International Journal of Trend in Scientific Research and Development (IJTSRD)
International Open Access Journal
ISSN No: 2456 - 6470 | www.ijtsrd.com | Volume - 2 | Issue – 1

Significant factors for Detecting Redirection Spam

Dr. Prabha Shreeraj Nair
Dean Research, Tulsiramji Gayakwade Patil College of Engineering and Technology, Nagpur

ABSTRACT

Redirection spam refers to a technique where a genuine search user is befooled and made to pass through a chain of redirections and ultimately presented with a compromised web page that may be an adware or an irrelevant content for the user. As a consequence the search engines earn a bad name in quality information retrieval and moreover the users are too dissatisfied. Also, there is sever wastage of expensive network resources like bandwidth. Detecting these malicious redirections is important for quality information retrieval from web. But detecting such redirections is a very tedious task due to the genuine usage of redirections to provide load balancing and URL shortening features. Also, the conventional methods of spam detection such as blacklists, whitelists are not very successful as they need to be updated every time. Many factors have been discussed in previous work that facilitates redirection. To design a more robust and reliable approach, this paper presents some new factors that facilitate redirection spam detection. We also explored the operational profile of each identified factor along with the criteria for its selection.

Keywords: *redirection spam; malicious redirection; spam detection; web security; Iframe; JavaScript redirections*

I. INTRODUCTION

Redirection Spam has emerged as a prime challenge in the quality information retrieval in the current scenario. The openness of web is a blessing for its growth on one side whereas at the same time this openness presents web with a major challenge of spam which acts as a hindrance in quality information retrieval [1]. Redirection spam refers to a technique where a genuine search user is befooled and forced to pass through a number of redirections and finally reaching a compromised web page. Gyongyi and Garcia-Molina [2] state that as redirection is encountered, browser is made to visit another URL at the time of page load. The miscreants try to induce malicious redirections with the aim to damage the competitors, to earn a better page ranking, phishing or business promotions through publicity [3]. A study [4] indicated that the syndicate programs that redirected towards adware pages were prevalent and covered 40% of the observed redirection cases. As stated by Zhou and Ding [5], the search engines are not capable of indexing the dynamic scripts. The client side scripting languages provide features that facilitate redirections[6], therefore taking advantage of this fact, the spammers intelligently craft the chain of redirections so as to evade detection by embedding their redirect codes inside dynamic scripts The genuine search user becomes the victim of such spammers and loses trust in search engines. Spam has adverse effects on the integrity and security of information. It also affects the search engine ranking of web sites. Also, it harms the reputation of search engines and face the expense of crawling the spammed web sites.

Redirection attacks have been more prevalent in the recent years. A Report produced by Dell 2016 states that spammers employed redirection chains to spread the SPARTAN exploit kit with the aim to download malware and infect the machines which led to heavy losses. According to a report [8], heavy redirection attacks were triggered on the corporate accounts and banks by a Trojan named Dridex. This attack was very similar to the Carbanak and Dyre Wolf attacks which hit the year 2015. Another report [9] states that Goznym redirections attacks affected 24 banks in America in April 2016, and further spread to the other

countries like USA and Germany in another two months. Another heavy redirection attack TrickBot [10] hit the corporates and business accounts in October 2016 that spread through malvertising. These reports indicate that addressing the redirection spam is the need of today's scenario. It becomes absolutely essential to take active measures against the evil of redirection spam. In this paper we aim to identify the critical factors for redirection spam detection. Section 2 presents the related work. Section 3 describes the significant factors for detecting redirection spam which is followed by conclusion and future work in Section 4.

II. RELATED WORK

Spam detection has been emerging research area for both academia and industry. Lot of efforts have been made in detecting redirections. Wang and Ma [11] developed a system that aims at querying the search engines for commerce terms and visiting those URLs thereby recording all traffic information. Their main focus was to perform similarity analysis and identify spammers based on similar domains. They observed that most spammed websites made use of front-end servers that automatically redirected the browser traffic to an exploit server working at a back-end point. Their system made use of honey-monkey that collected spammer targeted keywords. They used the persistent –state changes and changes in window registry entries as their main feature to detect exploit sites. They used a blackbox, signature free approach and run their system inside using virtual machines. However, the system designed could not detect the exploits that do not make any changes in persistent state.

Chellapilla and Maykov [12] studied the distribution of different types of redirections on web. They made use of a JavaScript parser/editor and JavaScript runtime engine for understanding the run time behavior and exploits. They concluded that JavaScript redirections were very difficult to detect because JavaScript has some features like eval, window. location, and location. Replace that support script obfuscation or dynamic code injection.

Wu and Davison [13] conducted an exploratory study on redirections and considering redirection as a technique for web spam. They found Http status code and html refresh tag of each URL for detecting redirections. Their work was limited to analyzing the distribution of Http and Meta redirections. JavaScript redirections, owing to their complex nature were not considered.

Niu et al. [14] regarded redirections as a major technique used behind blog and forum spam. Their employed a behavior based and signature approach that considered top domain analysis and third party domains to identify the backdoors and redirects. They made use of Strider URL tracer which provided a top domain view by visiting each URL.

Bhargrava et al. [15] performed a detailed study on web redirections. They observed that both spam URLs and legitimate URLs are using redirections with almost same frequency i.e. 43.63% and 40.97% respectively. They also observed that JavaScript redirections were more prevalent in the spam.

Leontiadis [16] tried to identify redirection chains in medical firms. Their technique was based on checking referrer field and probing backwards tracing all intermediaries. Also they considered Http status code and the user agent field to know whether the requests are coming from server or crawler. Their main objective was to study the impact of malicious redirections in pharmacy.

Takata et al.[17] studied the sequence of packets in each session between browser and server and performed referrer lookup and a host test to identify redirections. They extracted Http header variables like referrer, Server and Vary Field and also GET variable in the URL to conduct their study. However, their work does not analyze JavaScript code in detail.

Lu et al. [18] monitored the page load and extracted network features to detect malicious search user redirections. Their features mainly included number of redirection hops, landing to terminal distance, search rank and keyword poison resistance which they fed to a statistical classifier. Their component however worked only for malicious search user redirections.

Lee and Kim [19] tried to identify series of URL redirections in twitter stream. They constructed a feature vector including features like URL redirect chain length, Frequency of entry point URL, Relative number of different initial URLs. They assumed that owing to the limited resources attackers may reuse them, therefore all correlated redirect chains might share the identical URL. Their focus was to detect co-related URL redirect chains and to determine whether URL is suspicious.

Mekky et al. [20] followed an approach of browsing the nodes of user activity tree with simultaneous recording of network traffic. They analyzed this traffic to detect Http redirections. The source URL was considered as the node of tree and if it triggered a request for URL of child node, then an edge was assumed to exist between them. Their laid emphasis on factors like Http status code, path length, and edge duration and used a decision tree classifier.

Rahman et al. [21] observed that Facebook was used as a platform for redirecting genuine users to download the malicious apps. They designed a system called Frappe which detects URL redirects by calculating trust reputation score of each URL obtained after redirection. They found that a number of URLs were hosted by a single domain and various malicious apps were redirected to these URLS after app installation.

Zhang et al. [22] designed a system called Vishunter to detect malicious redirections from visible servers to invisible servers and uncover the malicious web infrastructures. They extracted features based on location, graph, role and relation

From this extensive literature survey, it is evident that many efforts have been made in detection of malicious redirections. Some researchers have used signature based approaches and some have used client side honeypot to trace redirections. Although the available approaches are able to detect malicious redirections to great extent, but in our view, the World Wide Web is dynamic in nature and spammers keep on evolving new tricks to evade detection. Also, there is a need to devise a more robust solution that could improve the accuracy and reduce the cases of misclassification of URLs. Therefore, some soft computing based techniques can prove useful for redirection spam detection.

III. SIGNIFICANT FACTORS

Based on the extensive literature survey and our previous work to study the actual redirection attacks and their behavior [4]. We identified the most significant factors that could facilitate in detecting malicious redirections. These factors are described as follows:

A. Number of Domains

Considering the URL structure, the right most part of URL represents the top level domain (TLD) and the next left part of it represents the second level domain (SLD) and so on. A domain change is reflected when a user shifts from one TLD to TLD or from one SLD to another SLD. We identified this factor as important in case of redirection spam detection because spammers try to hide their mischievous activities behind the scenes by switching to different domains. More number of different domains represent higher chances that the redirection is spam.

B. Number of Script Generated Redirections

JavaScript is a versatile language that enables programmers to develop and design websites but at the same time, it has some functions like eval(), window.location() that can be exploited by spammers. If usage of such functions is found in conjunction with a different URL in script, we have considered it as malicious redirection.

C. Number of Redirection Hops

A hop is counted when a node is crossed in reaching from source to destination. i. e the number of times a user is redirected to a new node or machine. The more number of redirection hops represent higher chances of redirection spam.

D. Delay in Page Refresh

Spammers set the spammed URL link in Meta tag and setting the page reload time to less than 5 seconds. The Meta tag has option to set the time in seconds for the page reload or refresh. This practice is referred to as Meta tag redirections. This feature is exploited by spammers and such cases are considered as malicious redirection. The lesser is the time specified the more is the risk for redirection spam. The assumption is spammer wants the spammed web page to be loaded as soon as the request is made.

E. Http Status Code

A webpage request by the client to the server leads to transmission of request and response headers. The status code is a three digit numeral and an integral part of response header which represents the status of request. Each code has a different meaning for the network. We have identified the status code as significant factor as the status code with 3xx value represents a redirection.

F. Use of Iframes

Spammers are using iframes to set their spammed URL link in iframe tag and making it invisible by setting its border property to zero. This tactic is used by them to evade detection. This kind of redirection is referred to as iframe redirections and has now become a common practice for spammers as it provides for doing spam activities behind the scenes. Such iframes can be checked for in scripts to check for malicious redirections.

IV. CONCLUSION AND FUTURE WORK

Spam has adverse effects on the integrity and security of information. This paper deals with the issue of redirection spam detection. To design a more robust and reliable approach, this paper identifies the significant factors for redirection spam detection. We also explored the operational profile of each identified factor along with the criteria for its selection. In future, we will consider these factors for designing a soft computing based approach that would provide a more robust and reliable solution for the evil of redirection spam.

REFERENCES

1. C. Castillo and B. D. Davison, "Adversarial Web Search," in *Foundations and Trends® in Information Retrieval*, vol. 4, no. 5, 2011, pp. 377–486.
2. Z. Gyongyi and H. Garcia-Molina, "Spam: it's not just for inboxes anymore," *Computer*, vol. 38, no. 10, pp. 28–34, Oct. 2005.
3. V. M. Prieto, M. Álvarez, R. López-García, and F. Cacheda, "Analysis and {Detection} of {Web} {Spam} by {Means} of {Web} {Content}," *Proceedings of the 5th {Conference} on {Multidisciplinary} {Information} {Retrieval}*, pp. 43–57, 2012.
4. K. Hans, L. Ahuja, and S. K. Muttoo, "Characterization and detection of Redirection Spam," in *Wilkes-100 International Conference on Computing Sciences (ICCS'13)*, 2013, pp. 325–331.
5. Jingyu Zhou and Yu Ding, "An Analysis of URLs Generated from JavaScript Code," in *2012 IEEE/ACIS 11th International Conference on Computer and Information Science*, 2012, pp. 688– 693.
6. V. M. Prieto, M. Álvarez, R. López-García, and F. Cacheda, "Analysing the Effectiveness of Crawlers on the Client-Side Hidden Web," in *Trends in Practical Applications of Agents and Multiagent Systems.*, 2012, vol. 32, pp. 141–148.
7. Dell, "Security Annual Threat Report," 2016. [Online]. Available: http://www.netthreat.co.uk/assets/assets/dell-security-annual-threat-report-2016-white-paper-197571.pdf. [Accessed: 09-Jun-2016].
8. Trojan Variants, "Dridex — NJ Cybersecurity," *New Jersey Cyber Security Cell*, 2016. [Online]. Available: https://www.cyber.nj.gov/threat-profiles/trojan-variants/dridex?rq=dridex. [Accessed: 02-Jan-2017].
9. Limor Kesor, "GozNym's Euro Trip: Launching Redirection Attacks in Germany," 2016.
10. Lior Keshet, "An Aggressive Launch: TrickBot Trojan Rises With Redirection Attacks in the UK," *Security Intelligence*, 2016. [Online]. Available: https://securityintelligence.com/an-aggressive-launch-trickbot-trojan-rises-with-redirection-attacks-in-the-uk/. [Accessed: 05-Mar-2017].
11. Y. M. Wang and M. Ma, "Strider search ranger: Towards an autonomic anti-spam search engine," in *In Fourth International Conference on Autonomic Computing*, 2007, pp. 32–42.
12. K. Chellapilla and A. Maykov, "A taxonomy of JavaScript redirection spam," *Proceedings of the 3rd international workshop on Adversarial information retrieval on the web - AIRWeb '07*. ACM, Alberta, Canada, pp. 81–88, 2007.
13. B. Wu, Baoning and Davison, "Cloaking and Redirection: A Preliminary Study," in *First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb'05)*, 2005, pp. 7–16.
14. et al Niu Yuan, Wang Yi-Min and Chen Hao, "A Quantitative Study of Forum Spamming Using Context-based Analysis cloaking redirection," in *Proceedings of 15th Network and Distributed System Security (NDSS) Symposium*, 2007, pp. 1–15.
15. Krishna Bhargrava Vangapandu, Douglas Brewer, Kang Li, "a study of URL redirection

indicating spam.pdf," in *Sixth Conference on E-mail and Anti-Spam*, 2009, pp. 1–4.

16. N. Leontiadis, T. Moore, and N. Christin, "Measuring and Analyzing Search-Redirection Attacks in the Illicit Online Prescription Drug Trade," in *20th USENIX Security Symposium.*, 2011, pp. 1–17.

17. Y. Takata, S. Goto, and T. Mori, "Analysis of Redirection Caused by Web-based Malware," in *Proceedings of the Asia-Pacific Advanced Network*, 2011, vol. 32, pp. 53–62.

18. L. Lu, R. Perdisci, and W. Lee, "SURF: Detecting and Measuring Search Poisoning," in *Proceedings of the 18th ACM conference on Computer and communications security*, 2011, pp. 467–476.

19. 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.

20. H. Mekky, R. Torres, Z. L. Zhang, S. Saha, and A. Nucci, "Detecting malicious HTTP redirections using trees of user browsing activity," in *IEEE Conference on Computer Communications*, 2014, pp. 1159–1167.

21. S. Rahman, T. Huang, H. V Madhyastha, and M. Faloutsos, "Detecting Malicious Facebook Applications," *IEEE/ACM transactions on networking*, vol. 24, no. 2, pp. 773–787, 2016.

22. J. Zhang, X. Hu, J. Jang, T. Wang, G. Gu, and M. Stoecklin, "Hunting for Invisibility : Characterizing and Detecting Malicious Web Infrastructures through Server Visibility Analysis," in *The 35th Annual IEEE International Conference on Computer Communications, IEEE INFOCOM*, 2016, pp. 1–9.