Poster:An Efficient and Scalable Cyberlocker Traffic Tracking Method

Chao Zheng[1,2,3)], Peng Zhang[1,2)], Shu Li[1,2)] , Jianlong Tan[1,2)],Qingyun Liu[1,2,3)]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] National Engineering Laboratory for Information Security Technologies, Beijing, China
[3] University of Chinese Academy of Sciences, Beijing, China
{zhengchao,pengzhang,lishu,tanjianlong,liuqingyun}@iie.ac.cn

I. INTRODUCTION

Cyberlocker is an on-line internet site for storage of personal digital files and the surge in popularity of Cyberlocker traffic has been reported in literature. Gehlen et al.[1] found that a Cyberlocker was among the top-10 Web applications and constituted 5% of the total Web traffic. Maier et al. [2] reported that a single Cyberlocker consumed 15% of total Web bandwidth in a large residential network. Allot [3] reported that Cyberlocker traffic accounted for 19% of the total mobile broadband traffic. Despite their popularity, there has been limited work on understanding the characteristic of traffic flows originated from Cyberlocker.

Generally, when a user intend to download the resource shared by Cyberlocker through clicking the hyperlink, browser will pop up a page with download button, also called entry point. This page will show the resource information, such as publisher, posting time and so on. After user clicks the download button, his browser will automatically send out a series of requests to the server to generate redirection chains until successfully establishing the HTTP session to download resource.

In this work, we aim to design and build a system to track the Cyberlocker resource's entry point in large-scale network traffic, called Cyberlocker traffic tracking, is crucial for understanding the characteristic of traffic flows originated from Cyberlocker.

II. CHALLENGES

While investigating the possibilities we identified the following two challenges.

Building resource redirection chain: Tracking down redirection chains is difficult.[4] Using HTTP *Referer* field is simple but not always feasible, because *Referer* as an optional field of HTTP , only 17.7% HTTP sessions have this field in our observation of real network traffic. Meanwhile, the technologies such as NAT(Network Address Translation) lead the IP address of HTTP session cannot be as the sufficient evidence for accurately tracking redirection chain.

Locate the entry point: URLs of different Cyberlocker services are various and changeable. Locate entry point's URL in redirection chains with manually predefined templates is feasible, but it is a very high cost to maintain these templates.

For these reasons, we design a method for identifying and tracking the entry point from the redirection chains in large-scale network flow without templates, named CookieID, of which the contributions are two-fold: (1) Scalable HTTP redirection chain tracking with very few HTTP fields, which is using less memory and indifferent to the redirections hops, could easily be applied on large scale network traffic. (2) Benefit from the non-reusable of transitional URL and the stability of entry point URL, few of the Cyberlocker redirection nodes will appeared twice or more. After merging same resource's different redirection chains, we can discover the entry point without templates by calculating every node's repeat times efficiently.

III. THE IMPLEMENTATION OF COOKIEID

We designed CookieID to run on real network traffic with Cyberlocker entry point tracking functions which presents in Figure 1. It mainly contains five parts: (1) HTTP session header collection module prepare the URLs and cookies for (2) HTTP session indexing module to store them in hash tables. When a cyberlocker resource downloading initiated, (3) HTTP redirection chain tracking module build the chain of the resource by searching the hash tables. And the (4) Candidate entry point extraction module response for merge multi redirection chains together to locate the entry point. At last (5) Entry point validation module examine the result for reliability. In what follows, we will elaborate on the implementation of each module.

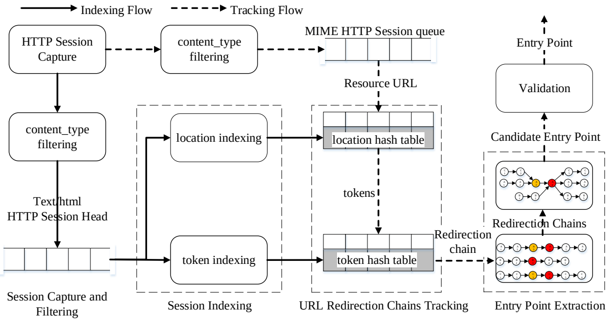

Fig. 1. The tracking process of entry point

The research work is supported by Strategic Priority Research Program of the Chinese Academy of Sciences under Grant (No.XDA06030602) and National Natural Science Foundation under Grant (No.61402464).

HTTP Session Header Collection Module: This module parses HTTP network traffic, and filter out two kinds of HTTP session header. 1) Possible cyberlocker redirect session which content type of HTTP session is text/html and the cookie field exists; 2) Possible cyberlocker download session which content type of HTTP session is multimedia, such as video/x-ms-wmv. The two kinds session's URL， cookie and time stamp is cached. Due to the download resource is a TCP flow in network traffic, we adopt a simple algorithm to streaming compute a 64 bit resource ID to represent the download resource, where the algorithm only performs calculation on the first 10% sampling resources. In our experiment, the generated resource ID's collision rate of different download resources is no more than 0.4%. It's a valuable trade-off between performance and uniqueness.

HTTP Session Header Indexing Module: This module split every possible cyberlocker redirect session's cookie into tokens by semicolon, and then caches each token and its URL as key and value in token hash table. A token may point to one more URLs during indexing.

URL Redirection Chain Tracking Module: Firstly, it find possible cyberlocker download session's cookie in location hash table. Secondly, it search token hash table with every token in the cookie and traverse pointed URLs. During traverse nodes in singly linked lists, we build some doubly linked lists to calculate appeared frequency of these URLs. Candidate redirection node are the ones which appeared frequency exceed a certain threshold. Finally, after sorting these nodes by time stamp, we got a redirection chain of a specifically Cyberlocker resource, e.g. 1-2-4-8 in figure 2.

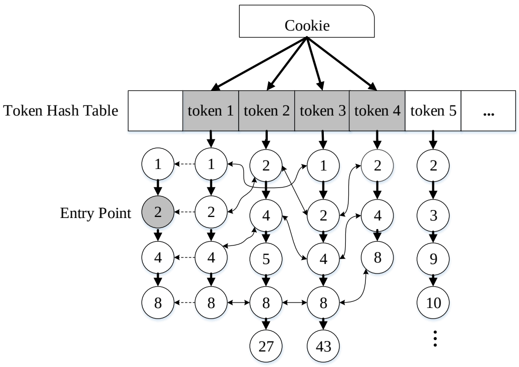

Fig. 2. URL redirection chain tracking process

Candidate Entry Point Extraction Module: This module is responsible for merging all URL redirection chains identified by same resource ID, and calculating the frequency of each node in the redirection chains, the most frequent node is extracted as a candidate entry point. If two or more nodes in the chains have same frequency, the URL nearest the downloaded resource will be select as the candidate entry point.

Entry Point Validation Module: This module is responsible for comparing the MD5 of the resource downloaded from candidate entry point with that from MIME HTTP head session, and determining the real entry point. If two resource's MD5s are same, the entry point is determined, otherwise the candidate is dropped.

IV. EXPERIMENTS

We evaluate the effectiveness and scalability of CookieID with four evaluation measures: precision, recall, search time and scalability. The precision is calculated as the correct number of found entry points divided by the total number of all found entry points. The recall is calculated as the number of the correctly found entry points divided by the total number of all entry points. The search time is calculated as the time spending on finding the entry point. The scalability is calculated as the number of found entry points per unit time. Firstly, we collect 500 shared video resource hyperlinks by exploiting Baidu Cyberlocker search engine. Then, we exploited LoadRunner [5] to simulate users to click these hyperlinks to collect the URL redirection chains, respectively. Finally, we tested CookieID method and List method on the created dataset. Figure 3 shows the results. As shown in Figure 2(a) and (b), in terms of precision, CookieID is slightly lower than List method. And in terms of recall, CookieID is slightly higher than List method. In addition, as shown in Figure 2(c), the search time of CookieID is significantly faster than List method. Meanwhile, from Figure 2(d), we can see that the number of found entry points per unit time of CookieID remains linear growth, while List decreases, which means that CookieID has good scalability.

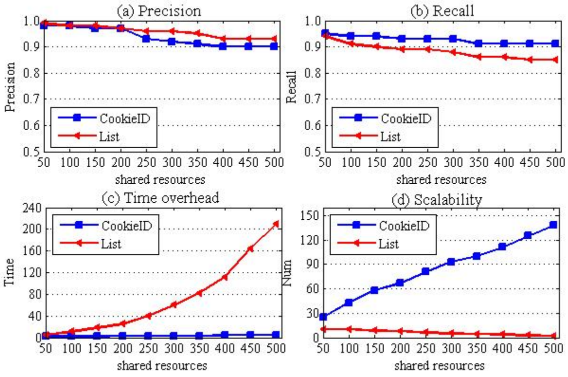

Fig. 3. The performance and scalability evaluation

REFERENCES

[1] V.Gehlen, A. Finamore, M. Mellia, and M. Munafo, "Uncovering the Big Players of the Web," in Proc. Traffic Monitoring and Analysis Workshop, Vienna, Austria, March 2012

[2] G. Maier, A. Feldmann, V. Paxson, and M. Allman, "On Dominant Characteristics of Residential Broadband Internet Traffic" in Proc ACM SIGCOMM Internet Measurement Conference, Chicago, USA, November 2009.

[3] Allot Communications, "Mobile Trends: Global Mobile Broadband Traffic Report," White Paper, 2010, http://www.allot.com/mobiletrends.html

[4] Z Li, S Alrwais, XF Wang,"Hunting the red fox online: Understanding and detection of mass redirect-script injections,"2014 IEEE Symposium on Security and Privacy, San Jose, CA

[5] B Patel, P Jay, S Rushabh. "A Review Paper on Comparison of SQL Performance Analyzer Tools: Apache JMeter and HP LoadRunner." (2014)