Anomaly Detection for Web Log Data Analysis: A Review

Barun Kumar Jha*, Nisha**

*M.Tech Scholar , Department of Computer Science , RN College of Engineering , Panipat
**H.O.D, Department of Computer Science, RN College of Engineering, Panipat

***Abstract*: Many methods have been developed to protect web servers against attacks. Anomaly detection methods rely on generic user models and application behaviour, which interpret departures as indications of potentially dangerous behaviour from the established pattern. In this paper, we conducted the use of a systematic review of the anomaly detection methods to prevent and identify web assaults. Many techniques of anomaly identification for automated log analysis have been suggested to minimise manual work. However, due to a lack of evaluations and comparisons of various anomaly detection techniques, engineers may still decide which detection methods should not be used. We offer a comprehensive analysis and evaluation of six existing log-based detection techniques, including three monitored and three unchecked modes, as well as an open toolkit that allows for simple reuse, to address these problems**

***Keywords*: Anomaly Detection, Web attacks, Log Anomaly, Auto Encoder, CNN, Deep Learning, LSTM, Log Parsing.**

I. Introduction

Due to various their high value, Web servers are gradually becoming targets for assaults as the information technology sector advances. SQL injection and cross-site scripting (XSS) threats have been increasingly common in recent years, which is why Web security has received more attention from academic and industry communities. Anomaly is a term used in internet security research. The analysis of log data is used in web detection. Log files, as crucial recording data, may reveal extensive information at the time of system operation and may be used to trace the majority of assaults. However, log systems create a lot of data, and critical information might be lost in the shuffle.

Furthermore, due to the ever-changing nature of assaults and hacking techniques, gathering anomaly data has become increasingly complex, leading to the current problem that manual log file analysis is inadequate to meet log testing standards. In addition, conventional intrusion detection techniques involve operators or programmers to remove the attack features manually, and detect common attack patterns depend on searching of keyword and rule matching [1]. In another words, the conventional approach cannot detect unidentified attacks and leads to fail.

A variety of anomaly detection techniques are being suggested to solve the limitations of conventional methods in order to overcome the shortcomings of previous years. Many machine learning methods are being utilised in the identification of log-based anomalies as a result of advances in machine learning [2]. Anomaly detection techniques are typically split into two groups based on the kind of data and the use of machine learning technology: supervised detection [3] and unsupervised detection [4, 5]. Normal training data, properly described in both happy and bad circumstances, is required for the supervised approach. Unsupervised techniques, on the other hand, do not need labels at all. Their research is based on the fact that a traumatic experience may sometimes act like a far-fetched advertisement.

In this paper, machine learning based system of anomaly detection is proposed for Weblog file. To reduce the above-mentioned disadvantages of traditional method, this system uses an algorithm of Machine learning on two levels. The decision tree classifier is used to choose standard log files, and then Markov's hidden model is used to build a standard data model set (hereafter HMM).This is an example of a model for detecting anomalies. It generates automated knowledge and training based on a large amount of data, raising the stakes in the online security debate to new heights. With an accuracy of 93.54 percent and an error rate of 4.09 percent, an examination of 4,690,000 messages from real-time industrial and real-time import samples demonstrates the efficacy of our anomaly detection technology.

Despite the limited data, we believe that these findings, along with the accompanying conclusions, can be used as recommendations for implementing these methodologies and as guidelines for further development. In conclusion, the following contributions were made by this paper:

- A method for detecting weblog files has been presented as an anomaly detection system.
- After comparing many Machine Learning algorithms and discovering that this anomaly detection system has a small number of facts to discover the truth without giving a solid precision, the system uses a two-level machine learning algorithm, a decision tree algorithm, and HMM to detect undesirable data and anonymous attacks.
- Weblogs are categorized into real-world industrial scenarios with several instances of actual assaults, implying that the data setup is widespread and successful.

II. Related Work

The logs were analyzed. Log analysis has been used to increase software system dependability in a variety of ways, including anomaly detection [10], failure diagnosis [1], programme verification [11],[42], and act prediction [16]. The majority of these log analysis approaches are divided into two steps: log parsing and log mining, both of which have received a lot of attention in current years. He et al. [24] compare the efficiency of four non-system source code offline log parsing methods: SLCT [45], IPLOM [29], LogSig [44], and LKE [20]. [34] proposes an offline log parsing solution which requires linear time and space. Using system

sources, Xu et al. [47] offer an online log processing method. Xu et al. [47] employ PCA to find abnormalities, with the input being a matrix built from logs.
Beschastnikh et al. [11] create a finite state machine that defines system runtime behaviour using system logs. Unlike these articles, which use log analysis to resolve a range of complications, we focus on log-based anomaly detection methods.

2.1. Anomalies Detection

Anomaly detection is the process of looking for out-of-the-ordinary behaviour that can be stated to manual examination and debugging engineers. Bovenzi et al. [13] present an operating system-level method for detecting abnormalities that is suited for mission-critical systems. Venkatakrishnan et.al [46] identify safety vulnerabilities before a system is compromised.
In contrast to past efforts that concentrated on discovering individual anomalies, this study analyses the efficiency of anomaly detection strategies for generic irregularities in large-scale systems. Babenko et al. [9] offer an algorithm for automatically creating explanations from anomaly-detected failures.

2.2. Empirical Research

Since empirical research may often provide practical insights to both academics and developers, there has been a lot of empirical research on software dependability in recent years. Yuan et al. [48] investigate open-source logging practices and offer advice to developers.
Fu et al. [21],[49] investigate the logging industry empirically. Pecchia and colleagues [37] look into the goals and difficulties of logging in industrial settings. The use of decision tree approaches to detect smells in code is investigated by Amorim and colleagues [7]. Lanzaro and his colleagues [25] look on how library code flaws emerge as interface issues. [40] Take a look at long-living bugs from five different angles. Milenkoski and colleagues [33] investigate and organise typical computer intrusion detection approaches. Take, for example, Chandola. [14] Survey anomaly detection methods that employ machine learning practices in a range of domains, but this research focuses on assessing and evaluating existing work that employs log analysis to discover system anomalies.

2.3. Review of Log Anomalies and Deep Learning

To identify suspect business-specific activity and user profile behaviour, T.F. Yen et al. [29] used SIEM log data composed from over 1.4 billion logs each day. Scalability, data noise, and a lack of ground truth were all challenges for this project. The suggested solution demands the generation of a feature vector based on historical data for each internet host. To detect potential security problems, they utilise unsupervised clustering using data-specific characteristics. Manual labelling experts must be aware of the absence of ground-based reality. The technique is rule-based, and historical log processing requires subject-matter expertise. Min Du et al. [2] proposed an architecture for detecting anomalies in log data that does not need any former knowledge of the domain. The
proposed method includes a process for diagnosing log key and parameter value abnormalities, as well as a mechanism for identifying log key and parameter value abnormalities from logs. The probability of the next log key is predicted using a neural network-based method.
A log parameter sequence abnormality can similarly be detected using a comparable LSTM neural network. The software also uses false-positive manual feedback to improve future accuracy. The LSTM considers the log series to be a natural language sequence that may be processed accordingly. Using datasets from BGL, Thunderbird, Open Stack, and IMDB, Amir Farzad et al. [6] suggested a deep learning model for detecting log message abnormalities and compared these models to boost efficiency. The IMDB dataset is used to demonstrate how their method can be used to a range of classification challenges.

III. Data Set Used

Log Datasets: Firms rarely publish production logs due of privacy concerns. Our research yielded two log datasets, HDFS data [47] and BGL data [36], appropriate for evaluating existing anomaly detection algorithms. They contain a total of 15,923,592 log messages and 365,298 anomaly samples. We consider these designations (anomaly or not) to be ground truth.
Additional data about statistical data sets can be found in Table I. There are 11,175,629 log messages on the Amazon EC2 platform [47]. Each block operation in HDFS logs has a unique allocation, writing, replication, and deletion block ID. As a result, session windows, as introduced in III-B, can capture log operations more naturally, as each distinctive block ID can be utilized to break logs into a number of log sequences. We then create 575,061 event count vectors by extracting vectors from these log sequences. There are 16,838 samples that are considered to be random. The LLNL Blue Gene/supercomputer L's system gathered 4,747,963 log messages in BGL data [36]. BGL logs, unlike HDFS data, do not have a unique identification for each task. To segment logs, we must first create log sequences using fixed windows or external doors, and then take out the relevant event count vectors. However, the number of windows is determined by the window's size (and step size). BGL data failures account for 348,460 log messages, and every log sequence that contains any failure records is considered as an anomaly

IV. Specific Attack Detection/ Prevention

We discovered during our research that several of the studies we looked at focused on protecting web servers from certain sorts of assaults, such as DDoS and Injection Attacks. Table 5 shows the specific attacks, the number of research dealing with each attack, and a list of pertinent citations.

Table 1. Detail of attacks

Attack	Number of Studies	Citation
DDos	11	[32-42]
Injection	10	[43-52]
Botnets	2	[53,54]
Defacement	2	[55,56]
Other Attack	62	[57-118]

4.1 Denial of Service (DOS) Attacks

DoS attacks attempt to make a network resource inaccessible by flooding the resource or computer with an excessive number of packets, causing the resource to crash or significantly slow down. DDoS (Distributed Denial of Service) is a large-scale, internet-wide denial-of-service attack. In the first phase, the attacker detects and exploits vulnerabilities in one or more networks in order to remotely control multiple computers by installing malware programmes on multiple systems. “These compromised systems are then used to transmit a large number of attack packets to the target (s), which is usually outside the original computer network. These attacks are carried out without the awareness of the hosts [119]. A system for detecting DDoS assaults was proposed by Thang and Nguyen [32].

4.2 Injection Attacks

This allows a program to communicate harmful code. Using shell commands to access external programs or backend databases are examples of these attacks (i.e. SQL injection). SQL Injection (SQLI) is a common online attack. Poor input validation may allow an attacker to gain direct database access (121). Kozik, Choras, and Holubowicz [3] employed non-supervised token extraction and evolutionary token alignment to detect SQLI and XSS attacks. Wang et al. [44] proposed FCER Mining as a new method for fast locating valid rules in vast data on Spark. It was tested using the SQLMAP map tool against SQLI attacks.

4.3 Botnet Attacks

A bot is a compromised computer that can execute its master's commands, and bots are connected in a botnet according to the master's topology. Due to the presence of Command and Control (C), which sends bot-to-bot commands, botnets are unique sorts of attacks. Bots always hide in the hope of finding an unattended victim to report to the bot-master [23]. A 4 parameter semi-Markov model for browsing behaviour was developed by Yu, Guo, and Stojmenovic [53]. Statistics assaults are impossible to detect if the attacker botnet has a high enough number of active bots (though it is hard for botnet owners to successfully carry out a mimicking attack most of the time).

4.4 Defacement

A bot is a hijacked computer that can execute commands from its master, and botnets are composed of bots [122]. Botnets are distinguished from other forms of attacks by the presence of Command and Control (C & C), which communicates bot-master-to-bot orders to the bots. Bots are always hidden when seeking for an unattended victim, and when they discover one, they report it to the bot-master [123].

V. Conclusion

The main objective of our research is to study various papers related to web attacks and the techniques used for anomaly detection. One of the primary limitations identified in this systematic review is the absence of a standardized, up-to-date, and properly labelled dataset that enables the verification of experimental results acquired in various investigations. It is concerning that only 29.55 percent of the investigational results; found in the reviewed studies are dependent on publicly available datasets, with approximately half of these being heavily criticised by the scientific community, implying that additional research efforts are required to enable the creation and validation of a publicly available dataset. Additionally, a modest number of research utilising dimensionality reduction strategies have been identified. If PCA is utilised, it is recommended to use robust approaches, as the PCA method is extremely sensitive to outliers. If one of the variables in an observation is abnormal, the variance in this direction will be unnecessarily high. Due to the fact that PCA attempts to locate the paths with the greatest variance, the resulting subspace will have been excessively directed in this direction. The majority of the papers evaluated employ K-means and GMM classification algorithms in conjunction with Markov and SVM type models. It is usual to combine two or more clustering and/or classification techniques. Classic measures are commonly employed in studies relating to vulnerability identification.

While these latter measurements may be useful, the authors feel that additional research should be undertaken in this area to develop a clear technique that enables the selection of certain metrics.

References

[1]Liao, H.J.; Richard Lin, C.H.; Lin, Y.C.; Tung, K.Y. Intrusion detection system: A comprehensive review. J. Netw. Comput. Appl. 2013, 36, 16–24.
[2]Jyothsna, V. A Review of Anomaly based Intrusion Detection Systems. Int. J. Comput. Appl. 2011, 28, 26–35.
[3]Kakavand, M.; Mustapha, N.; Mustapha, A.; Abdullah, M.T.; Riahi, H. A Survey of Anomaly Detection Using Data Mining Methods for Hypertext Transfer Protocol Web Services. JCS 2015, 11, 89–97.
[4]Samrin, R.; Vasumathi, D. Review on anomaly based network intrusion detection system. In Proceedings of the 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), Mysuru, India, 15–16 December 2017; pp. 141–147.
[5]Kitchenham, B.; Charters, S. Guidelines for Performing Systematic Literature Reviews in Software Engineering Version 2.3; Technical Report; Keele University: Keele, UK; University of Durham: Durham, UK, 2007.
[6]Brereton, P.; Kitchenham, B.A.; Budgen, D.; Turner, M.; Khalil, M. Lessons from applying the systematic literature review process within the software engineering domain. J. Syst. Softw. 2007, 80, 571–583.
[7]Budgen, D.; Brereton, P. Performing Systematic Literature Reviews in Software Engineering. In Proceedings of the 28th International Conference on Software Engineering, Shanghai, China, 20–28 December 2006; Association for Computing Machinery: New York, NY, USA; pp. 1051–1052.
[8]Kitchenham, B.; Pearl Brereton, O.; Budgen, D.; Turner, M.; Bailey, J.; Linkman, S. Systematic literature reviews in software engineering—A systematic literature review; Inf. Softw. Technol. 2009, 51, 7–15.
[9]Kitchenham, B.; Brereton, P. A Systematic Review of Systematic Review Process Research in Software Engineering. Manuscr. Publ. Inf. Softw. Technol. 2013, 55, 2049–2075.
[10]Patel, A.; Taghavi, M.; Bakhtiyari, K.; Celestino Júnior, J. An intrusion detection and prevention system in cloud computing: A systematic review. J. Netw. Comput. Appl. 2013, 36, 25–41.
[11]Raghav, I.; Chhikara, S.; Hasteer, N. Article: Intrusion Detection and Prevention in Cloud Environment: A Systematic Review. Int. J. Comput. Appl. 2013, 68, 7–11.
[12]Patcha, A.; Park, J.M. An overview of anomaly detection techniques: Existing solutions and latest technological trends. Comput. Netw. 2007, 51, 3448–3470.
[13]Chandola, V.; Banerjee, A.; Kumar, V. Anomaly Detection: A Survey. ACM Comput. Surv. 2009, 41.
[14]Jose, S.; Malathi, D.; Reddy, B.; Jayaseeli, D. A Survey on Anomaly Based Host Intrusion Detection System. J. Phys. Conf. Ser. 2018.
[15]Fernandes, G.; Rodrigues, J.J.P.C.; Carvalho, L.F.; Al-Muhtadi, J.F.; Proença, M.L. A comprehensive survey on network anomaly detection. Telecommun. Syst. 2019, 70, 447–489.
[16]Kwon, D.; Kim, H.; Kim, J.; Suh, S.C.; Kim, I.; Kim, K.J. A survey of deep learning-based network anomaly detection. Clust. Comput. 2019, 22, 949–961.
[17]Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A Detailed Analysis of the KDD CUP 99 Data Set. In Proceedings of the Second IEEE International Conference on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 8–10 July 2009; IEEE Press: Piscataway, NJ, USA, 2009; pp. 53–58.
[18]McHugh, J. Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory. ACM Trans. Inf. Syst. Secur. 2000, 3, 262–294.
[19]Mahoney, M.V.; Chan, P.K. An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection BT—Recent Advances in Intrusion Detection. In Recent Advances in Intrusion Detection; Vigna, G., Kruegel, C., Jonsson, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2003; pp. 220–237.
[20]Brugger, T. KDD Cup '99 dataset (Network Intrusion) considered harmful. KDnuggets News 2007, 7, 15.
[21]Ieracitano, C.; Adeel, A.; Gogate, M.; Dashtipour, K.; Morabito, F.C.; Larijani, H.; Raza, A.; Hussain, A. Statistical Analysis Driven Optimized Deep Learning System for Intrusion Detection BT. In Advances in Brain Inspired Cognitive Systems; Ren, J., Hussain, A., Zheng, J., Liu, C.L., Luo, B., Zhao, H., Zhao, X., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 759–769.
[22]Ieracitano, C.; Adeel, A.; Morabito, F.C.; Hussain, A. A novel statistical analysis and autoencoder driven intelligent intrusion detection approach. Neurocomputing 2020, 387, 51–62.
[23]Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J. Survey of intrusion detection systems: Techniques, datasets and challenges. Cybersecurity 2019, 2, 20.
[24]Ahmed, M.; Naser Mahmood, A.; Hu, J. A survey of network anomaly detection techniques. J. Netw. Comput. Appl. 2016, 60, 19–31.
[25]Kotu, V.; Deshpande, B. Chapter 13 Anomaly Detection. In Data Science, 2nd ed.; Kotu, V., Deshpande, B., Eds.; Morgan Kaufmann: Burlington, MA, USA, 2019; pp. 447–465.