Response time Lag in Accessing an Application When Exposed to Dynamic and Static Security Scans:Using Data-Mining Techniques

Mohamed Fazil Hussain
New Jersey, USA
hfazil@lvl237.com

Salwa Sayeedul Hasan
Hyderabad, India
hasan.salwa@outlook.com

Hasan Rauf
Texas, USA
hasan.rauf@gmail.com

***Abstract*—The development of quicker web interfaces has enabled us a range of applications for all of our needs in communication, banking, shopping, etc. The technology and the IT infrastructure that hosts these applications is constantly evolving, there is a substantial risk of Cyber-attacks on the systems and the application. To protect these, both static and dynamic scans are used constantly to protect online resources from harmful attacks. As a result, the infrastructure owners and application develope must be aware that it causes latency while accessing these applications.**

This study is an effort to analyze the impact of response time on the application using Data Mining techniques. The Google Cloud Platform (GCP)® is used to host a sample application, which is subjected to static and dynamic scan using an industry-standard tool such as Twistlock®. The user interaction response time is then recorded using the TCP/IP 3-way handshake. This data was captured to only the source and destination of the IPs involved in the response of the request while discarding re-transmission and malformed packets in the interaction. Then the collected data is compared to an Ideal scenario, where no scans are performed against captured data when static and dynamic scans are used. The data is further analyzed using industry-recommended data mining techniques and the results and observations are captured. This study focuses on measuring lag in the response time due to constant security scanning while accessing an application. This is done by using TCP/IP 3-way handshake data capture, a data-mining technique such as hierarchical and k-means Clustering, and uses rationale on the anomalies observed such as outliers in the collected data.

***Index Terms*—Clustering, Google Cloud Platform (GCP), K-Means, Hierarchical, Outliers, TCP/IP 3-way hand shake Delta time, TCPDump, ...**

I. Introduction

The advancement in the cloud computing environment arises new security challenges. The usage of cloud-related services for consumers is on an increase, and security experts are required to continuously monitor/scan IT resources to protect consumer access/usage. Failing which the companies/Enterprises are prone and subject to Cyber-attacks that could result in lost business, damage to the name brand, and huge ransomware payoffs as cited in the reference [2] and depicted in Figure (1).

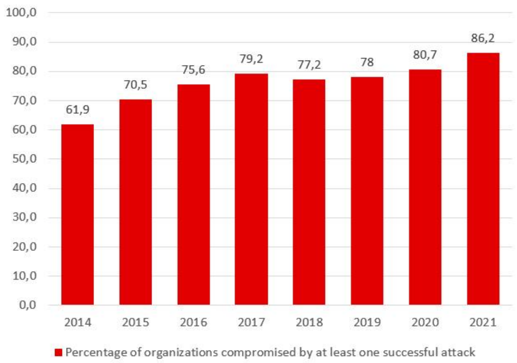

Fig. 1: Graph depicting the increase of Cyber-attacks

To prevent such cyber-attacks, those who are in the security business must routinely check for security gaps and close them before an attack occurs. The most common way to achieve this objective is to make firms/enterprises proactively run their static and dynamic security scans as shown in Figure(3) to find and fix vulnerabilities before an attacker exploits them. Although system administrators frequently apply patches to systems to address vulnerabilities, occasionally administrators fail to apply patches/check vulnerabilities, thus missing updates/fixes are discovered, which are part of static scans of the resources.

Malicious Attacks mostly happen by way of injecting malicious code into the system such as (SQL Injection, aka code injection), cross scripting, etc to name a few see OWASP/NIST standards for more details (https://owasp.org/www-project-application-security-verification-standard/). To prevent such vulnerabilities in the application system a dynamic vulnerability scan is a must, which scans every http/REST request which comes into the system requesting access to the application resources. Failing to detect the dynamic vulnerabilities in the incoming HTTP/REST requests can result in Cyber-attack and ransomware attacks [2].

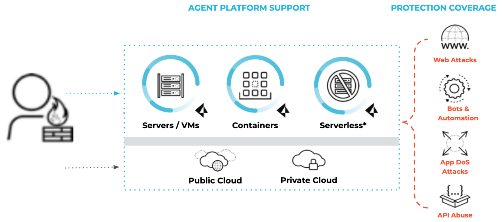

Fig. 2: The flow of Security scan

The Dynamic scans on the cloud are performed while the application is running in a containerized environment in the cloud as shown in the Figure (2). The dynamic scanning of the incoming network traffic viz HTTP/HTTPS/REST/WSDL type TCP/IP requests, as shown in Figure(3). In a dynamic scan, every incoming request is scanned for malicious content hence there is a lag in the response time of the real request/response from the user. The content in the Figure(3), is explained as follows: the user wants to access the application "App Foo" which is hosted in the container by issuing an HTTP command say GET index.html before the index.html content is presented to the user, the request is scanned by the defender and only it passes the scan and the GET is executed. A status code of 200 okay is presented and a response is received by the user.

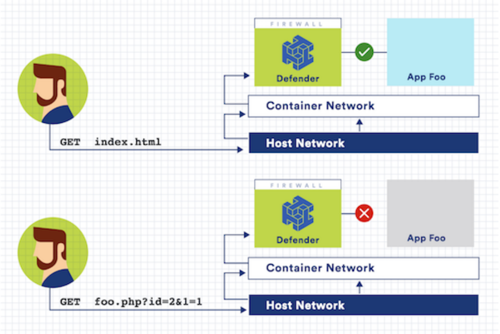

Fig. 3: Http(s) Get/Post Request on an App running in a Container

Static scans are primarily checking for vulnerabilities in the OS Environment and the GIT repository where the code is kept, before being in running mode. In other words, static scans assess the application's underlying structure rather than its functionality. while these scans happen, a consumer or user of the application under scan may expect a lag in the response of accessing the application, while these scans are running and monitoring the system for security.

In this study the response of an application, while dynamic scans are conducted, is computed. Then an analysis of the computed data was conducted in the cloud using Data Mining Techniques and results on the lag in the response times are reported of the analysis which was conducted. Further more information regarding the comparison of k-Means clustering with other techniques is discussed [6][7].

II. Problem Statement

Modern-day applications are hosted in a containerized environment (docker/Kubernetes) either on the cloud or On-Prem, see Figure(4) for the complete user data flow of the application, when the application is under scan. The 3-way handshake data and its explanation is provided in the legend below where each numbered flow is explained. These applications have exposed public interfaces for user access, and they are prone to malicious attacks. To protect an application from such attacks, security scans such as a. Dynamic scan, b. Static scan, and c. The firewall scans also known as port scans are performed, as shown in Figure(3).

These scans cause response time lag when a user accesses an application under dynamic/static scan. A study was previously conducted by the authors on analyzing the amount of lag caused by these continuous scans, and the findings were reported in [1].

In our previous work [1] we have used k-Means as the Data Mining technique, to calculate the amount of lag experienced by a user while accessing an application. Clustering is considered an important unsupervised machine learning technique that deals with a large amount of data, and the clustering techniques play an important role in grouping similar data and its analysis, reported in [10].

It was reported in [5], that k-Means as a clustering technique fairs poorly when a large number of outliers are present in the data. Also, it was suggested to start with the Hierarchical clustering technique first, and follow the methodologies as outlined and given in the Figure(5).

This paper is an effort to validate the data mining techniques previously used and to draw a process for selecting an appropriate Data Mining Technique along with the required Data-Analytics methodologies to properly measure the response time lag in an application that is constantly running under dynamic/static scans.

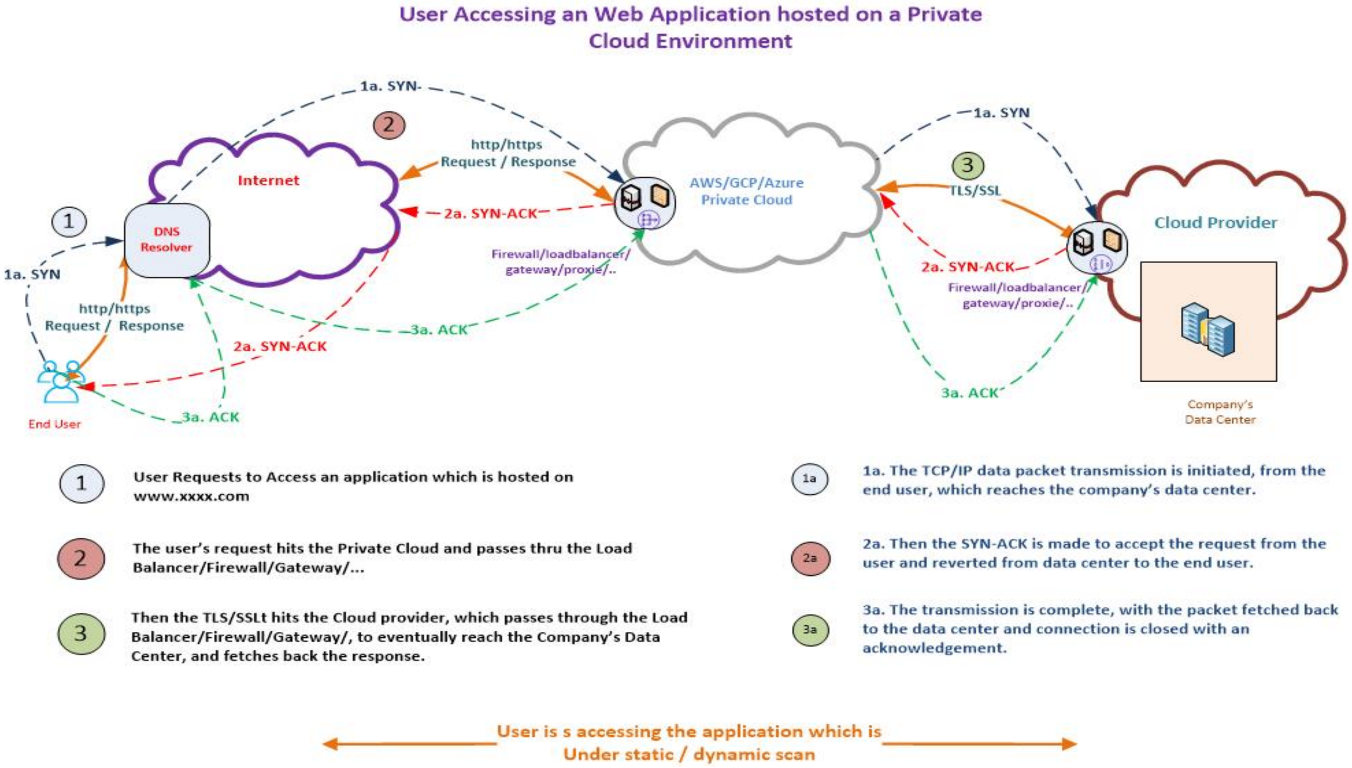
User Accessing an Web Application hosted on a Private Cloud Environment
1a. SYN-
2
http/https
Request / Response
Internet
1
DNS Resolver
2a. SYN-ACK
1a. SYN
http/https
Request / Response
Firewall/loadbalancer/ gateway/proxie/..
AWS/GCP/Azure Private Cloud
3a. ACK
2a. SYN-ACK
End User
3a. ACK
1a. SYN
3
TLS/SSL
Cloud Provider
2a. SYN-ACK
Firewall/loadbalancer/ gateway/proxie/..
3a. ACK
Company's Data Center
1
User Requests to Access an application which is hosted on www.xxxx.com
2
The user's request hits the Private Cloud and passes thru the Load Balancer/Firewall/Gateway/...
3
Then the TLS/SSLt hits the Cloud provider, which passes through the Load Balancer/Firewall/Gateway/, to eventually reach the Company's Data Center, and fetches back the response.
1a
1a. The TCP/IP data packet transmission is initiated, from the end user, which reaches the company's data center.
2a
2a. Then the SYN-ACK is made to accept the request from the user and reverted from data center to the end user.
3a
3a. The transmission is complete, with the packet fetched back to the data center and connection is closed with an acknowledgement.
User is s accessing the application which is Under static / dynamic scan

Fig. 4: Data Flow of User Accessing an Web Application hosted on a Cloud Environment

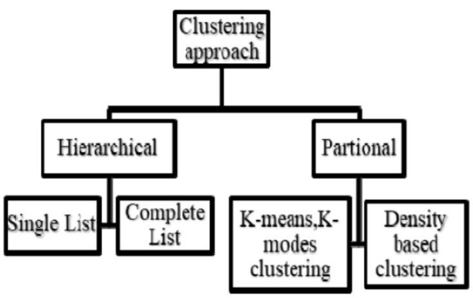

Fig. 5: Taxonomy of clustering techniques

III. Solution Methodology

Steps	Description
Step 1	Data Mining uses TCP/IP 3-way handshake data measuring the lag in the Application Response Time.
Step 2	Using the Hierarchical Clustering Technique in Analyzing Response time Data for more insights.
Step 3	Quantifying the User Response Data using Quantile-Quantile plot for distribution fitting, [9] and Kolmogorov-Smirnov Test. [10]
Step 4	Removal of Outliers Using the $\overline{x}\pm3\sigma$ limit which covers 99.7% of data, so that erroneous data can be avoided.
Step 5	Using the K-Means Clustering Technique for in depth analysis using the data gathered Before and After Outlier Removal.

TABLE I. Solution Methodology

Our solution methodology revolves around five steps as outlined in Table(I) in simple words. TCP/IP 3-way handshake data is used in measuring the number of lag user experiences. Using the Data Mining guidelines as provided in Figure(5), we will explore various data mining techniques and apply those to the data collected, and document the findings. That will allow us to conclude to determine the right approach to measure the lag in a user response time. The following sections of the paper provide such observations in detail. [6]

IV. Computation of Lag in Response Time Using TCP/IP 3-way Handshake Data

A. *Experimental Setup*

The Experimental Setup comprises hosting a web application in NGINX® running in a Docker container in the Google Cloud®. The application receives (Rx) POST/GET requests from the client. POST requests are primarily to save a file that is being uploaded to the cloud, and GET request is to check the size of the file which was uploaded. All POST/GET requests were executed using Python code. We ran the tests concurrently which involves uploading 100Mb and 300Mb files to the cloud. Complete interaction of GET/POST requests and responses were captured using TCPDump® and saved. The captured file provides the interaction taking place between the End user and the application server in the form of TCP 3-way Handshake data, as shown in Figure(4). The collected TCPDump® data is parsed for the elapsed time or Delta time to measure the elapsed time needed to upload a 100/300 Mb file on the cloud and to query the size of the uploaded file [8].

Cases	Description
Case 1	Control/Ideal Policy Setup with defender or CNNF/WAAS enabled without any Load
Case 2	Minimum Policy Setup with Defender running and CNNF enabled
Case 3	HardEnd Policy Setup with Defender running and WAAS CNNF enabled

TABLE II. Test Cases with their description

The elapsed time was computed for three different case studies, as presented in Table(II).

The elapsed times calculated in Cases 2 and 3 are contrasted with one another and the idle setup, i.e. Case 1, where no scan is being run. Traditional data analysis methods are used to compare these timings, and the results are presented. The Quantitative Analysis did not produce any tangible results, however the Test of Hypothesis (ToH) and Analysis of Variance (ANOVA) study demonstrates that there is a significant difference in the performance lag in the application response times when a Full scan or Minimal scan is conducted; for more information, see [1].

In this study, we are investigating the proper use of the data mining technique as provided to measure the amount of lag in user response and chart out appropriate data mining technique as explained in Figure(5).

V. Analysis of Application Response Times using Hierarchical Clustering

Using the data mining approach as outlined in Figure(5), we have started analyzing the lag time data collected using the Hierarchical Clustering Technique, Hierarchical clustering algorithm creates clusters in a hierarchy. It begins by assigning each data point to its cluster. Then, until one mega cluster is produced, the two closest clusters are combined into one cluster. It is well depicted by a respective Dendrogram (Diagram Representing a Tree), which shows the fusion of clusters [6][7].

The Hierarchical Clustering technique was applied to the lag response time data, the technique was applied in comparing Case 1 vs Case 2, Case 1 vs Case 3, and Case 2 vs Case 3 scenarios, and their Dendrogram was computed using Python code. The pictures of the Dendrogram obtained using the data mining technique of the Hierarchical clustering technique are presented in Figure(6).

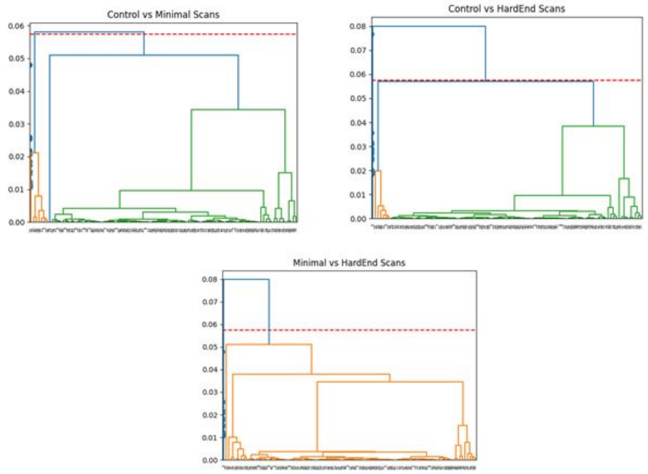

Fig. 6: Dendrogram Plots Comparing Minimal and Hard End policy with Control Policy

A. *Observation on applying Hierarchical Clustering Technique*

Based on Figure (6), presented we see there are outliers when the ideal case is compared with minimal and full scan, and when full scan is compared with minimal scan. Hence analysis of lag in response time data analysis using the K-Means Algorithm warrants an outlier check. In a nutshell, the following are the pros(III) and cons(IV) of using the Hierarchical Clustering technique on the lag in response time data as outlined below.

Hierarchical clustering technique is very helpful when:

Points	Description
(a)	It provides a visual representation of the cluster centers using dendrogram tree presentation which other techniques do not,
(b)	This technique can be easily applied in cases where the measure is non-euclidean in nature, unlike k-Means which uses only euclidean measure, very helpful when NLP(Natural Language Processing) is used, and
(c)	It provides information about outliers in the data centered around the cluster centers, a helpful tool to know if one is applying k-Means algorithm.

TABLE III. Pros of Hierarchical Clustering

The cons of the Hierarchical clustering technique are as follows:

VI. Removal of Outliers from the Data Collected

The data mining technique using Hirearichal clustering provided clues that there are outliers in the data, and hence direct application of K-Means will provide erroneous results. To remove outliers one has to estimate the parameters of the fitted distribution. We started with testing whether or not the

Points	Description
(a)	When used on huge data sets, visually looking at the dendrogram it produces, can result into inconclusive analysis as the graphs tend to be very cluttered,
(b)	Does not work properly when the data sets contain more than two cluster centers resulting from huge variability in the data around its centers, and
(c)	As the number of data points grows, the performance of the hierarchical clustering technique performs poorly in execution.

TABLE IV. Cons of Hierarchical Clustering

lag in response time data when applications are dynamically or statically scanned is normally distributed or not. We have used Quantile-Quantile (Q-Q) plot test [9] and Kolmogorov-Smirnov (K-S) test [10] to verify the same.

A. *Analysis of Application Response Time Data using Q-Q Plot*

As seen in the Figure(7) the Q-Q plot obtained for all three data sets viz (Control policy: Case 1, Minimal scan policy: Case 2, and Hard-end policy: Case 3) is in a straight line, all three data sets of response time captured is normally distributed since these Q-Q plots are at a 45°angle, states, the data is proportionally skewed [9].

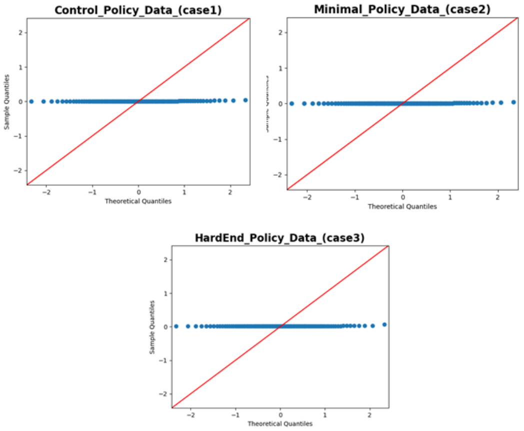

Fig. 7: Q-Q plots of the Data in three Test Cases

B. *Analysis of Application response time data using K-S Test*

The Kolmogorov-Smirnov(K-S) test is used in comparing a sample with a reference probability distribution, or to compare two samples. We have performed the K-S test on the three data sets used previously, with reference probability distribution being normally distributed. Test results from the K-S tests conducted is provided in the following Table(V)[10].

It can be seen the critical value obtained using the K-S test is greater than the normal test statistic used. Hence it is confirmed with 95% confidence level that our data comes from a normal distribution.

Data Set	Statistic and p-value	K-S test result
case 1	statistic=0.394, pvalue=1.623e-14	Normal Distribution
case 2	statistic=0.385, pvalue=7.131e-14	Normal Distribution
case 3	statistic=0.422, pvalue=1.009e-16	Normal Distribution

TABLE V. Kolmogorov–Smirnov test results on the Data Sets

Knowing that the response time data captured fits a normal distribution is of great value because the $\overline{x}\pm3\sigma$ rule for normally distributed data covers 99.7% of the population. Hence it is wise to discard any data which is on either side of the tail of the normal distribution viz. $\overline{x}\pm3\sigma$, as outliers.

VII. Analysis of Application response time using K-Means Clustering

The response time data captured and analyzed this far was found to be normally distributed. As a result, we can remove the outliers from the response time data collected using the $\overline{x}\pm3\sigma$ rule. A data mining analysis using k-Means was conducted on the response time data collected for the test case scenario as previously defined in Table(II).

The K-Means analysis is applied on (Idle scenario vs Minimal scan), (Idle scenario vs Hard-End scan), and (Minimal scan vs Hard-End scan) datasets with or without outliers. This analysis was conducted to see the effect of a shift in cluster centers before and after removing outliers, and then compute the impact by measuring the response time lag of an application, using the mapped cluster centers information.

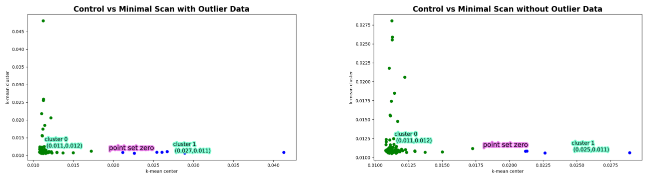

Fig. 8: Case 1 vs Case 2

Figure(8) presents the K-means analysis of Idle scenario vs Minimal scan response times before and after the removal of the outliers. The following can be inferred from the analysis, that yes!, there is an impact on the response time when minimal scan is done, and it is to the order of .0161 seconds (before outliers removed) while uploading a file of 300 Mb and query the file existence. The lag in response time reduces to 0.0132 seconds when outliers are removed from the data , thus an overall decrease of .000029% in the lag in response time is measured.

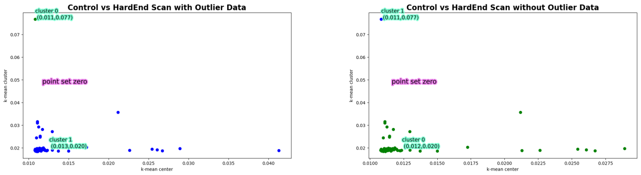

Fig. 9: Case 1 vs Case 3

Figure(9), presents the K-means analysis on idle scenario vs hard-end scan scenario, before and after the removal of outliers. Based on the results of the analysis, yes! there is an impact on the response time when a hard-end scan is performed, and it is to the order of .0568 seconds (before outliers are removed) while uploading a file of 300 Mb and querying the file's existence. The lag in response time has neither increased nor decreased when outliers are removed.

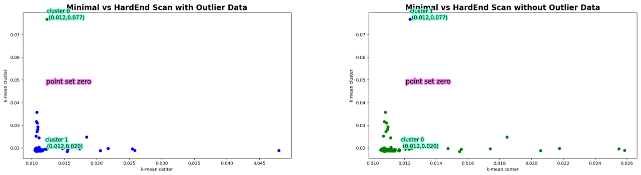

Fig. 10: Case 2 vs Case 3

The Figure(10), presents the K-means analysis of the Minimal scenario vs Hard-End scan scenario, before and after the removal of outliers. The following can be inferred from the analysis, yes! there is an impact on the response time when a Hard-End scan is done when compared to a minimal scan, and it is to the order of .0568 seconds (before outliers are removed) while uploading a file of 300 Mb and querying the file's existence. The lag in response time has reduced to 0.0567 seconds when outliers are removed and an overall decrease of .0000001%.

The complete summary of this results are tabulated in Table(VI) for a quick view.

Cases	Response Time Impact	Change in the Response Time	Change in Response Time (after outlier removal)	Response Time Impact Difference (%)
Case 1 vs Case 2	Yes	0.0161 sec	0.0132 sec	-0.000029%
Case 1 vs Case 3	Yes	0.0568 sec	0.0568 sec	0%
Case 2 vs Case 3	Yes	0.0568 sec	0.0567 sec	-0.000001%

TABLE VI. Impact observed on application response time for Respective Cases

The analysis can conclude as follows, that there were no outliers found when the hard-end scan was performed, as the response time is longer compared to the minimal scan where there were outliers. The findings on the lag in response time did not change much before and after the removal of outliers.

This points to further study on testing with various load sizes and comparing the lag in response time.

VIII. RELATED WORK

From open source to commercial software, security is a challenge that affects both. Therefore, there is a very high necessity to identify security issues in source code early on in the development process. Static and dynamic scans are two complementing strategies used to find security flaws. In addition, selecting the appropriate clustering technique and doing a thorough study is crucially important to produce the best results.

Yousif et al. [3], describe the advantages of Cloud computing being an advanced technology, which offers compute and store services in a pay-as-you-go manner. An in-depth discussion is made on the Cloud environment and its useful uses and application. The study using the k-Means clustering technique proposes a strategy that seeks to place the virtual machines allocated to the tasks from complemented groups or clusters on the same physical machines. Such placement prevents competition for the resources of the same physical machine, which may enhance system performance in the cloud data center. Besides k-Means, density-based clustering was also involved in the study. The software used is WEKA, which comprises machine-learning algorithms.

Bokyo et al. [4] describes a concept of cloud storage offered. Integrated data mining is being used for extracting potentially useful information from unprocessed data. The methods of data analysis are quite important with cloud computing. He discusses one of the most important issues that should be considered in cloud storage, which is quick access to the data stored in it. The approach they proposed involved the implementation of a hierarchical clustering algorithm in the cloud-data centers for the organization of data according to their type. Their proposed method has shown a set of advantages since it provides fast access to data, statistics on the use of disk space in the cloud show scalability, and helps in the analysis of large amounts of data that are inhomogeneous.

Lin et al. [5], discusses the growth of data, and the traditional clustering algorithms running on separate servers, which don't meet the demand. And also discusses more researchers implementing the traditional clustering algorithms on the cloud computing platforms, specifically for the K-means clustering. And explains the instability caused by the random initial centers created. This paper proposes a K-Means algorithm, which works on optimized initial centers, which improves the stability of this algorithm and the results have proven to improve the accuracy of the test set.

Gulati et al. [6], discusses the various clustering algorithms like partitional clustering, hierarchical clustering, density-based clustering; Grid based clustering, and their time and space complexities. He also discusses that partitional clustering algorithms are very useful when the clusters are of convex shape having a similar size and the number of clusters can be identified prior. Due to the disability in predicting the number of clusters in advance, Hierarchical clustering algorithms are used. They divide the dataset into several levels of partitioning called dendograms. These algorithms are very effective in mining but the cost of formation of dendograms is very high for large datasets. Besides this Density based clustering techniques are very useful in mining large datasets because they can easily identify noise and can deal with clusters of arbitrary shapes.

Arora et al. [7], talks about the advancing era of big data and how difficult it has become to analyze huge amounts of data. He talks about data mining as a technique that can be used to extract hidden and valuable information from data. Clustering being one of the major techniques used for data mining, this paper discusses the current data mining clustering techniques such as k-Means, Hierarchical Clustering Techniques, Other Partitioning Techniques, Density-based Clustering Techniques, and Generic Clustering Techniques. Under these techniques, several algorithms are discussed in detail.

Singh et al. [8], discusses a variety of contemporary methods for assembling big amounts of data. It discusses techniques for information extraction from data that are systematically analysed. Among these data analysis techniques, clustering analysis, specifically k-Means, is popular. In this study, the Hierarchical clustering technique and k-Means clustering are used (agglomerate). These methods' benefits and drawbacks are described. The findings of the study demonstrate that k-Means are more effective. The performance of these strategies was calculated based on accuracy and execution time using the data mining program WEKA.

IX. Preliminary Results and Conclusions

Our Contribution to this paper is threefold.
cons of using K-Means and Hirearichal Clustering techniques

First a methodology of how to collect response time using TCP/IP Sync and 3-way Handshake is presented and it works well in computing the 3-way handshake data for both on-cloud and on-prem applications. Once the data was collected we have shown how this data can be analyzed using ToH/ANOVA, and other statistical tools of Data Mining. Out of this we have compared and shown how to effectively use Hierarchical and K-Means Clustering techniques to measure the lag in the response time. Also provided analysis, pros, and on the application response time data collected using TCP/IP 3-way handshake protocol.

Effect of Measuring Lag in response time Using Data Mining Techniques

- k-Means is an extremely nice tool when there are no outliers present. One has to eliminate the outliers by using some data mining and data analytic techniques. In our case, we found the data to be normally distributed as a result removal of outliers was easy. But further tests are needed to study varied response times to see the impact of outliers to measure the lag in response time.
- Hierarchical analysis using dendrograms is a useful tool to check for outliers. It does not provide a better view when the cluster centers are more than two, and the tree diagram representation gets cluttered.

In conclusion the amount of lag in the response time of an application when it is under static/dynamic scan can be measured using data mining techniques. Viz, the ToH, K-means Clustering techniques along with distribution fitting algorithm and their tests using Q-Q and K-S tests.

References

[1] M. Athamnah, M.F. Hussain, and S.S. Hasan, "Impact of Running Dynamic/Static Scans on the Performance of an App Running in a GKE Clusters" 2021 Second International Conference on Intelligent Data Science Technologies and Applications (IDSTA).

[2] A. Tomar, D. Jeena, P. Mishra, and R. Bisht, "Docker security: A threat model, attack taxonomy and real-time attack scenario of dos," in 202010th International Conference on Cloud Computing, Data Science 'S' Engineering (Confluence). IEEE, 2020, pp. 150–155.

[3] Yousif, S., & Al-Dulaimy, A. (2017, July). Clustering cloud workload traces to improve the performance of cloud data centers. In Proceedings of the World Congress on Engineering (Vol. 1, pp. 7-10).

[4] Boyko, N., Mykhailyshyn, P., & Kryvenchuk, Y. (2018). Use a cluster approach to organize and analyze data inside the cloud. ECONTECH-MOD: An International Quarterly Journal on Economics of Technology and Modelling Processes, 7.

[5] Lin, K., Li, X., Zhang, Z., & Chen, J. (2014, August). A K-means clustering with optimized initial center based on Hadoop platform. In 2014 9th International Conference on Computer Science & Education (pp. 263-266). IEEE.

[6] Gulati, H., & Singh, P. K. (2015, March). Clustering techniques in data mining: A comparison. In 2015 2nd international conference on computing for sustainable global development (INDIACom) (pp. 410-415). IEEE.

[7] Arora, S., & Chana, I. (2014, September). A survey of clustering techniques for big data analysis. In 2014 5th International Conference-Confluence The Next Generation Information Technology Summit (Confluence) (pp. 59-65). IEEE.

[8] Singh, N., & Singh, D. (2012). Performance evaluation of k-means and heirarichal clustering in terms of accuracy and running time. IJCSIT) International Journal of Computer Science and Information Technologies, 3(3), 4119-4121.

[9] Marden, J. I. (2004). Positions and QQ plots. Statistical Science, 606-614.

[10] Berger, V. W., & Zhou, Y. (2014). Kolmogorov–smirnov test: Overview. Wiley statsref: Statistics reference online.