IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol. 9, Issue.6, June 2021 E-ISSN: 2347-2693

Anomalous Traffic Detection System for Enterprise using Elastic stack with Machine Learning

Ruchita R Biradar[1*], Nagaraja G.S.[2]

[1,2]Department of Computer Science and Engineering, R. V. College of Engineering, Bengaluru, India

[*]*Corresponding Author: ruchitarbiradar.cs17@rvce.edu.in, Tel.: +91-97898-05721*

DOI: https://doi.org/10.26438/ijcse/v9i6.1318 | Available online at: www.ijcseonline.org

Received: 15/Jun/2021, Accepted: 21/Jun/2021, Published: 30/Jun/2021

***Abstract*—** The logs in a network are not bound to be perfect perpetually. The behavior of the network traffic is bound to deviate from the expected one sometimes and when that occurs, the traffic is said to be anomalous. Anomalous traffic can be problematic for various reasons, be it external attacks, or transfer of outdated data or even serving customers for networking companies. When the network size is at a large scale, this becomes an even bigger problem to tackle. The anomaly detection systems currently in place are either trained with aging datasets or are not able to handle large loads efficiently. Hence arises the need for a scalable solution which can provide security to a network by detecting anomalies in it and alerting with quick response when an anomaly occurs by learning from its past behavior. The paper offers an end-to-end solution for the introduction of an anomaly detection system using machine learning into an enterprise environment, right from the collection of logs to the generation of alerts, effectively. This is implemented with an infrastructure that includes Elasticsearch, Logstash and Kibana along with the added feature of Machine Learning.

***Keywords*—** Alerting, anomaly detection, machine learning, networks

I. INTRODUCTION

The times are such where everyone is observing a paradigm shift in increased use of internet and digitization of things that used to be offline. Owing to the increased number of users and traffic on networks, there is a need for enhanced security for these said networks.

A DDOS attack, or an outdated version of a client's OS, or even poor network performance, all affect the working of an organization. When the organization is such that it deals with millions of clients on its network, one starts to look for a scalable and secure anomaly detection system which can deliver required results. What is more important to note is that, since time is of the essence in a networking security environment, the system must also deliver a speedy response.

In September 2020, a cybersecurity company, Crowdstrike issued a statement that the company had encountered more intrusion attempts in the first half of 2020 than it had in the entire year of 2019. It goes to show how much the world of cybersecurity has been impacted by the explosion of digital engagements, and it has been predicted that this trend will only go upwards. It also emphasizes on the need for better security systems which predict network vulnerabilities instead of having to mitigate these issues whenever they arise.

Vulnerabilities like software or hardware defects in the network infrastructure enable outsiders to exploit them [1]. There needs to be constant awareness of new problems that are likely to arise, and then decide an action targeting them. In the past decade, Machine learning has been applied in intrusion detection systems for achieving better detection rates and adaptability [2]. And furthermore, while supervised learning gives better accuracy on data with known attacks, unsupervised learning is a better option for dealing with unexpected type of data.

Elasticsearch is popularly being adopted as a storage component by huge organizations like Netflix, GitHub and more, to handle growing data and its storage [3]. It also proves to be promising for researchers to explore mining modern data repositories. Its popularity can also be chalked up to its speed in real-time data analysis, which is made possible by its unique data sharding and standardization process [4].

The paper proposes a proactive and scalable anomaly detection system which includes Elasticsearch, Logstash and Kibana, popularly called together as the ELK stack or Elastic stack, along with machine learning to analyze network logs, detect anomalies, and generate alerts all in real time, for which the system administrator may take suitable action. This solution is specifically targeted at enterprise environments which have large volumes of network logs.

The sections in the paper are organised as follows: section II talks about related work in this domain, section III provides information on the dataset used, section IV discusses the metrics chosen for analysis, section V describes the methodology and components of the ELK

infrastructure used. Section VI provides some insights and results of the system and section VII discusses the future scope of the system. Section VIII presents the conclusion of the paper.

II. RELATED WORK

Where there is a network, there is a potential threat to its security. There are a number of threats that exist today, some popular ones of which are viruses, worms and malware. Not only this, but as security systems are constantly updated to handle these threats, the threats are also being constantly updated so as to not be detected on the network.

In [5] it is mentioned that there are some domains of networks like the defence and hospital data that cannot be compromised at all costs, but these still remain under threat. Lack of good cyber security infrastructure combined with continuously evolving technologies highlight the need for a better line of defence. In this case, it had been proposed to encrypt data to ensure confidentiality using symmetric and asymmetric encryption, create backups for integrity and usage of up-to-date software for availability. Encryption is a right step towards protecting data but we need a solution to secure all forms of data which is not as computationally expensive.

In 2017, Xiaokui Shu explored a solution [6] using digital signatures, adaptive warning strength, connections among alerts and zero trust strategy but it was found that digital signatures are difficult to be applied on customers computers as they may have programs from unidentified vendors running and processing large amount of security alerts is challenging. Moreover, Zero trust strategy requires huge computation power to be utilized to monitor traffic since it tries to tackle both outsider and insider attacks. Hence it was not easily scalable in very large networks.

Systematic Literature Reviews (SLR's) have been conducted on data mining techniques used in intrusion detection systems in the past [7]. What those reviews concluded was that data extraction was not a simple task. As volume of networks grew each year, real-time monitoring has shown various challenges in research. In addition, most of the existing intrusion detection systems work with outdated datasets like KDD'99, which was developed in 1999, when half the threats today did not exist.

For detecting anomalous activities on a network, there has been a comparison of adaptive graph-based optimization approaches to one-class SVM [8]. The approach though, used only two datasets due to space constraints, as well as in only two different settings, hence the performance was not generalized. The increased complexity of the approach also leaves room for improvement.

The content-based data leak prevention and detection (DLPD) systems for enterprises face the main demanding issue of scalability, which means they are unable to process large volumes of network logs in time [9].

Many papers discussed anomaly detection systems that can detect anomalies from training and test datasets obtained and stored. But network traffic occurs as time-series data and these solutions do not have the ability to perform time-series analysis and give quick alerts for anomalies in real time. They also do not perform well in real world scenarios with actual data since they have been trained and tested with outdated and predictable data. Supervised learning techniques suggested in some papers would not work well against unpredictable data hence unsupervised learning needs to be explored. Finally, the solution should be scalable to be suitably used for enterprises.

Thus, the paper proposes an anomaly detection system that uses unsupervised learning to model normal behavior and predict unforeseen problems, be able to work with large volumes of data, and create alerts from a stream of logs generated in real time.

III. DATA USED

While the datasets currently in use for anomaly detection systems vary in creation time from 1999 to 2018, they can still be outdated for unforeseen circumstances.

Hence the data used here is real time network logs taken from an enterprise. The data available for use originates from January 2021 till date, which amount to 6 months' worth of network logs, and continues to update as and when the logs are generated. The logs are collected from an environment where a lot of testing of applications occur, hence there are also some failure cases that can be used.

The proposed solution aims to be scalable, so it means that it should be able to process millions of network logs of data. This means that storage space availability should also be accordingly available. In this case, 6 months of data amounts to 20 GB approximately. Datasets like DARPA-2009 have packet capture files for one-minute windows with each file taking up 1 GB of space [10]. Hence a better option would be to use enterprise data with only necessary fields included in it. The resources used up for machine learning, called the established model memory, take up space from 100 KB up to 700MB for complex cases. Hence the default memory limit for required resources is set to 1GB, also known as model memory limit, indicating that the utilized memory should not cross this. The limit can be changed to a higher limit if the user desires, provided they have that much extra memory available.

IV. METRICS

For each machine learning job, there is a certain metric to be observed. It could be the client IP addresses occurring on an applications' network, or the response codes being generated for a particular app. These differ as per the use cases. Detectors are used to aggregate the metrics and

configure the type of aggregation that will occur for the metrics. Few examples of detectors include count, average, median, high median and high mean. The difference between median and high median is that median would give an anomaly if the actual value was much higher or much lower than the typical value. But high median gives an anomaly if the actual value is only higher than the typical value. This helps in cases where you want to observe spikes in network activity only, and low network activity is considered insignificant.

Influencers can also be set along with detectors, if it is observed that there are certain entities contributing to anomalies significantly. Then it gets simpler to understand why the anomaly is occurring, For example, if spikes in network activities are mostly being associated with a few troublemaking client IPs, you can deduce that client IPs are influencers for the anomalies occurring and set that as an influencer. This will help attribute the cause of an anomaly to a particular reason during analysis.

V. METHODOLOGY

Each component of the system plays a significant role in achieving the desired anomaly detection results. Beats for streaming data, Logstash is used for formatting and processing the data before sending it to Elasticsearch which is the storage component, Kibana which provides visualizations to investigate trends and shifts in the data. The purpose of using machine learning in the Elastic stack is for predicting possible complications before they occur.

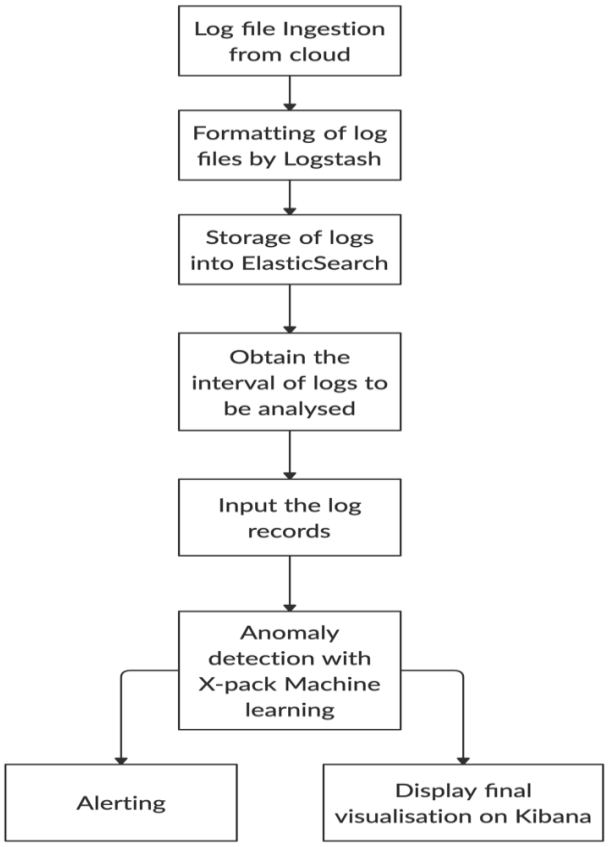

Figure 1. System Flow diagram

Anomaly detection is about identifying things we probably cannot visually and taking action based on it. Fig. 1 depicts the flow of the system. The system performs anomaly detection on time-series data, with unsupervised learning. It is a combination of clustering, time-series decomposition, Bayesian distribution modelling and correlation analysis.

A. Filebeat

Beats is a component that streams data to and from machines and the Elastic Stack. One such Beat is Filebeat, that can read and send whole log files. The network logs generated by an Nginx server reside on the cloud in an unformatted manner. Firstly, access to the logs has to be ensured. Then Filebeat takes logs from cloud and sends them to a particular path, one which Logstash listens to for obtaining them later. Filebeat also adds tags to the data, this is due to the reason that there are many operations in Logstash and Kibana that could potentially use these tags for easy identification of the data.

B. Logstash

Logstash obtains data by listening to a particular path for data, obtains the network logs, and specifies an endpoint for them, in this case the Elasticsearch cluster. Logstash also formats the logs which so far consist of only values, by modifying them into key-value pairs. Hence Logstash creates a pipeline for the data to travel from its origin to the endpoint, meanwhile also formatting the data in the process.

C. Elasticsearch

Elasticsearch denotes the storage component of the stack. It consists of a cluster of nodes, on which the formatted network logs are divided and stored, along with their replicas configured. The logs are indexed on a daily basis here and separate indices are created for each date for storing them. This data can later be aggregated and obtained from the indices based on the time interval to be analyzed for anomaly detection.

D. Kibana

The Kibana interface is the main interaction the user has with the anomaly detection system. It provides all the visualizations of the data, baseline behavior, as well as the anomalies. It can generate visualizations such as graphical representations in a single metric viewer or in the form of heatmaps in an anomaly explorer. There is an option to create and set a dashboard that can contain all different details of the data which are easily readable due to the availability of vast number of options. Some of these include[11] coordinate map, gauge, line graph and pie chart. Kibana also has a feature called Graph, where the relationships between entities of your choice can be visualized using nodes and edges.

E. Machine Learning

The anomaly detection is intuitively done by creating machine learning jobs. The X-Pack extension is used to add machine learning into the system.

The options considered are single-metric job, multi-metric job and a population job. The three jobs can be compared for a use case : Studying the response codes for an application.

- Single-metric job: In a single metric job, one can observe any one metric by using a single detector. For the use case, it is only possible to observe the number of network logs by using detector count. There is no option to split the logs by response code and observe them. The job works for observing a single time series, using a single detector.
- Multi-metric job: In a multi metric job, there are options to use multiple detectors. The job tracks multiple time series via detectors for each of them. Hence for the use case, the network logs can be partitioned by response codes, and each set of logs are observed individually by using the detector count, i.e., the baseline behavior of logs for one response code will not be affected by baseline behavior of another response code, they will be independent. For each partition, a baseline behavior is observed, and anomalies are detected in them. If the number of 404 logs have experienced a spike compared to past behavior, then an anomaly is reported specifying it occurred for response code 404. This is unaffected by other response codes. Then the system administrator observing this would get the idea that the application is generating a lot of 404 error response at that particular time and is not functioning properly.
- Population job: In single and multi-metric jobs, the anomaly was detected by comparing each metric with its own past behavior. In a population job, a metric is observed as a population. This observes any anomalous activity in the behavior of the population. Now instead of observing the response codes individually, they are studied together by the system. For this use case, one can set the population as response codes and behavior of any response code deviating from the population of other response codes is reported as an anomaly. For example, if the trend is that the occurrence of response code 200 in 5 minutes is usually 1000 logs and response code 404 is 258 logs, the system understands that frequency of 404 is almost 1/4th that of 200 in a time interval. Then if 404 shoots up to 500 but the relative behavior of 200 does not change accordingly, then it is an anomaly. The population job does this for each response code, by comparing it with every other response code. The population can also be split by a metric, say user ids, then the population of response codes for each user id is observed individually.

Having seen the different machine learning jobs possible, the memory occupied by each of these jobs are to be taken into account. These are tabulated for the response codes use case mentioned above.

Population job is useful when the general behavior of the metric is not too different from one another. It does not require much space compared to a multi metric job and is scalable. However, if the unique count of the metric chosen is low, then this field would probably not be the right choice as population. In that case, a multi metric job might be considered.

Table- I: Resource utilization of Machine learning jobs

Job type	Memory used	Remarks
Single-metric	50.2 KB	Not really useful, as good as observing anomalies in number of logs generated than in response codes
Multi-metric	882.4 KB	Works well, to observe anomalies in response code behavior individually
Population	109.1 KB	Excellent for learning population of response codes, and then observing which response code deviates from the population

F. Watcher Alerts

Having performed anomaly detection, an important aspect is alerting. Based on a certain condition or threshold, the system administrator would require an alert whenever an anomaly occurs within real time. For this, Watcher is used to create alerts in the form of emails or logging. Trigger intervals can be given as to when the watch must be triggered. For example, the watch can be triggered every 30 seconds whether the condition of the watcher, which is that an anomaly of score above 75 has occurred, has been satisfied. If the condition is satisfied, alert is sent. To prevent too many alerts at once, the watcher can be throttled from sending alerts for a set time interval, say 1 hour, configured as throttle period. Hence in a throttle time interval, the system administrator will only get alerted once, avoiding spamming in the process.

VI. RESULTS AND ANALYSIS

A. Visualization of data

In Kibana, the machine learning model has a blue shaded region within which it predicts the typical value to occur. That is the baseline behavior. If the actual value that occurs is very different from this behavior, then it is reported as an anomaly. It does this by calculating the probability of an observation being an anomaly and projecting it on a scale of 1 to 100 for easy understanding. This is called the anomaly score, and the scores are categorized into 4 severity levels : warning (0 and above), minor(25 and above), major(50 and above) and critical(75 and above) along with different colors for representation.

Figure 2. Graphical representation of data

In Fig. 2 above, the graph represents the data and orange dot implies an anomaly occurred in the data, in the major anomaly category.

Another visualization Kibana called Graph is used for viewing relationships between different fields in the log files using nodes and edges. In Fig. 3 below, the relationships between response code 404, URL accessed, and client IP, represented by pink, green and yellow nodes respectively, are observed using sample data from Kibana.

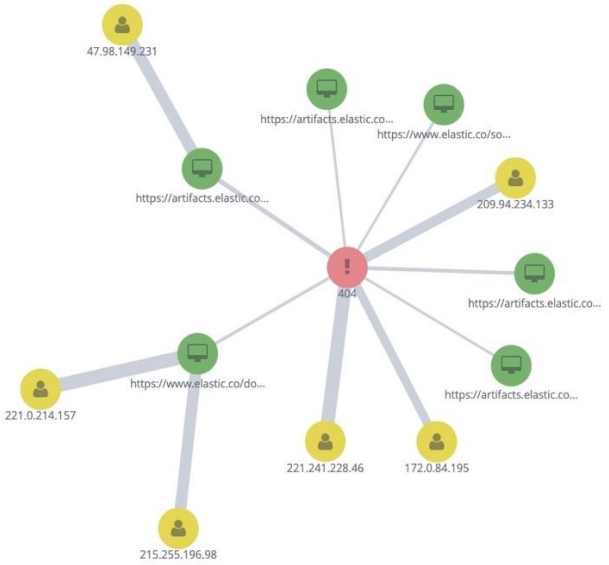

Figure 3. Kibana Graph of nodes

The graph can be modified in hops as well, in the first hop the entities directly related to the freshly created nodes are added, then in the next hop, entities directly related to the newly added nodes, as well as indirectly related to the original nodes are added. The number of nodes added in each hop are limited by a maximum number set by the user.

B. Filtering data

Often there are cases where one does not want a specific value of the metric to come in as part of the anomaly detection. For this filter lists are available to filter out data containing values from the list, and custom rules can be applied to skip model updates and/or skip result.

C. Email alert

Watcher alerts configured in the form of emails are sent to the system administrator. Based on throttle interval they can be generated within few seconds of the occurrence of the anomaly or sent altogether after a longer time interval like an hour, aggregating a few anomalies.

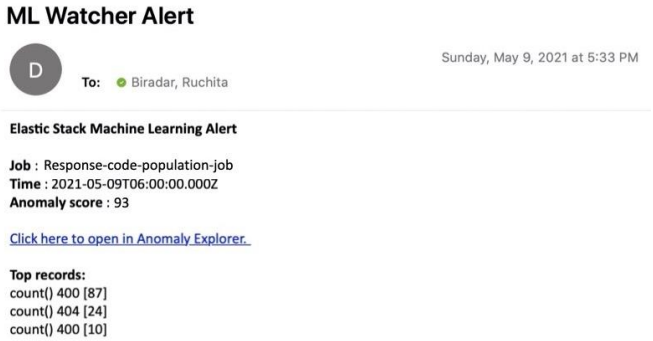

Figure 4. Alert received by the system administrator

As in Fig. 4, the alert includes details like the job ID, timestamp surrounding the anomalies that occurred for different metric values around the same time, and a hyperlink that leads to a visualization in Kibana of the data around the time the anomaly occurred. Here the anomaly score is an aggregate of all the anomalies occurring around this timestamp. The top records of anomalies are also shown. Though this is a default template of an email alert, the body of the email can be accordingly modified.

D. ELK compared to Splunk

Splunk is a log analysis tool similar to Elastic stack but with a different approach. However, as per Google trends, the ELK stack has overcome Splunk in terms of popularity and is predicted to become even more widely used as more features get added to it.

Fig. 5 shows this particular shift, with the blue line representing ELK and red representing Splunk where the shift occurred mainly between the years 2013 and 2015. In [12], it has also been mentioned for retrieving a thousand million records, the Elastic configuration took 1 min and 14 seconds, while Splunk took about 1 min and 22 seconds so the performance of Elastic stack is slightly better.

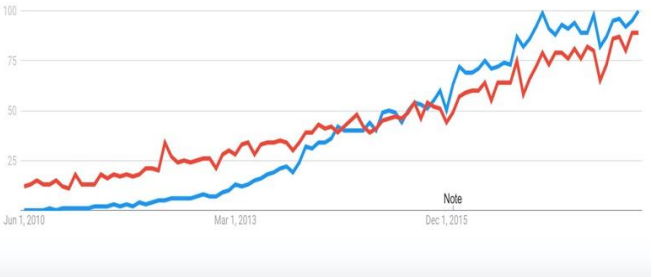

Figure 5. Growing popularity of ELK stack

E. Performance analysis

It is found that multi-metric jobs take up the most space among the three machine learning jobs discussed, due to

observation of each partition of logs independently. Response times of Kibana are tracked according to number of requests as below in Fig. 6.

On an average, Kibana showed a response time of 1ms for 2 client requests. When the load is optimized on Kibana, the response as well as health of Kibana improves. This is an important aspect to consider since the users' main interaction with the system is through the interface offered by Kibana.

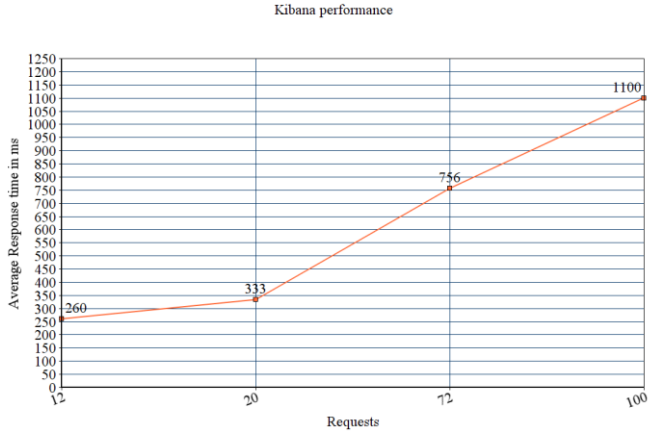

Figure 6. Kibana performance trend

VII. FUTURE SCOPE

The system, though scalable and effective, does have some scope for future improvements which are identified below:

1) X-pack of ELK stack which supports machine learning features is a paid version and license needs to be acquired. Free alternatives to introduce machine learning in ELK are yet to be explored
2) More efficient utilization of memory by machine learning models can be investigated
3) Kibana can sometimes give server errors or give slow response if the data load is high

VIII. CONCLUSION

The paper provides a suitable choice for introduction of an anomaly detection system using machine learning in an enterprise environment.

The solution is very useful for companies which have large amounts of network traffic involving their clients. Hence, being largely scalable and having the ability to process millions of log records, the proposed solution is a torchbearer for anomaly detection systems to come in the future.

REFERENCES

[1] S. N. Hussain, N. R. Singha., "*A Survey on Cyber Security Threats and their Solutions*", International Journal for Research in Applied Science and Engineering Technology, Vol. **8,** Issue. **7**, pp. **1141-1146**, **2020**.

[2] M. Zamani, "*Machine Learning Techniques for Intrusion Detection*", arXiv:**1312.2177, 2013**.

[3] O. Kononenko, O. Baysal, R. Holmes and M. W. Godfrey, "*Mining modern repositories with Elasticsearch*", In the Proceedings of the 2014 Conference on Mining Software Repositories, **India**, pp. **328–331, 2014**.

[4] N. Shah, D. Willick and V. Mago, "*A framework for social media data analytics using Elasticsearch and Kibana*", Wireless Networks, Vol. **24**, Issue.**8**, pp. **1-9, 2018.**

[5] Sharma, Chalsi and Maurya, Satish, "*A Review: Importance of Cyber Security and its challenges to various domains*", International Journal of Technical Research & Science Special(Issue.**3**), pp. **46-54**, **2020.**

[6] X. Shu, K. Tian, A. Ciambrone and D. Yao, "*Breaking the Target: An Analysis of Target Data Breach and Lessons Learned*", arXiv:**1701.04940**, **2017.**

[7] F. Salo, M. Injadat, A. B. Nassif, A. Shami and A. Essex, "*Data Mining Techniques in Intrusion Detection Systems: A Systematic Literature Review*", in IEEE Access, Vol. **6**, pp. **56046-56058**, **2018.**

[8] S. D. Bhattacharjee, Y. Junsong, Z. Jiaqi, Y. Tan, "*Context-Aware Graph-Based Analysis for Detecting Anomalous Activities*", In the Proceedings of the 2017 IEEE International Conference on Multimedia and Expo (ICME), **China,** pp. **1021-1026, 2017**.

[9] L. Cheng, F. Liu and D.Yao, "*Enterprise data breach: causes, challenges, prevention, and future directions*", WIREs: Data Mining and Knowledge Discovery, Vol. **7,** Issue.**5, 2017.**

[10] N. Moustafa and J. Slay, "*Creating Novel Features to Anomaly Network Detection Using DARPA-2009 Data set*", In the Proceedings of the 2015 14th European Conference on Cyber Warfare and Security ECCWS-2015, **UK**, pp. **204-212**, **2015.**

[11] P. P. Bavaskar, O. Kemker and A. Sinha, "*A SURVEY ON: "LOG ANALYSIS WITH ELK STACK TOOL"*, International Journal of Research and Analytical Reviews (IJRAR), Vol. **6**, Issue.**4,** pp. **965-968, 2019.**

[12] S. J. Son and Y. Kwon, "*Performance of ELK stack and commercial system in security log analysis",* ", In the Proceedings of the 2017 IEEE 13th Malaysia International Conference on Communications (MICC), **Malaysia,** pp. **187-190**, **2017**.

AUTHORS PROFILE

Ms. Ruchita R Biradar is a 4th year Student at R. V. College of Engineering, Bengaluru, pursuing a Bachelor's Degree in Computer Science and Engineering. Her interests include Networking, Machine Learning and Software Development.

Dr. Nagaraja G. S. is a Professor and Associate Dean of the Department of Computer Science and Engineering at R. V. College of Engineering, Bengaluru. His experience includes 25+ years of teaching, 1 year of Industry and 16+years of Research. His areas of interest include Computer Networks & Management, Multimedia Communications, Computer Architecture and Protocol Design

