HTTP Logs Analysis & Its Use Cases - Survey

[1]Malika Makker, [2]Rohan Simha, [3]Dr. Anala M.R., [4]Revathi S.A.
[1] Undergraduate, [2]Undergraduate, [3]Assosciate Professor, [4] Assistant Professor
[1]Computer Science and Engineering,
[1]R.V College of Engineering, Bangalore, Karnataka, India

Abstract : World wide web is a huge platform for users to disseminate and consume information. Several web applications hosted on world wide web record their respective users activities through logging. Every web application maintains their web access logs or HTTP logs in the servers where they are being hosted - for example, Microsoft IIS Server, Apache server, etc. The potential of these logs to further secure the website or gain more visitors remains unexploited. HTTP logs have been analysed before for user profiling, analysis of traffic, monitoring for any kind of suspicious activities. User profiling helps in targeted advertisement or recommendations - generally carried out by analysis of user's click stream. Analysis of traffic helps in gaining an insight on the kinds of users - crawlers, bots, spiders, human users, human users looking for vulnerabilities in the system, etc. This analysis could be used to make the website more user specific and also to better the visibility to web crawlers. Analysis of suspicious activities could help gauge the vulnerabilities of a web application and thereby, make it more secure. This paper also enumerate the various kinds of attacks a website could be exposed and the ways to protect against them. Availability of various logs analysis frameworks has also been discussed.

Index Terms **– HTTP logs, traffic, anomaly detection, attacks, user profiling**

I. INTRODUCTION

Most of the web applications resort to maintaining their log records to monitor the access to their website**.** These logs have a general format consisting of timestamp of access, HTTP method, user's IP address, HTTP status code, etc. These logs carry sufficient information to analyse the minutes of traffic visiting a website and the pages they request for. These log records are generally maintained on the server hosting a web application. These logs have been analysed for securing the website, user profiling, analysing traffic patterns, etc.

A website can be attacked by an external attacker or even internal users with authorized accounts. There are a myriad of attacks which can be carried out and each of them generate different HTTP request methods. Understanding of the ways an attack can be carried out and filling those gaps, can be easily done by a deep analysis of HTTP logs. The most known attacks are SQL injections, cross-site scripting, etc. Each of them show certain patterns in HTTP logs [6]. Many experiments have been conducted to automate the process of detecting any suspicious activities - termed as 'anomaly detection' in HTTP logs. Many machine learning techniques have exploited to carry out anomaly detection on patterns of logs seen.

User profiling is one of the prominently deployed techniques to better the user experience when using a search engine or even exploring a web application like an e-commerce website. Search engines resort to user profiling for displaying more relevant results and even target advertisements based on their interests. E-commerce website again heavily rely on user profiling to carry out recommendations. User profiling is mostly carried out by analysing user click streams [4]. It helps in personalizing user experience.

Analysis of traffic for a particular web application can give many useful insights. Most importantly, it helps in determining the navigation profiles of users visiting a website. Navigation profiles includes bots, crawlers, spiders, regular users or malicious users. These users shows some trends in their navigation profiles which are very prominent [1],[3]. On identification of their existence, the temporal distribution of their requests can be plotted and how they might overload the servers.

The dataset used in [7] for intrusion detection was the DARPA 1999 dataset, consisting of forty-two parameters primariy consisting of the audit logs, internal and external sniffing data and dumps from some selected directories. Through this paper it was evident that the nature and the type of logs determine the kind of attack it can accurately detect. In [7], a fuzzy based rule system was developed to detect attacks. Whereas, in [8] the dataset was the web access logs representing HTTP requests received by NASA Kennedy Space Center Server located in Florida, this dataset was used to understand the clicks of the user and analyse the user's behaviour which was done efficiently in [8]. The primary fields in this dataset were the host, a timestamp, request, response code and the content size. Apart from the dataset and the methods used to analyse attacks and user profile, it is also necessary for any industrial grade product to store and collect logs in the most efficient way processing wise and also memory wise. This has been discussed in section VI.

Overall, this paper summarises the use cases of HTTP logs and how they can used in making a web application more secure and robust to incoming traffic.

II. TYPES OF ATTACKS

HTTP log records often contain some logs indicating an intrusion attempt or some sort vulnerabilities logged by a vulnerability scanner. To dig deep into how these attacks are carried out and their prevention mechanisms, we first need to understand the various types of attacks which can be carried out in the first. As in [6], we would briefly discuss the top threats enlisted by the Open Web Application Security Project (OWASP). The top threats identified were as follows:

2.1 Injections

Injections are carried out by executing SQL commands to insert malicious data into the database. These injections mostly use some stored procedure, which are pre-written SQL commands, requiring some data as input. Attackers look for these stored procedures which could execute SQL to shut down the server, make some changes in the registry, cause termination of some processes, etc. In [6], the HTTP logs of their portal showed some traces of SQL injections by merely trying to execute some stored procedures.

2.2 Cross-Site Scripting (XSS)

It's a type of attack of which exposes the vulnerability of a website accepting unvalidated inputs from the users and then executing those inputs elsewhere. Attackers use this glitch mostly to steal cookies of authenticated users. In [6], they found a couple of attempts of trying to execute javascript codes by XSS. However, since they had used some vulnerability scanner, like Nessus, they has some javascripts execution showing "Nessus was here", thereby exposing some unseen vulnerabilities.

2.3 Flaws in Authentication Mechanisms

Unencrypted URLs or disclosing authentication information in URLs, are some of the flaws in authentication mechanisms. Attackers could use this vulnerability to steal some sensitive information. These kind of attacks are hard to detect with HTTP logs.

2.4 Referencing Unauthorized Objects

This attack happens when some of the pages of the web application can be accessed without any authorization. An authorized user can tweak the URL a bit by trial and error or fuzzy testing, and gain access to a URL requiring no authorization. To avoid this error, it should be ensured that all pages require some form of authorization.

2.5 Cross-Site Request Forgery

This attack requires the attacker to perform an act of social engineering to lure the user into clicking a link which would perform an action which the user is not aware. Generally, the website stores the authorization credentials of the user, which would allow the attacker to carry out this attack. This process can only be stopped by CORS (Cross Origin Resource Sharing). However, if the attacker's website returns ACCESS-CONTROL-ALLOW-METHODS as *, even CORS would not be of much help.

2.6 Security Misconfigurations

This vulnerability can be misused by both attackers and internal users with authorized accounts. Security misconfigurations can be found at all levels of web application stack - UX, platform or server. The attackers generally tries to find "cmd.exe" or "root.exe" file to execute some command line prompt commands.

2.7 Access to Unauthorized Pages

This attack is similar to referencing unauthorized objects. Here, the attacker lands on a page of website and tries various random URLs by looking at the content of the website. If the attacker discovers a page by the random URLs, which requires no authorizations, could lead to serious consequences.

2.8 Invalidated Redirects to Other Websites

If a website constantly redirects the user to another webpage, the attacker could plant a malicious URL on the redirected website, which the user would not be aware of. This redirects can be easily spotted in HTTP logs.

III. Analysis Of Traffic

A lot of research has gone into analyzing the traffic of a webpage in understanding the kind of users visiting a website, the intensity of traffic generated by them and the temporal distribution of their requests.

3.1 Navigation Profiling

In [3], the users visiting a website have been attempted to be classified into three major profiles -

- Crawlers/Bots/Spiders
- Regular Users
- Crawlers with a malicious intent

The dataset used by them was HTTP logs by two distinct web servers over a period of one year. Some of their observations to distinguish crawlers from regular users are discussed below.

The crawlers can be identified easily as they initially request for "/robots.txt" file. This makes their presence easy to detect. Also, they found that of all the web crawlers that visited their websites, close to 80% of them were Google, Microsoft and Yahoo. They used several parameters to recognise their clients - one of them being inter-reference time. One of their exclusive observations was that the inter-reference time of crawlers is greater than that of regular users. In [3], the arrived at an average inter-session time of 240 sec for crawlers and 120 sec for regular users.

Most of the clients send out requests in sessions. A session lasts for some time where the server receives a series of requests - this lead to observing the inter-session time. The other parameters used for distinguishing between clients were - number of sessions and their duration per client and number of requests per session. Crawlers showed higher average session duration and higher average number of requests per session. Crawlers intended to identify themselves, by showing a behaviour to reduce their requests impact on the web servers. Crawlers send out a lot of requests for purposes of search engine indexing. To reduce their impact on the servers, they were seen to not send requests in bursts. They were seen to visit very often, at an average after 31 minutes. Another observation was that the each of the crawlers visit at least thrice which is a behaviour seen in even regular clients. However, malicious crawlers with an intention to not be identified, don't even send out a request for "/robots.txt".

In [1], they intend the classify user visiting their web application into three major profiles -

- Crawlers/Bots/Spiders
- Regular User
- Scanners by malicious users before an attack

It's been observed that detecting a scan before an attack is easier than detecting an attack itself [1]. The above mentioned user profiles were categorized into type-1, type-2 and type-3 respectively.

The author of [1] categorically mentions that the method used for identifying user profiles is a rule-based detection method. Author of [1] has stated following as some of the reason for not using machine learning algorithms for detection of user profiles:

1. To obtain highly accurate machine learning models, a large set of training data is required - which either makes the model untrainable or the training phase lasts for days together. An additional toll on memory consumption entails.
2. A highly accurate machine learning model would also result in overfitting. Whereas rule based detection would be based on past user data, and would not be a very complex method.

The very first rule is for elimination of SQL injections and XSS attacks. These two attacks are identified in following ways:

1. SQL injections are detected by carrying out a search for certain SQL commands characters such as *',==,#,exec(),* etc.
2. XSS attacks are carrying out inserting some scripts, therefore a search is carried out for HTML tags like *<src>,<img>*,etc.

In the second step, an attempt is made separate type-1 from type-2 based on IP addresses. All the well-known web crawlers have their IP addresses publicly available. Even if this data is not up-to-date following rules can be used to distinguish:

1. Higher frequencies of '4xx' requests
2. request for "/robots.txt" file
3. higher rates of unmentioned request originators

This segregates type-1 from types-2. To separate type-3 from others, following rules were established:

1. The request HTTP methods were very peculiar such as Netsparker, Track, Propfind, etc.
2. Higher frequency of 404 requests
3. User-agents in header field was compared with that of well-known vulnerability scanners
4. More than 100 HTTP requests at a point of time

The accuracy of rule-based detection proposed by [1] was 99.38%.

3.2 Automated Log Analyzer Tools

Usage of HTTP logs analysis tools is booming as it helps in getting some real-time metrics on the visitors visiting a web application. There are several web logs analysis tools available online, which could be compared to suit user's needs [5].

In [5], features offered by various logs analysis tools like Google Analytics, Web Logs Expert, etc. have been discusses. Some of the metrics which can be gauged using these tools are:

1. Activity Statistics - Provides an insight on the number of visitors per hour/day
2. Access Statistics - It gives visualisation for a page of website in terms of number of visitors and the days of visits - this could be used to better organise the pages of a website
3. General Statistics - Total hits, total bandwidth, etc.
4. Visitors - Capable of enlisting the IP address of the user who access the website, along with number of hits of that user
5. Browsers - Gives a pie chart visualisation of the percentage of users using a particular browser, this could help in making the website better compatible with a particular browser
6. Errors - Gives a pie chart representation of the type of HTTP status code error encountered by a percentage of users

IV. ANOMALY DETECTION TECHNIQUES

Automating detection of any kind of intrusion by an attacker into server of a web application is a real boon. Its virtually impossible for anyone to manually look for anomalies through terabytes of log records generated by a web application. In [2], a method to deal with high dimensionality of log records data has been proposed.

The method proposed first carries out features extraction to build a feature matrix. Feature extraction is done by carrying out 2-gram words extraction from raw log files. The obtained feature matrix is given for dimensionality reduction. There are three techniques of dimensionality reduction that have been compared in [2] are random projection, diffusion maps and principal component analysis.

The training data used was HTTP access logs from various web servers and intrusions were manually added. After feature extraction and dimensionality reduction, a mean point and the average distance from the mean point is calculated using all the training data. To detect an intrusion, the new feature set is dimensionally reduced and its distance from the mean point is calculated. If the distance is found to be greater than average distance, it is deemed to be an intrusion.

Random projection could analyse large amounts of data in the least amount of time; diffusion maps offered the most accurate results. Principal component analysis didn't seem to offer any exclusive advantages.

V. USER PROFILING

User profiling is one of the ways to automate the understanding of user's interests and provide relevant results to the user. It helps in building personalized applications and enriching user experience. Applications include recommender systems, search engines, etc.

User profiling can be carried out in the same way as topic modelling [4]. Here, documents for topic modelling would be web access logs. LDA topic modelling can be used for user profiling [4]. The dataset used was click streams of 7500 students giving 40 GB of data.

For word abstraction, user's click stream was divided into sessions based on some predefined threshold. For all the requests in a session, URL abstraction was carried out. For example, a URL "https://example.com/abc.html" would be reduced to "https://example.com/". This abstracted URL would be then matched to a broader concept - termed as Cross-Hierarchical Directory Matching(CHDM). For CHDM, an open directory like Yahoo! Directory could be used [4].

The LDA modelling gave as output the most popular topics which were common to more than 1% of students. The author of [4], then gave some unique categories to 24 major topics obtained. Some examples have been shown in table 5.1

Table 5.1: Exampled for LDA topic modelling

Topic	Category
Job Search Portals, Recruitment Tests	Job Hunting
C-language tutorials, Python tutorials	Technology Oriented

VI. LOGGING FRAMEWORKS

A resilient, robust logging framework is always expected in an industrial grade application, to provide the clients a sense of transparency and as a fall back to administrators whenever the automated system fails. A framework is complete end to end system starting from aggregation of logs and presenting verbose log lines in a compact format and presenting it in a concise manner with automated actions during issues. In [7], the log aggregation starts off with division of packets into two categories i) s-groups, a fixed number of packets, ii) t-groups, a set of packets within a defined time interval. Upon this is a running apriori algorithm which maintains and defines the rules of each feature. The primary objective of this work was to find the minimal set of features that help in identifying an anomaly. A sudden change in any of the feature values results in the rules being broken and an anomaly is signalled. Mamdani inference was used to infer the threshold for the rules. A membership of BELOW, AVERAGE and ABOVE were defined for the five primary attributes ICMP, UDP, TCP, SYN and FIN.

In [8], the authors proposed a resilient high speed analysis platform for web logs using a combination of kafka and Apache Spark streaming modules. Kafka fetches the web logs and sends it over to an Apache stream processor consisting of two components - Spark Streaming core and a Spark Engine Core. The former takes care of the log collection across defined intervals which is controlled by two variables - WINDOW_INTERVAL and the SLIDE_INTERVAL. Whereas, the latter takes care of generating, processing and distributing the RDDs (Resilient Data Distribution), an entity of spark which enables it handle and serve data of large volumes. After the processing, the data is sent into storage to house the necessary takeaway from the logs in a very concise format and also a visualization dashboard which is crucial to clients, to see the logs in the form of graphs rather than verbose text.

This framework in [8] is very simplistic and straightforward, any large number of hits from an unidentified source would be treated as an anomaly. This framework is also used to identify broken links, a broken link is identified when a 404 error is returned as the status code. A further proposed prospect of this paper is to look on the search trail of a user based on the IP (websites visited) and suggesting similar websites based on the assessed user browsing pattern.

VII. CONCLUSION

Through this survey, we see a huge scope for the usage of HTTP logs. Not only do they offer daily traffic statistics, but with application of a good algorithm, unthinkable insights can be drawn. Apart from drawing insights, they can be used to discover loopholes in the web application design and hence, help in making them more secure. With navigation profiling, the web application can be made more compatible for all kinds of clients.

However, plainly relying on HTTP logs for all kinds of analysis may not be a great idea. HTTP logs have certain drawbacks, such as, lack of request and response headers, which limits the capabilities of HTTP logs to give the best analysis. There are certain injections which can be carried out with a POST request. Since certain parameters of a POST request are not logged, it becomes difficult to detect an injection. There is a lot of research which has been carried out on using execution logs to detect anomalies. Using both HTTP logs and execution logs, could be a holistic approach for application of logs in the understanding of running of a web application.

VIII. REFERENCES

[1] Baş Seyyar, M., Çatak, F. Ö., & Gül, E. 2018. Detection of attack-targeted scans from the Apache HTTP Server access logs". Applied Computing and Informatics, 14(1), 28–36

[2] Juvonen, A., Sipola, T., & Hämäläinen, T. 2015. Online anomaly detection using dimensionality reduction techniques for HTTP log analysis. Computer Networks, 91, 46–56

[3] Calzarossa, M. C., & Massari, L. 2011. Analysis of Web Logs: Challenges and Findings. Lecture Notes in Computer Science, 227–239

[4] Fujimoto, H., Etoh, M., Kinno, A., & Akinaga, Y. 2011. Topic Analysis of Web User Behavior Using LDA Model on Proxy Logs. Lecture Notes in Computer Science, 525–536

[5] Goel, Neha, and C. K. Jha. 2013. Analyzing users behavior from web access logs using automated log analyzer tool. International Journal of Computer Applications 62.2

[6] Santos, R. D. C., Grégio, A. R. A., Raddick, J., Vattki, V., & Szalay, A. 2012. Analysis of web-related threats in ten years of logs from a scientific portal. Cyber Sensing.

[7] Shanmugam, B., & Idris, N. B. 2009. Improved Intrusion Detection System Using Fuzzy Logic for Detecting Anamoly and Misuse Type of Attacks. 2009 International Conference of Soft Computing and Pattern Recognition

[8] Agarwal, S., & Prasad, B. R. 2015. High speed streaming data analysis of web generated log streams. IEEE 10th International Conference on Industrial and Information Systems (ICIIS)

[9] Hassani, Mehran, Weiyi Shang, Emad Shihab, and Nikolaos Tsantalis. Studying and detecting log-related issues. Empirical Software Engineering 23, no. 6: 3248-3280

[10] Durga Choudhary, Subhash Chandra Jat, Pankaj Kumar Sharma. 2016. Adaptive Query Recommendation Techniques for Log Files Mining to Analysis User's Session Pattern. International Journal of Computer Applications (0975 – 8887) Volume 133 – No.17

[11] Farshchi, Mostafa, et al. 2018. "Metric selection and anomaly detection for cloud operations using log and metric correlation analysis." Journal of Systems and Software 137: 531-549

[12] Dietz, Marietheres, and Günther Pernul. 2018. "Big Log Data Stream Processing: Adapting an Anomaly Detection Technique." International Conference on Database and Expert Systems Applications. Springer, Cham

[13] Miranskyy, Andriy, et al. "Operational-log analysis for big data systems: Challenges and solutions." IEEE Software 33.2 (2016): 52-59.