makes use of their access records to find potential malicious web pages. In the following paragraphs, we further describe the components in GuidedTracker in more detail:

Seed. The seed is a set of web pages which has been previously identified as malicious pages. As the major input of victim detector, the quality of seed is the key factor of the tracking process. And it is produced by specialized analyzer. In fact, whenever the analyzer mentioned above discover new malicious web pages, they can be added to the seed pages for the following victim detection process. Besides, there are two main types of pages in the seed. First, there are pages that are directly set up by the attackers or criminals. Usually, these pages either link to a malware program directly, or contain malicious code that can be carried out under certain conditions. Besides, these pages are often linked to each other to increase the possibility of a successful invasion. The second type of pages usually belongs to legitimate sites. Different from other legitimate pages, they have been compromised by attackers and usually redirect users to malicious sites by embedding a piece of JavaScript. By adding these pages to the seed set, it is possible for our system to track these victims into their access history to find potential malicious pages (① in Figure 1).

Victim detector. Victim detector is the key component of GuidedTracker. It takes in the seed (including known malicious pages) as well as "multi-to-multi" visit relation. Based on the analysis of known malicious pages and visit relations, the detector generates the set of victims (② in Figure 1). The victims are those who once visited the known malicious sites and have the potential to visit them or similar pages later. The set of victims returned by the detector are then delivered to suspicious URL parser. By detecting victims in the network flow, the system has the potential to figure out their access history and find other malicious sites.

Suspicious URL collector. The function of suspicious URL collector is to collect victims' visited URL from the access logs based on the victim set. It is equivalent to a funnel that only victims' visited URLs can bypass it. In this way, it produces a set of URLs which are much more likely to be malicious, and sends it to a group of (existing) analyzer for further detection (③ in Figure 1).

Specialized analyzer. Our specialized analyzer is made of Google Safe Browsing blacklist [5]. This blacklist is publicly accessible through its Safe Browsing API, and has been used to analyze more than one billion pages daily. Besides, it is a constantly updated blacklist [3], with a very low false positive

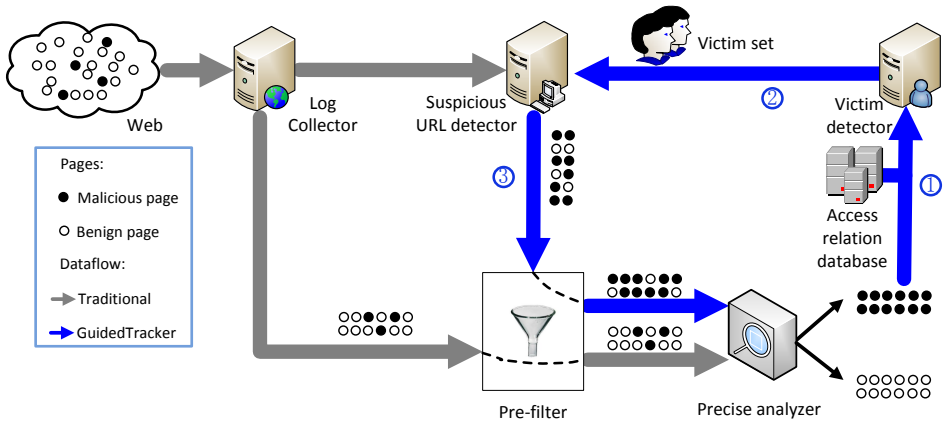

Fig. 1. Framework of GuidedTracker.

rate. We do not implement a pre-filter in the current version of GuidedTracker to discard massive benign URLs. But it is certainly an option as shown in Fig. 1. And it will not influence the computation result of "density" (the percentage of malicious URLs).

IV. Evaluation

In this section, we evaluate the performance of our GuidedTracker approach by using the real-world data set. Our data set, as captured from a real campus gateway, is very large and extremely imbalanced. That is, the number of malicious URL and benign ones is not in an order of magnitude. Therefore, it is not suitable to measure our system with accuracy and recall rate. Instead, we utilize the other two key indicators that were once created by [3] to validate the effectiveness of our system: *toxicity* and *expansion*.

The *toxicity*, equivalent to "density", represents the percentage of new URLs submitted to the specialized analyzer that are eventually proved to be malicious. And higher toxicity indicates that the resources used for detection are used in a relatively efficient way. For instance, if a suspicious URL collector delivers 100 suspicious URLs to the analyzer, and 10 of them are identified as malicious eventually, then the toxicity of the system is 0.1.

The *expansion* represents the average number of malicious URLs that are found by GuidedTracker for each seed. A higher expansion implies that a large number of malicious URLs are found for each seed, and the seed is used much more efficiently.

There is a trade-off between these indicators. Considering the practical application of our approach, especially the imbalance data set we face, to obtain a higher toxicity is relatively much more important for us. By evaluating these indicators, we hope to answer the following questions: How efficient is GuidedTracker in finding new malicious URLs? How long will GuidedTracker spend for its execution?

We start this section with the description of the data set, followed by the analysis and discussion of experimental results.

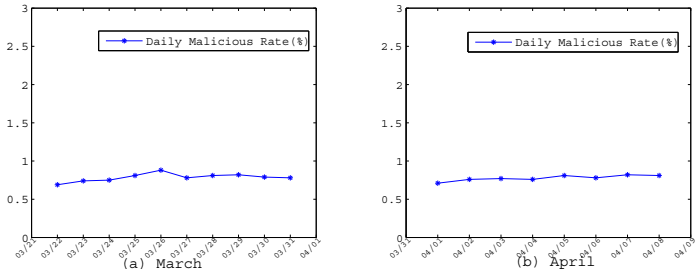

Fig. 2. Malicious Rate of Each Day.

TABLE I
The Information of Data Set

	Total Number	Number of Attacks	Malicious Rate
Access Records	12,357,243	98,134	0.79%
Users (Victims)	171,146	2,475	1.45%
Web Pages	3,155,234	40,581	1.29%

TABLE II
Example for User Access Behavior

User ID	IP address	Time	Visited URL
000012	210.242.14.xx	22/3/2013 14:52:47	yahoo.com/example.htm

A. Data Set

The data set is provided by a network operator and collected from a campus gateway during March 22,2013 - April 8,2013. All the URLs in the dataset are labeled as benign or malicious in advance. The breakdowns of the data set are shown in Table I.

We run our experiments on a machine with 8 core 2.13 GHz Xeon processors with 16GB memory. And an example of access record is shown in Table II. As shown in Table II, the log collector records the information of every access, including the visitor, visited time and visited web page. These web pages are evaluated and labeled by introducing Google Safe Browsing [10] for judgement.

In our experiments, the data set is segmented by days. The

daily malicious rate for each day is shown in Figure 2. The data sets have 686,513 access records for each day, but only contain no more than 1% malicious records. In traditional classification schemes, the classification model may classify all the URLs in the data set as benign ones in order to correctly predict most instances.

B. Performance Evaluation

GuidedTracker focuses on collecting potentially malicious URLs from access log, so we ran our system in parallel with traditional inspection methods on access log. First, we conduct an in-depth inspection with a small proportion (0.2% in this paper) of web pages to feed GuidedTracker. In other words, whenever a malicious URL is found by the specialized analyzer in the start operation, it is added to the seed set used by GuidedTracker (shown in Figure. 1). Then, GuidedTracker makes full use of the seeds to detect victims and produce malicious URLs, and compares the impact of victim selection ratio to the detection result. In order to simplify the comparison between victims, the system rank them before analyzing URLs. Finally, we conduct two reference approaches for comparison: a traditional in-depth inspection with half / full amount of web pages recorded in the access logs.

Table III shows an overview of the comparison results of our experiments. The start operation discovers 67 malicious URLs (these are used as the seed for GuidedTracker), for a toxicity of 1.06%. Based on these seeds, GuidedTracker submitted up to 18,440 URLs to the precise analyzer, of which 254 were found to be malicious, for a toxicity of up to 1.94%. Interestingly, the toxicity of victim's access result is 6.97%-50.38% higher than the full inspection of access records, likely indicating that the access of victims is highly predictive than that of others.

The seed expansion refers to the ratio between the number of malicious pages which were finally identified and the initial seed. As Table III demonstrates, inspecting the victims' visited pages can generate a stream of novel malicious URLs which is at least 3.25 times larger than the number of seeds.

C. Time Performance

Table IV shows the comparison of average time performance among these approaches. It is clear that using victim as guiders is a useful way to find more malicious URLs, reducing the averagely processing time with up to 33.89% discount of time. Interestingly, if the system chooses a smaller ratio of victims, it will spend a much shorter average time to find a malicious URL. It is likely because we rank victims with their occurrences and choose them from high to low for each time.

D. Discussion

Overall, our results have shown that GuidedTracker clearly outperforms in toxicity(1.29% vs 1.94%) against full inspection. Besides, the usage of GuidedTracker will reduce the average time by up to 33.89%. In addition, these results also indicate that given a few malicious pages as the set of initial seed, GuidedTracker is able to find a huge amount of additional malicious pages. More precisely, GuidedTracker discovers over three times more malicious pages than the full inspection of log records.

V. Conclusion

Although many conventional schemes have been proposed, how to detect malicious web pages effectively and efficiently remains a practical and vital problem. The most challenging work of this problem is to find a relatively small portion of malicious URLs out of a large volume of URLs. Besides, the usage of cloaking techniques also increases the difficulty of conventional detection process. In order to cope with these challenges, it is necessary to improve the detection techniques in an efficient way.

In this paper, we proposed a novel approach, called GuidedTracker, which focuses on the improvement of efficiency for the search process of malicious URLs. We utilize a set of known, malicious web sites as seeds and mine the victims who get access to them. Then, the victims will guide our system to locate potential malicious web sites. In this way, GuidedTracker can generate a set of candidate pages that owns a higher ratio of malicious web pages compared to the log inspection. Moreover, GuidedTracker is able to detect more than 3 times of new malicious urls than the number of seeds. In this way, it will significantly reduce the amount of URLs that receive precise inspection. Therefore, it is possible to detect potential malicious pages effectively by using GuidedTracker.

Next, we will first introduce white-list in order to obtain a higher toxicity. Then, we will study how to improve the experimental results in dynamic environments. .

Acknowledgment

This work was supported by The National High Technology Research and Development Program of China (863 Program), Grant No. 2011AA010703; the National Natural Science Foundation (Grant No. 61070026).

References

[1] Z. Zhao, G.-J. Ahn, and H. Hu, "Examining social dynamics for countering botnet attacks," in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*. IEEE, 2011, pp. 1–5.

[2] D. Maslennikov and Y. Namestnikov., "Kaspersky security bulletin. statistics 2012," http://www.securelist.com/en/analysis/204792255/Kaspersky.

[3] L. Invernizzi, P. M. 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*. IEEE, 2012, pp. 428–442.

[4] J. P. John, F. Yu, Y. Xie, M. Abadi, and A. Krishnamurthy, "Searching the searchers with searchaudit." in *USENIX Security Symposium*, 2010, pp. 127–142.

[5] N. P. P. Mavrommatis and M. A. R. F. Monrose, "All your iframes point to us," 2008.

[6] M. Cova, C. Kruegel, and G. Vigna, "Detection and analysis of drive-by-download attacks and malicious javascript code," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 281–290.

[7] 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*. ACM, 2011, pp. 197–206.

TABLE III
GUIDEDTRACKER EVALUATION:DOMAINS AND URLS

Source	Seed pages	Victims selected(ratio)	URLs analyzed	Malicious URLs	Toxicity	Expansion
Seed Detection (Start operation)			6,310	67	1.06%	
GuidedTracker						
	67	223(10%)	11,215	218	**1.94**%	3.25
	67	1,115(50%)	16,974	247	1.46%	3.69
	67	2,230(100%)	18,440	254	1.38%	**3.79**
Pre-filter/Log Analysis						
Half Inspection			1,606,014	17,511	1.09%	
Full Inspection			3,155,234	40,581	1.29%	

TABLE IV
THE COMPARISON OF TIME PERFORMANCE

Source	Victims selected(ratio)	Malicious URLs	Processing time(s)	Average time(ms)	Reduction(%)
Seed Detection (Start operation)		67	8.14	121.48	
GuidedTracker					
	223(10%)	218	14.46	**66.3**	**33.89**
	1,115(50%)	247	21.89	88.6	11.66
	2,230(100%)	254	23.79	93.64	6.36
Half Inspection		17,511	2071.63	118.3	-17.96
Full Inspection		40,581	4070.12	100.29	

[8] Y.-M. 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*, 2006, pp. 35–49.

[9] C. Seifert and R. Steenson, "Capture-hpc," *Internet: https://projects. honeynet. org/capture-hpc*, 2008.

[10] G. T. Report, "Making the web safer," http://www.google.com/transparencyreport/safebrowsing/?hl=en.

[11] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: learning to detect malicious web sites from suspicious urls," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 1245–1254.

[12] C. Kolbitsch, B. Livshits, B. Zorn, and C. Seifert, "Rozzle: De-cloaking internet malware," in *Security and Privacy (SP), 2012 IEEE Symposium on*. IEEE, 2012, pp. 443–457.

[13] T. Kohno, A. Broido, and K. C. Claffy, "Remote physical device fingerprinting," *Dependable and Secure Computing, IEEE Transactions on*, vol. 2, no. 2, pp. 93–108, 2005.

[14] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Learning to detect malicious urls," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, p. 30, 2011.

[15] S. Lee and J. Kim, "Warningbird: A near real-time detection system for suspicious urls in twitter stream," *IEEE Transactions on Dependable and Secure Computing*, vol. 10, no. 3, pp. 183–195, 2013.

[16] P. P. Tsang, A. Kapadia, C. Cornelius, and S. W. Smith, "Nymble: Blocking misbehaving users in anonymizing networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 2, pp. 256–269, 2011.

[17] A. Le, A. Markopoulou, and M. Faloutsos, "Phishdef: Url names say it all," in *INFOCOM, 2011 Proceedings IEEE*. IEEE, 2011, pp. 191–195.

[18] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri, "Know your neighbors: Web spam detection using the web topology," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2007, pp. 423–430.

[19] M.-S. Lin, C.-Y. Chiu, Y.-J. Lee, and H.-K. Pao, "Malicious url filteringła big data application," in *Big Data, 2013 IEEE International Conference on*. IEEE, 2013, pp. 589–596.

[20] A. Moshchuk, T. Bragin, D. Deville, S. D. Gribble, and H. M. Levy, "Spyproxy: Execution-based detection of malicious web content," in *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, no. 3. USENIX Association, 2007, pp. 1–16.

[21] A. Ikinci, "Monkey-spider: Detecting malicious web sites," *Master's thesis, University of Mannheim*, 2007.

[22] P. Ratanaworabhan, V. B. Livshits, and B. G. Zorn, "Nozzle: A defense against heap-spraying code injection attacks." in *USENIX Security Symposium*, 2009, pp. 169–186.

[23] J. W. Stokes, R. Andersen, C. Seifert, and K. Chellapilla, "Webcop: Locating neighborhoods of malware on the web," in *USENIX Workshop on Large-Scale Exploits and Emergent Threats*, 2010.

[24] T. Moore and R. Clayton, "Evil searching: Compromise and recompromise of internet hosts for phishing," in *Financial Cryptography and Data Security*. Springer, 2009, pp. 256–272.

[25] H. Zhang, D. D. Yao, and N. Ramakrishnan, "Detection of stealthy malware activities with traffic causality and scalable triggering relation discovery," in *Proceedings of the 9th ACM symposium on Information, computer and communications security*. ACM, 2014, pp. 39–50.

[26] S. Mongkolluksamee, K. Fukuda, and P. Pongpaibool, "Counting natted hosts by observing tcp/ip field behaviors," in *Communications (ICC), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1265–1270.

[27] R. Beverly, "A robust classifier for passive tcp/ip fingerprinting," in *Passive and Active Network Measurement*. Springer, 2004, pp. 158–167.

[28] L. Rui, Z. Hongliang, X. Yang, L. Shoushan, and W. Cong, "Passive natted hosts detect algorithm based on directed acyclic graph support vector machine," in *Multimedia Information Networking and Security, 2009. MINES'09. International Conference on*, vol. 2. IEEE, 2009, pp. 474–477.