Improving Data Quality of Proxy Logs for Intrusion Detection (Poster Abstract)*

Hongzhou Sha[1,3], Tingwen Liu[2], Peng Qin[2,3], Yong Sun[2,3], and Qingyun Liu[2,3]

[1] Beijing University of Posts and Telecommunications, Beijing, China
[2] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[3] National Engineering Laboratory for Information Security Technologies, Beijing

1 Extended Abstract

Log correlation analysis plays an important role in many information security areas. For example, it can be used to help find abnormal navigation behaviors in inside threat detection. Besides, it can be used as the data source for intrusion detection [1]. However the original logs are filled with noises. Therefore, data cleaning is an indispensable preprocessing step in log correlation analysis in order to improve detection efficiency and reduce storage space.

Many methods have been proposed to improve data quality by removing irrelevant items such as jpeg, gif files or sound files and access generated by spider navigation. Most of them are designed for web servers (such as e-commerce web site). These methods work by inspecting the fields of user-agent, http status and URL suffix in web requests. However, they cannot be used to address the problem of improving data quality of proxy logs (recording web requests through intermediate roles) very well. Because proxy logs show different features compared with server logs. The biggest difference is that proxy logs should be cleaned without knowing the information of the web site accessed by a web request, such as its web structure and content type. It makes traditional data cleaning methods incapable of filtering specific noises in proxy logs, such as software updates and requests from network behavior analyzers. Moreover, proxy logs experience rapid growth of web requests that are generated by unlimited websites and users, which makes the problem more difficult to tackle.

In this paper, we start our work with the insight that automatic requests change more regularly with time than normal requests that users really want to trigger. To validate the insight, a statistical analysis is made on the accessed times for a given URL. It takes one day as a unit, and divide the day into multiple statistical periods. In order to facilitate comparison, the accessed times is divided into several statistical periods by the average accessed times in the day for a URL, referred to as relative accessed times. We observe the corresponding results of four consecutive days for the most frequently accessed URLs in the traffic of one backbone network access point in China. Among these results, two representative ones are shown in Fig. 1 and Fig. 2. One is scoreboard, which is generated by a network behavior analyzer automatically. The other is scholar,

* Supported by the National High-Tech Research and Development Plan 863 of China Grant No. 2011AA010703.

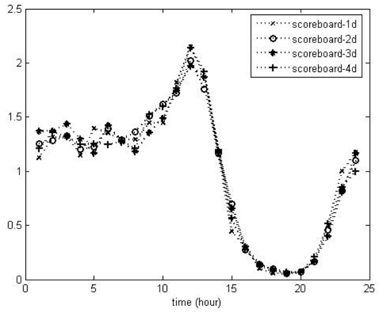

Fig. 1. Comparison of relative accessed times among four days on scoreboard

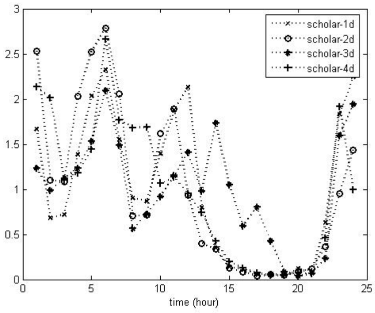

Fig. 2. Comparison of relative accessed times among four days on scholar

which is annotated in accordance with three major academic sites. Obviously, scoreboard belongs to typical automatic requests while scholar belongs to typical normal requests. From these two figures, it can be found that the relative accessed times of scoreboard are similar in different days, on the contrary the result of scholar is much more complex and the characteristic is not too obviously.

In this paper, we evaluate our work with a real traffic trace from a backbone network. There are 304,577 URLs accessed by 249 million times in total. The most accessed 500 URLs which are accessed by 35.1 million requests are taken as our experimental data, and label each request by analyzing the URL manually.

Firstly, LODAP [2] is used to filter out some irrelevant items. Then we introduce a method named FMTC to filter the remaining irrelevant items. For each URL, if the similarity between its historical data and new arrived data is larger than a predefined threshold k, the URL is considered to be triggered automatically, and should be filtered out. In this paper, Cosine Distance and Euclidean Distance is used to measure the similarity between the two data sets. Each set consists of the relative accessed times of all periods in a day cycle for every URL.

It can be found that increasing k will increase precision rate while decrease filtering rate and recall rate. When k is 0.485, FMTC method can achieve 83.13% filtering rate at the cost of 0.8% wrong filtration. This implies that FMTC is effective in improving the quality of proxy logs.

Although the experimental results may not be conclusive, as the traffic trace and experimental data used are limited and private, the preliminary results are very encouraging. In the future, we plan to capture traffics from more network links and label more requests.

References

1. Chu, J., Ge, Z., Huber, R., Ji, P., Yates, J., Yu, Y.C.: ALERT-ID: Analyze Logs of the Network Element in Real Time for Intrusion Detection. In: Proc. RAID. (2012) 294–313
2. Castellano, G., Fanelli, A., Torsello, M.: LODAP: A Log Data Preprocessor for Mining Web Browsing Patterns. In: Proc. WSEAS. (2007) 12–17