www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 4 Issue 9 Sep 2015, Page No. 14417-14420

A Platform Independent Log Analyzer with Searched Keywords Reporting Feature

Krishna Kumar Lahoti[1], Jitendra Kumawat[2], Preeti Gupta[3]

[1]Department of Computer Science and Engineering,
Amity School of Engineering and Technology, Amity University Rajasthan,
NH-11c, Jaipur, Rajasthan, India.
klahoti76@gmail.com

[2]Department of Computer Science and Engineering,
Amity School of Engineering and Technology, Amity University Rajasthan,
NH-11c, Jaipur, Rajasthan, India.
jkumawat@jpr.amity.edu

[3]Department of Computer Science and Engineering,
Amity School of Engineering and Technology, Amity University Rajasthan,
NH-11c, Jaipur, Rajasthan, India.
preeti_i@rediffmail.com

Abstract: ***Analyzing logs of a network can help us extract important information about users and the usage of network bandwidth. This information can then be used by the network administrator to take necessary administrative actions in order to enhance the security of network. There are several log analyzers available in the market, but have limited features or are platform dependent. In this paper, an efficient and platform independent log analyzer is proposed. This log analyzer along with all essential features regarding users, traffic and bandwidth usage, also has inbuilt features for search engine analysis and contains searched keywords and phrases in its reports. These reports give a easy-to-understand view of bandwidth usage by different users and also the type of content the users are accessing. The searched keywords and phrases report gives an idea about what is being searched on the network. The network administrator, who has the ability to filter users and traffic on the network can then take necessary steps to ensure that the network security policy of the organization remains intact. This also gives the network administrator flexibility, in taking decisions regarding maintaining the security of network. This log analyzer also overcomes issues faced in other tools by procuring reduced memory consumption and by being platform independent.***

Keywords: Network Security, Network Administrator, Log Analysis, Information, Searched Keywords.

1. Introduction

The logs of a network have copious amounts of information regarding its users and traffic [1], [2]. A log analyzer extracts critical information about users, bandwidth usage and traffic on the network which may then be used by the network administrator to enhance the security of the network by taking necessary preventive steps. To analyze logs from multiple users on a network, browser logs cannot be used, as it would be inefficient and time consuming. An alternative to this, is to use logs from a proxy server. Squid is a commonly used caching proxy server which also maintains logs of its users [3], [4]. It's configuration can be changed to adapt to the network and allow or deny specific local networks, users and traffic. The logs can be formatted to accommodate distinct information about users and their activities on the network. There are five different inbuilt log formats namely native, common, combined, user-agent and referrer which allow us to view multitudinous information about users and bandwidth. There is also a provision in Squid to have a custom log format based on our requirements. This log format ***access.log*** can then be parsed to extract important information to enhance the security of network.

After analyzing different existing log analyzers which provide vital information about the users and the bandwidth usage [5]-[7], there are eleven fields which have to be incorporated in the logs so that the network administrator has all the information necessary to get an idea about the users and the bandwidth utilization of the network. This information may then be used to improve the security of the network by taking necessary measures. The ***access.log*** file can be parsed through a program built on a platform independent language such as Java, with HTML template as output. The first output file may contain the analysis of relation between users and their accessed sites along with size in bytes and content type, to get an idea about the bandwidth usage by different users. The Parser program also contains search engine analysis features, with searched keywords and their count as output in another file. Graphics can be used to show the relation between the accessed sites and the number of hits generated by a particular site. The searched keywords can also be displayed graphically, using the relation between the searched keywords and their count. These reports give the network administrator decisive information about users and their individual bandwidth consumption, accessed websites, type of content and what they are searching for on the network. Based on this knowledge, the network administrator can filter users and traffic on the network as per the security policy of the organization.

2. The Log Parser

DOI: 10.18535/ijecs/v4i9.61

The first step towards building a log parser, is to make sure that the logs contain the required information in a specific format which can be parsed to get all the necessary details in the output files. Thus, there is a need for a tool which maintains logs of the network and which also has features to configure format of the log file as per the requirement. The following sections describe such a tool and how a parser can be built to analyze logs of the network. The flowchart given below summarizes the whole process of building such a log parser, on a platform independent language.

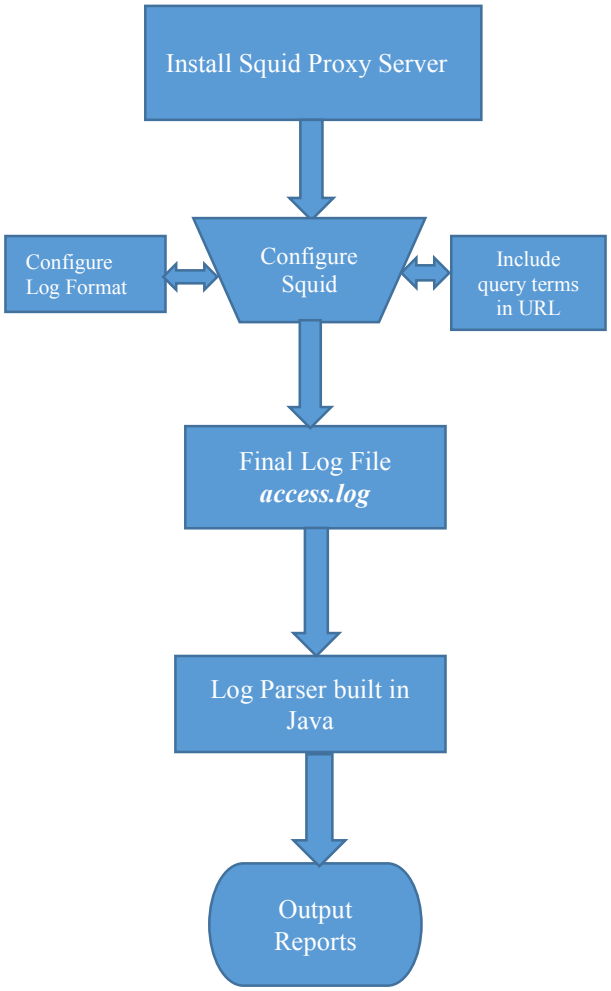

Figure 1: The process of building the log analyzer

2.1 Installing and Configuring Squid

Squid is a free caching proxy server for the web. It supports HTTP, HTTPS, FTP etc. and also maintains logs of its users. It possesses a variety of features such as Access Control, Authentication Control, Network Options and Log File Options. It can be installed as a package from the console of a Linux based operating system and can also be downloaded from the web for other systems. The configuration of Squid can be edited to meet the requirements of the network administrator [8]. The configuration file ***squid.conf*** which is located in the system's ***/etc/squid3/*** directory, can be edited in any text editor available. The configuration directive ***http_access*** allows us to manage the access control configuration. The Access Control Lists (ACLs) have to be carefully configured to allow or deny specific networks and users based on the requirement of network administrator. Squid normally listens to TCP/UDP traffic on the port 3128, but with the directive ***http_port*** in the configuration file, other ports can also be added. When the proxy settings of the browser are changed to access the web through Squid, only the ports specified in the configuration file are given access. There is also a provision for authentication control, in which the network administrator can set a password to run and manage Squid. The log format can be configured by incorporating the required fields in the ***logformat*** directive. In this paper, the eleven fields incorporated in the log format ***access.log*** to be parsed are local date and time, GMT, duration of transaction, client IP address, Squid result codes, size of the reply sent including HTTP headers (in bytes), request method, requested URL, user id, hierarchy code and content type. In the default configuration of Squid, the query terms are not included in the log files. To include searched queries in the URL field of the logs, the configuration directive ***strip_query_terms*** is to be turned off in the ***squid.conf*** file. The **access.log** contains information about every incoming & outgoing request & reply. Each line in the log file corresponds to one request or reply. As the data in these log files keeps increasing as the web is accessed, it is possible that the log file becomes too large in size. The configuration directive ***logfile_rotate*** allows the network administrator to rotate log files by keeping the new entries in the main log file to be parsed and the old logs in other files with numbered extensions. Alternatively, a ***cron*** job can be used to do the same on a Linux machine. This completes the process of configuring Squid as per the network administrator's requirement. The proxy settings of the browser have to be changed to manual proxy configuration, to access the web through Squid. The users on the network, have to enter the proxy name or IP address and the port number provided by the network administrator to be able to access the web.

2.2 Programming The Parser

The parser has to be built on a platform independent language such as Java so that it is flexible to be operational on any operating system. The program asks for the input log file from the network administrator in the run-time and functions only if the log file ***access.log*** contains the eleven fields in specific order as dictated in the above section. The permission settings of the ***access.log*** should be set such that it is readable by the program. HTML templates can be used for the output reports. The parsing program also asks for the location to store the output HTML files. The fields in the input log file are tokenized in the program, so that they are displayed in a table under the corresponding columns named **User-IP**, **Date, Time, GMT, Requested URL, Bytes** and **Content type** in the first output file**.** The number of hits is calculated as the total number entries in the log file which do not have ***TCP_DENIED or UDP_INVALID*** as their Squid result code. Moreover, only these entries are considered for the table in the first output file. The number of page views is calculated as the total number of hits from the log file which have ***GET*** or ***POST*** as the request method and also which do not have ***json, javascript*** etc**.** in their content type. The searched keywords report can be obtained by using the URL field in the log file. The parser should be programmed to identify a list of HTTP based searched engines and extract the searched keywords or phrases by tokenizing the URL field after certain specific characters. In this paper, the search engine Bing is taken as a reference and its searched queries are considered for parsing. A list of HTTP based search engines is available on the web [9] which can be added to the program to include their corresponding searched queries in the searched keywords report. Additionally, the count for such searched keywords can also be maintained and displayed in the report next to the specific keyword. For graphical reports, web based APIs can be used to display the relation between the websites and the

number of hits for a particular websites. This gives an easy understanding of which sites are accessed the most and the amount of bandwidth utilized by a particular website. Graphics can also be used to show how many times a particular keyword was searched on the network. In this manner, the parser can be programmed to parse the log file and display the analysis and searched keywords of HTTP based search engines in its output reports.

2.3 Executing the Parser

When the parser program is executed, it asks for the location of input file from the network administrator and also the location where he/she wants to store the output files. The input file is then read and executed only if it has the eleven fields in the specific order as mentioned in the above sections. Based on this input file, the program parses it and extracts information about users and traffic on the network. The first output file contains the number of hits and the the number of page views. It also contains a table, depicting the relation between users and their accessed sites. The second output file contains the keywords or phrases searched on the network and the number of times a particular keyword has been searched. The third and fourth output files contain graphical reports and is only viewable if the system has access to the web, as these reports are based on web based APIs. These reports are a representation of the hits generated by a particular website and a particular keyword or phrase.

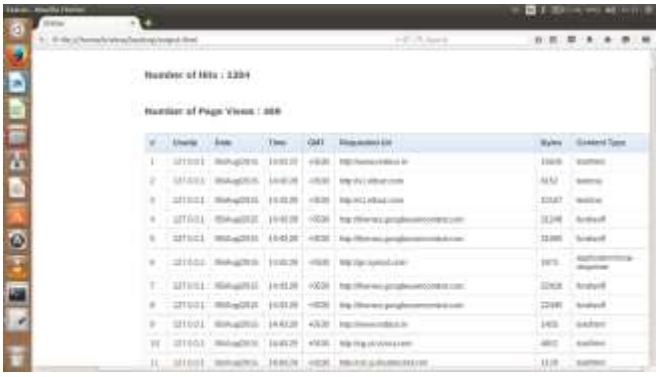

Figure 2: Log Analysis Report

Figure 3: Searched Keywords and Phrases Report

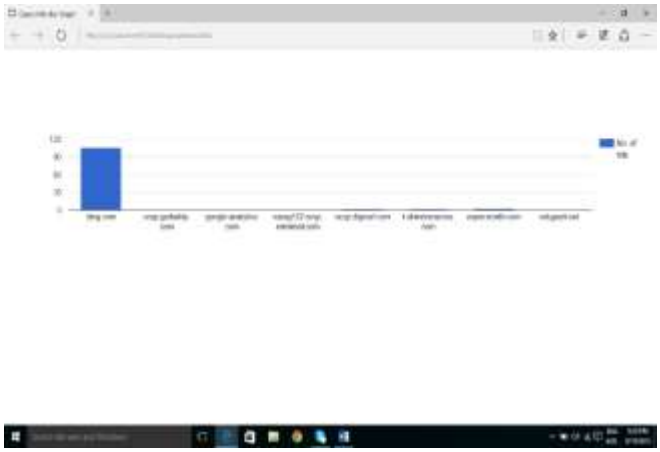

Figure 4: URL Hits Graph

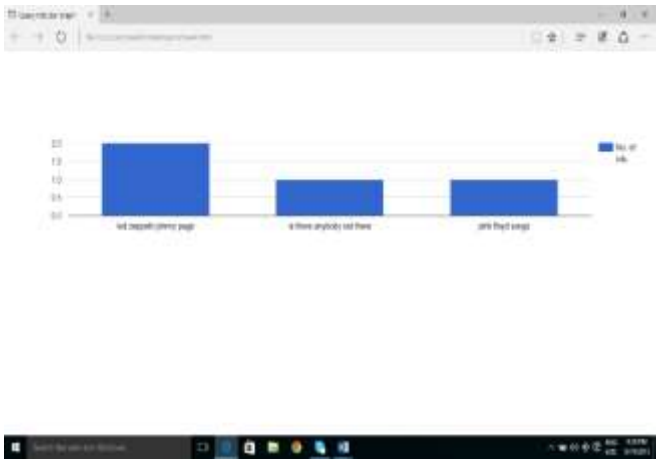

Figure 5: Query Hits Graph

3. Conclusions and Future Work

A log parser which extracts important details about users and traffic on the network is successfully created. It displays the analysis along with what keywords have been searched on the network, as output in HTML files. It is also platform independent and consumes about 40% less memory than other existing tools which do not even have searched keywords report. This log analyzer generates about 80MB of reports for every 1GB of throughput, when the searched keywords report and the graphical reports are also included in the output. The network administrator, after studying these reports, may then take necessary steps in order to maintain the security policy of the network.

This log parser can be enhanced to include the analysis of search engines based on HTTPS. For this, the proxy server Squid is to be built from scratch to include SSL and ICAP compile switches [10].This would allow Squid, to enable the ***ssl_bump*** feature in its configuration [11]. The proxy server then acts as man in the middle. Also, an SSL certificate is to be created which would notify the users on the network about the monitoring. The HTTPS requests are now considered as normal HTTP requests and allow Squid to log the queries of HTTPS based search engines such as Google and Yahoo. The parser program can be modified to include the list of search engines based on HTTPS and output their queries to the searched keywords report.

References

[1] James A. Pelletier and Tamraparni Dasu, "Mining Network Logs: Information Quality Challenges", IQ Conference 2005.

[2] Sweta Vinay Kamat, "Intrusion Detection System Using Data Mining" , International Journal of Advanced Research in Computer Science and Management Studies, Volume 2, Issue 6, June, 2014.

[3] The Squid Cache Website. [Online]. Available: http://www.squid-cache.org/

[4] Duane Wessels, *Squid: The Definitive Guide*, O'Rielly Publications, 2004.

[5] SARG homepage. [Online]. Available: http://www.sourceforge.net/projects/sarg/

[6] Light Squid homepage. [Online]. Available: http://www.lightsquid.sourceforge.net/

[7] Calamaris homepage. [Online]. Available: http://cord.de/calamaris-english/

[8] Squid Configuration webpage. [Online]. Available: http://www.squid-cache.org/Doc/config/

[9] List of Search Engines. [Online]. Available: http://www.en.wikipedia.org/wiki/List_of_search_engines

[10] Filtering HTTPS traffic with Squid. [Online]. Available: https://www.howtoforge.com/filtering-https-traffic-with-squid/

[11] Squid with ssl_bump. [Online]. Available: http://www.squid-cache.org/Doc/config/ssl_bump/

Author Profile

Krishna Kumar Lahoti is a research scholar in the Department of Computer Science and Engineering, Amity School of Engineering and Technology, Amity University Rajasthan. His areas of interest include Network Security, Information Security and Data Mining.