A Comprehensive Review of Anomaly Detection in Web Logs

Mehryar Majd, Pejman Najafi, Seyed Ali Alhosseini, Feng Cheng, Christoph Meinel
Hasso Plattner Institute (HPI), University of Potsdam
Potsdam, Germany
{firstname.lastname}@hpi.de

Abstract—Anomaly detection is a significant problem that has been researched within diverse research areas and application domains, especially in the area of web-based internet services or cybersecurity. Many anomaly detection techniques have been developed for specific application domains, while others are more generic. The Log files of Web-server give insight into the state of web-server and applications running on it and enable the detection of abnormal incidents or behavior. This paper focuses on particularly web-server HTTP logs to the problems of Web-server Log Anomaly Detection (WLAD) due to their own nature and features and aims to provide a brief review of different Data-driven techniques to get to the bottom of recent studies and developments made in the context of WLAD. Moreover, in this paper, the literature related to webserver logs analysis, as well as other closely related to the WLAD topic, are taken into consideration for review. We have classified existing techniques into different categories based on the underlying approach adopted. When applying a particular technique, these assumptions can be used as guidelines to assess the method’s effectiveness in this area. We also provide a basic security anomaly detection approach for each category and compare the existing methods as variants of the basic technique. Further, we identify the cons and pros of the current practices for each category. We also discuss the computational complexity of the methods, which is an essential issue in the domain of Big Data.

Index Terms—Anomaly detection, Web-server log analysis, Malicious HTTP, Web-server attack, Cybersecurity

I. INTRODUCTION

Web applications have been the most used internet-based services in current business practice, and various application services including computer resources, e.g., Web server, web-app server, and storage using web-based applications over the internet. Therefore, the usage of web applications is inevitable nowadays. Therefore service providers are always interested in monitoring malicious activities to manage their business processes. Recently, the rising number of security events and substantial economic losses, mainly due to Covid-19 outbreaks, warns us that the detection of malicious traffic has become a great challenge for both individuals and enterprises to protect their digital assets and services/applications. Intrusion detection systems (IDS) and Fraud Detection systems (FDS) play a critical role in any organization’s network security strategy as well as the lifeblood of network monitoring.

Traditionally, IDSs have two central schemes for webserver attacks: signature-based intrusion detection, known as misuse detection [6], [7], and Behavior-based solutions, also known as anomaly-based threat detection or anomaly detection [8]. Misuse detection can only detect known attacks and filter HTTP requests based on a predefined rule or pattern. Most Web Application Firewalls (WAF) use misuse detection methods, but the malicious request keywords are replaced or encoded multiple times, which can bypass WAF [9], [10]. Anomaly detection use deviation of normal data pattern among
activities that do not conform to normal/expected behavior. It can detect unknown and new attacks by creating a model of "normal" use.

Web logs collected from the web-servers (e.g., Apache HTTP Server), proxy servers or reverse proxy servers (e.g., Nginx Web Server) are a valuable source of information due to their characteristics discussed in subsection II-A, allowing one to find traces of possible attacks. Thus, Auditing and reviewing the logs from web servers frequently by developing the smart pipeline and improving threat intelligence can effectively establish a comprehensive security posture. In this regard, log analysis can help not only optimize or debug system performance but also detect malicious activities that have managed to bypass the majority of the IDS solutions. Anomalies in web-server logs are the patterns in data that deviate significantly from the expected logs. This deviation could indicate deviation from normal behavior that could be considered malicious [11].

Table I. Examples of web-server logs highlighting some attacks and exploitation of injections (e.g., XSS, SQLi, Log4j), where the attacker is attempting to manipulate the headers such as user agent (UA), referrer, and HTTP method. The last example is a representation of invalid characters (mostly unintentionally) in API within web-request possibly by the human factor to reach the end-point/service.

<table>
<thead>
<tr>
<th>No.</th>
<th>Suspicious log event examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&quot;GET /&lt;script&gt;PAYLOAD_INJECTED&lt;/script&gt; HTTP/1.1&quot; 403 &quot;UA&quot;</td>
</tr>
<tr>
<td>2</td>
<td>&quot;GET HTTP/1.1 403 - &quot;Sndj://a/pAYLOAD_INJECTED&quot;</td>
</tr>
<tr>
<td>3</td>
<td>&quot;GET HTTP/1.1 403 - &quot;1234&quot;&amp;&quot;IP&quot; + PAYLOAD_INJECTED &quot;</td>
</tr>
<tr>
<td>4</td>
<td>&quot;GET SPAYLOAD_INJECTED + /windows/win.ini HTTP/1.1&quot; 404 &quot;UA&quot;</td>
</tr>
<tr>
<td>5</td>
<td>&quot;GET XXX/HPI/HE/XXX/ HTTP/1.1&quot; 404 &quot;UA&quot;</td>
</tr>
</tbody>
</table>

The main contributions of this paper are summarised below:

- Providing necessary background knowledge and identifying requirements/motivations (Section I) and challenges for data-driven-based web log analysis. (Section II).
- Review and categorize available analytical approaches, applications, and use-cases. (Section III).
- Discussing the open questions and challenges and point out several valid key research issues for WLAD (Section IV).

II. BACKGROUND

A. Characteristics of Web server log

Mainly, web servers deploy the CLF (Common Log Format) for their server log files; there are other following types in which data can be separated into different logs (instead of being combined into a single file), as it is demonstrated in taxonomy in Fig. 2.

Every request is sent over the resource/website (pages, APIs) to the web server, and the response is sent back to the requester. Meanwhile, a log message is stored or logged in the web server log file, maintaining a page request history.

Every web-app service has a Uniform Resource Locator (URL), which could be created either after clicking a link on a web page, bookmark/email, or by entering the URL directly into the address bar. Then the protocol is connected to the domain name, and the domain name is connected to the file path. In RESTful systems, clients send requests to retrieve or modify resources, and servers send responses to these requests. The anatomy of the full web server log includes information about the client IP address, request date/time, requested page, HTTP code, HTTP method/verb, bytes served, user agent (UA), and referrer. The General Anatomy of the URL mainly consists of protocol, the domain name, the file path, and URL parameters/query strings within the RESTful API query. The anatomy of full Web server logs, as well as general URL, are shown in Fig. 3 and Fig. 4, respectively.
A. Underlying Detection Techniques

1) Rule-based Models: Despite the fact that rule-based systems are effective in detecting the known anomalies for which rules are given, they fail to adapt to the evolving network environment. Moreover, the detection rate of anomalies using different rule-based approaches depends heavily on network administrators’ expertise and domain knowledge about the known attacks [12]. Due to the fact that unknown attacks often have no designed rules by security experts, they may remain undetected by such systems [13]. Therefore to cover this shortcoming of rule-based systems, various machine learning-based algorithms have been proposed in recent years for Log-based Anomaly Detection [14, 15, 16, 17, 18, 19, 20, 21].

2) Statistical Models: In earlier years, some statistical techniques were utilized to develop WLAD systems. [22] proposed a quarantine service system that generates signature queries (using Regex [23]) to detect anomalous HTTP queries received from malicious users in log files of the web server. He offered two methods to analyze the value of the query parameters based on the length and character distribution to serve the detection model.

3) Supervised and Unsupervised ML Models: Supervised learning methods rely on tags in which data has been labeled, while unsupervised learning methods are based on clustering [24, 25] and invariant mining. Supervised learning has a very good effect in detecting known malicious behavior or abnormal state, but it cannot detect unknown attacks, as it depends on prior knowledge. The unsupervised method can be used instead to detect unknown exceptions, but most of the methods need to improve their accuracy in the absence of label/ground truth information or anomaly rate within data for WLAD.

Unsupervised techniques can be used to uncover hidden structures, like finding groups of events with similar patterns, but it’s difficult to implement and is not used as supervised learning. Zhao et al. proposed a novel feature-extracting mechanism, including log-line tokenization using an LSTM-based anomaly detection approach to identify attacks on two widely-used datasets for device access log and net access log. They showed that their proposed model outperforms one-class SVM, GMM, and Principal Components Analysis (PCA).[26] Apart from many studies for time-series analysis for clustering that have been researched so far, [24] summarize recent studies on log data utilizing clustering techniques that have been proposed lately. Earlier they researched string clustering [25] and proposed a dynamic log file anomaly detection methodology that incrementally groups log lines within time windows.

4) Deep Hybrid Models (DHM): Typically most of the approaches cannot handle unknown log types without taking advantage of the log semantic information. So most existing web log-based anomaly detection methods use a log parser to get log event indexes or event templates and then utilize machine learning methods to detect anomalies. Lv et al. proposed ConAnomaly, a log-based anomaly detection model composed of a log sequence encoder (log2vec) based on the Word2vec model and multi-layer Long Short Term Memory Network (LSTM) [27]. That captures semantic information in the log but also leverages log sequential relationships. Another approach proposed by Liu et al. is a heterogeneous graph embedding-based modularized method that converts log entries into a heterogeneous graph [28]. Wang et al. proposed an offline feature extraction model so-called LogEvent2vec, which takes the log event as input of word2vec to extract the relevance between log events and vectorize log events directly. [29, 30, 31] Also, Laskar et al. proposed an approach that combines the Isolation Forest with the k-Means algorithm, so-called the iForest-KMeans model for anomaly detection to detect anomalies in Big-data.[13] The most recent study [32] shows promising results for WLAD, passing NLP-tokenized feature vectors to the Tree-based EAD model, so-called ELSV.

B. Feature Representation

Mainly, the ML-based approaches for WLAD are very similar to malicious URL detection (MUD), which comprise two steps: first, to obtain lexical and host-based feature selection, and second, to use these features for training the predictive model to detect malicious URLs [33]. Also, we should take care of external dependency to acquire information, the associated time cost concerning feature collection and feature preprocessing, and the dimensionality of the features obtained. Lexical features are very efficient to collect, as they are basically direct derivatives of the URL string within web request (subject to web server log), but they have some limitations like failing to efficiently detect new words (unseen features) due to lack of a knowledge-based system (KBS) augmentation to examine web request or URL [33]. Another point is mostly they are high-dimensional because they are all stored as Bag-of-Words features and feature size consequently affects the training and test-time. Despite NLP-based tokenization and semantic embedding, the most recent studies researched a different combination to feed their model, such as Byte-Pair Encoding (BPE) [34]. Fig. 5 shows anomaly detection approaches in related work based on major components (Model and Feature Map).

Fig. 5. Anomaly detection approaches arranged in the plane spanned by two major components (Model and Feature Map) point of view distinguished with three main groups of approaches (Classification, Probabilistic, and Reconstruction) that all formulate Shallow and Deep models. These three groups are complemented by purely distance-based methods for shortlisted related works in Table II. The Decision Tree (DT) and its successors, such as iForest (IF) and RandomForest (RF), are invariant to the choice of distance metric (since they do not use a distance metric in the first place), but randomized splitting/partitioning instances recursively [35].
C. Applications Scenarios

Finding anomalous incidents or logs in the logged data that might represent malicious activity has a wide range of applications in cybersecurity and network security, with common characteristics that are interesting to the analyst. There are some use-cases that explored user behaviour analytics, and user profiling [36] for User Behaviour Anomaly (UBA) detection [37, 38, 39], a behavioural IDS [40], or Malicious Behavior Detection [41]. Additionally, Time-series Analysis (TSA) use-cases for performance and optimization of web-server/web-dispatcher [42, 43].

IV. DISCUSSION AND OPEN QUESTIONS

A. Outlierness Vs. Maliciousness

One of the important key points while using outlier detection methods in real-life cybersecurity application problems is understanding what outlierness is. Often outlier detection algorithms are used to identify outliers based on outlier score, which assigns an abnormal score by the algorithm to each data instance to indicate the degree of outlierness of each measurement, and it is parametric-dependent. So it means in the absence of data anomaly rate (sometimes so-called contamination ratio), often needs to set an anomalous threshold as one of the algorithm’s hyperparameters. The used model could predict the different values as anomalous/outlier with various configurations with respect to the model generalization condition and cause a considerable amount of false-positive due to the quality of the anomaly detection process. In practice, the characteristics of security-related data traffic are typically non-stationary, which means traffic distributions can vary over time [55]. Also, the superiority of adaptive thresholds over static in cybersecurity has been researched [56]. Thus better to consider Self-Adaptive Threshold [57] approaches within (self-)Adaptive Anomaly Detection [58] for WLAD problems that regularly adapt to newer log patterns to ensure accurate anomaly detection.

On the other hand, it is hard to characterize maliciousness, and harder yet to develop a maliciousness index to apply to models, and there is a research gap for assessment and measurement metrics for cybersecurity-related maliciousness [59].

Bad entries problems can be a good example of this issue, considering there could be invalid characters within a web request as it is shown in Table I and Extract, Transform, and Load (ETL) tools parsed data differently (depending on ETL configuration or possible bugs in ETL pipeline), which could affect on detection algorithm performance. These human factor outliers may generate due to unwanted typo mistakes unintentionally, making detecting malicious incidents challenging in micro-level clustering.

B. Feature Engineering (FE) & Contextualization

Web server logs and Web log data analysis are important in intrusion detection, and different ML techniques have been assigned for abnormal detection. However, compared to abundant research on machine learning, ways to extract features from log data are still under research. Considering the text-based nature of web server logs, applying NLP approaches helps to provide better feature representation for learning models [60]. This approach to various downstream machine learning algorithms has been applied and proved its usefulness [47]. In the absence of labeled web-server log data, limited research has been done over out-of-date public datasets with noticeable accuracy can be found in Table II. However, in practice scale of this data is in the scope of Big-Data or stream data which turns this task much more challenging due to different Feature Engineering stages for detecting different security threads as well as Web Application Vulnerabilities [4]. It has been shown for the tasks that need to detect injections, e.g., Cross-Site Scripting (XSS), SQL Injection (SQLI), etc. over HTTP requests; it is also useful to consider the feature extraction stage along with feature selection to extract meaningful features out of specific categorical features over URL parameters (addressed in the Table I). Those extracted features can potentially represent the length of payloads better besides numerical features [48]. Thus considering the synthetic nature of the web server log feature-wise, even though we used unsupervised methods, adapting suitable feature engineering (feature selection/extraction) depending on the task is inevitable for the analysis and parsing of unstructured cybersecurity incident data. Leveraging NLP-based tokenization and other semantic embeddings such as word-embedding and character-embedding to create a meaningful feature map is a key point [61].

C. High-dimensional Data

Typically for low-dimensional tasks, e.g., 1-2 dimensional data, identifying outliers/anomalies could be resolved by plotting the data (points far away from the rest). Algorithms generally consider for anomaly detection in low dimensional data are not suitable for high dimensional data. Thus, unsupervised anomaly detection is close to being a hopeless task due to the curse of dimensionality [62], which - in the sense of anomaly detection - means that every point eventually becomes an outlier. The problems of anomaly detection in high-dimensional data are 3-fold to detect: (a) global anomalies, (b) local anomalies, and (c) micro-clusters of anomalies [63].

Global anomalies can be detected easily since they are very different from dense areas with respect to their attributes. In contrast, a local anomaly is only an anomaly when it is distinct from, and compared with, its local neighborhood. Micro-clusters of anomalies have been paying little attention to this problem relative to the other two categories. Recently, some high-dimensional outlier detection algorithms have addressed this problem to some extent by grouping instances together by selecting a representative member from each cluster before the Nearest Neighbor (NN) distances computation over them [64]. However, it suffers from a few limitations that significantly hinder its ability to detect anomalies under certain situations, which are addressed and improved using the STRAY (“Search and TTrace Anomaly”) algorithm, including k-Nearest Neighbor (KNN) distance calculations [65].
D. Big-Data Analytics

With the advent of Big-Data, the processing efficiency of anomaly detection techniques becomes increasingly complex, specifically when the underlying probability distribution is unknown along with high data size. In other words, the volume feature of Big-Data stresses the storage, memory, and computation, and these requirements need to be increased [66]. The problem is most outlier detection algorithms can perform on small data size computation-wisely, whilst outlier detection tasks through Big-Data require distributed processing to scale out. In this regard, some outlier detection algorithms are not developed for distributed systems yet. The majority of today’s ML-based algorithms are designed for single-thread computation, whereas real-world Big-Data problems require distributed systems.

Following recent surveys [67] for general systematic log anomaly (not only cyber-related logs) shows that Word-embedding has proven significant results for capturing semantic information from log messages. The unsupervised deep learning models such as Variational Auto-encoder (VAE) and Deep Variational Auto-encoder (DVAE) provided promising results and it has already been shown that the DL algorithms perform better subject to the amount of data during the learning process compared to classic ML models plateaus [68].

### E. Imbalanced Data Evaluation

Typically, evaluating unsupervised anomaly detection techniques provides us with a better understanding of their performance in the presence of labeled data. There are two types
of errors we can consider for an anomaly detection model: (i) False Positive (FP) is about predicting a true normal data as being abnormal, causing high False Positive Rate (FPR) or sensitivity, and (ii) False Negative (FN) predicting a true anomaly as being normal and it means that model mispredict anomaly, which result in high False Positive Rate (FPR) or specificity. There is no rule of thumb to handle these two types of errors since it is vary depending on the application. Unlike medic data processing, in cyber-related monitoring tasks that involve large amounts of data, it can be more desirable to have a low FP, usually at the expense of a higher miss rate (TNR). Given the anomaly score (that indicates the “degree of anomalousness”) $s : X \rightarrow R$ of the model, where the decision threshold $\tau$, can be achieved by following Decision function:

$$\text{Decision} = \begin{cases} \text{outlier (positive class)} & \text{if } s(x) \geq \tau, \\ \text{inlier (negative class)} & \text{if } s(x) \geq \tau. \end{cases}$$  \tag{1}$$

Thus $\tau$ needs to be calibrated using domain experts’ knowledge for the specific application to minimize type I and type II errors (FP and FN). The Area Under the Receiver Operating Characteristic curve (AUROC), usually called Area Under the Curve (AUC), is used in classification analysis in order to determine which of the used models predicts the classes best by interpreting the probability that a random positive sample will have a higher score than a random negative sample. In other words, it provides an evaluation measure that considers the full range of decision thresholds on a given test-set [69]. The Receiver Operating Characteristic (ROC) curve plots all (false alarm rate, recall)-pairs that result from iterating over all possible thresholds covering every test set decision split, and the area under this curve is the AUC measure [70]. Suppose $A$ and $B$ are the distributions of scores the model produces for data points that are actually in the positive and negative classes, respectively, and $\tau$ denotes the cutoff threshold. Thus are related via the following relationship:

$$AUC = \int_0^1 TPR(x)dx = \int_0^1 P(A > \tau(x))dx \tag{2}$$

The downside of the AUC is that it can produce overly optimistic scores in the case of highly imbalanced data. Therefore the Area Under the Precision-Recall Curve (AUPRC) is more informative and appropriate to use when precision is more relevant than the false alarm rate [71, 72]. A common robust way to compute the AUPRC is via Average Precision (AP) [73]. One caveat of the AUPRC (or AP) is that the random guessing baseline is given by the fraction of anomalies in the test-set data and thus varies use-case by use-case. This issue makes the AUPRC (or AP) harder to interpret and less comparable over different application scenarios, but in those scenarios where the data is not highly imbalanced, the AUROC and AUPRC (or AP) measurements show the same trends [71, 74, 75]. Due to the point that AUC has been used most of the time to mean AUROC, which is a bad practice since it has been shown AUC is ambiguous (could be any curve) while AUROC is not [76]. In short, the ROC curve is suitable when the observations are balanced between each class or how good the model performs with no knowledge of the class imbalance, whereas the precision-recall (PR) curve is appropriate for imbalanced datasets or it uses the estimated class imbalance baseline to answer how good the model performs, given imbalanced data.

Although most of the related studies we shortlisted in Table II report high accuracy (ACC), however, there is incomplete information concerning error types, which is better to report the F1 score to understand better the used model performance, especially for imbalanced data tasks for further investigations to improve. Lately, there has been another straightforward evaluation approach to demonstrate or compare AD algorithm performance by calculating anomalies that could be found in the top $n$ data in addition to ROC-curve or PR-curve plots [77].

V. Conclusion and Future Works

This paper solely reviewed the literature on web-server logs analysis and WLAD, that one can also expand web-server logs to other similar structure logs such as HTTP Logs, Proxy Logs, etc. This study aimed to investigate and identify the various anomaly detection approaches for web server logs and evaluate their suitability and feasibility in the big data realm. For each category of anomaly detection techniques, we present the assumption regarding the notion of normal and anomalous data along with its strength and weakness. In addition, we discussed some of the main challenges and open topics in the context of WLAD, particularly, the ability to learn from outlierness leading toward maliciousness, as outlierness doesn’t necessarily translate into maliciousness. These assumptions can be used as guidelines to assess the technique’s effectiveness in the web-application domain and to detect right malicious logs with low false-positive rates. The lack of labeled web log data makes provision of the classic learnings challenging for web log anomaly detection on skewed class distribution; nevertheless, meaningful feature mapping besides the novel learnings like positive-unlabeled learning and active learning as well as self-supervised learning could improve weblog-specific augmentation. New learning-based anomaly detections are still active research, and a possible future work would be to extend and update this survey as more sophisticated techniques are proposed.

References


