JES
Journal of Engineering Sciences

Vol 13, Issue 05, MAY/2022
ISSN NO:0377-9254

EFFECT OF A DYNAMIC/STATIC SCAN ON THE RESPONSE TIMES OF AN APPLICATION RUNNING ON THE CLOUD

[1]SALWA SAYEEDUL HASAN, [2]MOHAMAD MISBAH UDDIN ZIA,[3]MOHAMMED SHOEB QURESHI, [4]Dr.MOHAMMED ABDUL BARI
[123]B.E STUDENT, [4]HOD & ASSOCIATE PROFESSOR
DEPARTMENT OF CSE
ISL Engineering College, Bandlaguda, Hyderabad.

ABSTRACT

The evolution of technology especially in the Cloud industry has generated a big shift from the traditional way in terms of the use of the IT resources and has also become increasingly important in our day to day lives. With advanced technology the risk of cyber-attacks to IT resources has also become a major concern. To prevent which, many types of scans such as static scans, dynamic scans, or firewall scans are used by companies to keep their resources/ data secure. With the evolution of Big Data, it has become more vulnerable to attacks than before.

This study is an effort to analyze the effect of an application response time, when the application is hosted on a cloud and it is exposed to security scans, whose nature can be dynamic/static scan or even firewall scans. We study this performance issue by the measure of the response time, and it is done by the means of Delta time which is associated with the TCP/IP 3-way handshake data.

The real-time data of a user application is collected which is hosted on a Google® cloud, whose Delta times are analyzed using statistical methods, such as mean, skewness, etc. yielding no definitive answer. Further by analyzing using ToH and ANOVA tests however resulted in showing a difference in Delta times when the application was exposed to full scans.

This is in depth analyzed using various clustering techniques, namely K-Means and Hierarchical Clustering techniques. A comparison between which, resulted in the discovery of the outliers. These are then removed and the results are computed showing a change in the response times, prior and post the outlier removal.

Based on the tests conducted, we found that there is not a significant difference in running dynamic or static scans on the application or the data and its impact is not statistically significant to the performance, and while the K-Means Clustering technique is sensitive to outliers, the hierarchical technique validates the choices of clusters and gives information regarding outliers.

1. INTRODUCTION

With the evolution of cloud computing, several other factors have also enhances such as Security, which was a major concern of people while dealing with IT assets. To provide a safeguard to which, the security engineers have to scan those entities constantly, thus allowing companies to be free from cyberattacks, which can harm the company's infrastructure.

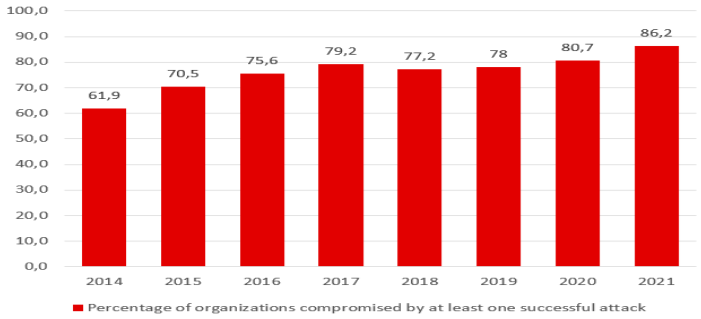


Figure 1: Graph depicting the increase of cyber attacks

As seen in the graph, due to lack of security measures takes, many organizations get effected due to cyber-attacks. And with the advancement of technology, this number is increasing day by day and remains a constant threat.

Modern organizations need to regularly evaluate their own systems so administrators can close holes to improve security. One way organizations can do this is to proactively run their own vulnerability scans to discover issues with systems, so the issues can be remediated before an attacker takes advantage of them.

The Static scans are those that will find vulnerabilities in the OS environment, the GIT repository where the code is stored, whereas the Dynamic scans are done while the application is running tracing the traffic of HTTP/HTTPS/REST/WSDL requests which comes in. In simple words, the static scans are a test of the internal structure of the application, rather than functional testing.

While the dynamic scans, examines in the running state and tries to manipulate it in order to discover security vulnerabilities. The other type of scan is firewall scan, whose purpose is to restrict unauthorized traffic from different layers of the network. Port scanning is a common technique used by firewalls in order to prevent threats from entering a network, which is based on Open or Accepted, Closed or Denied or Not Listening and lastly Filtered, Dropped or Blocked.

These test data is the user interactions with the IT resources of a company, and the response messages of those HTTP requests made are recorded. The collected response times are then compared and analyzed within our described scenarios. Further analyzing the data using various tools and tests, to establish if there exists a significant impact or not on the application. If the impact exists, then quantify the amount of impact on the application response time and make statistical inferences on the quality/quantity of the effect.

The following section focuses around the two common scenarios, portraying the working of the traditional approach of a user accessing an application on the perimeter. This is compared to the second scenario which portrays the user accessing the application on a Cloud Environment.

II. PROBLEM STATEMENT

Our study is to analyze the effect of an application response time, when the application is hosted on a cloud and it is exposed to security scans such as a. Dynamic scan, b. Static scan, and c. Firewall scan also known as port scan.

The user assessing the applications has also changed with time. Previously, the applications were hosted on the perimeter i.e. the secure data center of the companies. But post the evolution of cloud, or other technologies the ways to retrieve data has also changed. The user access any website of application might observes a change in

getting back its response. And this may be due to the multiple layers of security layers, such as authentication/authorization processes, SSO (Single sign-on), Firewalls, and other security tools. The security scans are crucial when it comes to the security. These are of many types such as static scans, dynamic scans, vulnerability scans, firewall scans, network scans, etc.

The measures are mainly used to authenticate the user upon using the service or application in order to provide a more secure environment. These security methods used are continuous and constant processes that can have an impact on the performance of the application in terms of responsiveness, thus making the consumer who is using the company's services experience sluggish response.

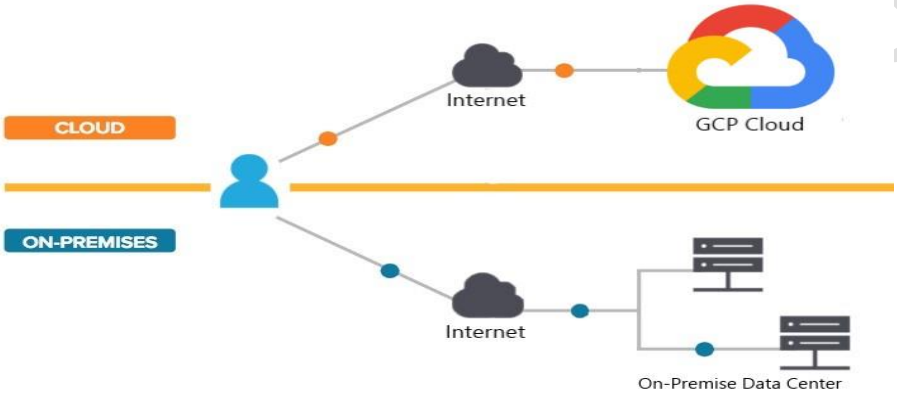


Figure 2: On Cloud & On-Premises operation

This problem statement is analyzed in depth using the two scenarios, the first one portraying the working of the traditional approach of a user accessing an application on the perimeter. This is compared to the second scenario which portrays the user accessing the application on a Cloud Environment; these are discussed in brief below.

II.A APPLICATION IS HOSTED ON THE PREM

The application response time is calculated by the exchange of the TCP packets between the end user and the server. The interaction taking place is the TCP 3-way Handshake, which is measured as the access request traverses from the segments as shown in the figure below.

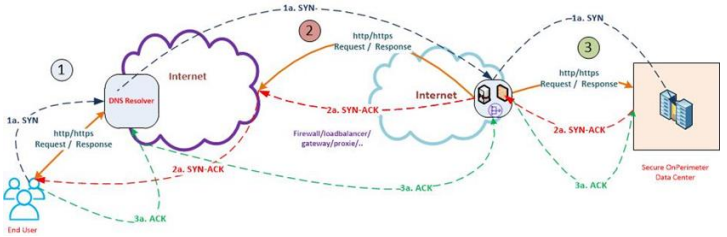


Figure 3: User Accessing a Web Application Hosted on the Company's On-Prem Data Center

This is a mode of communication between the users who are accessing the application/service with the server which is present in a company's data center. This is considered more of a traditional approach where the user sends a HTTP request, to access an application or website which is hosted on www.xxx.com, it firstly reaches the Internet which is connected to it, via the DNS Resolver. Via passing through various security mechanisms such as Firewall, load balancer, Gateway etc., the user's request

Vol 13, Issue 05, MAY/2022
ISSN NO:0377-9254

reaches the secured On-prem company's data center. Eventually, the HTTP response is reverted back to the user.

II. B APPLICATION IS HOSTED ON THE CLOUD

In a similar way as the On-Prem environment, the interaction taking place is the TCP 3-way Handshake, which is measured as the access request traverses from the segments as shown in the figure below.

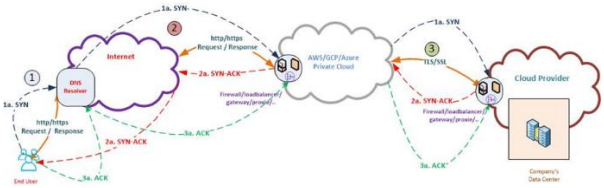


Figure 4: User Accessing a Web Application hosted on a Private Cloud Environment

In this scenario, as it is a Private cloud environment, there is an additional cloud and security mechanisms which offer more security to the environment. Here, the user communicates by sending the HTTP request, which reaches the Internet Cloud, via the DNS Resolver. Then it passes through the Firewall/Load balancer/Gateway etc., and reaches the Private Cloud, as the web application is hosted there. After which the TLS/SSL connection hits the Cloud provider which provides a secured connection throughout the process. By the second pass through the Security components, the company's data center is reached. Eventually after which the response is reverted back to the user, who is accessing the application.

III. LITERATURE REVIEW

In (A. Aggarwal and P. Jalote, 2004) the author proposes a methodology that integrates the two approaches in a complementary manner. It adopts the strengths of the two and eliminates their weaknesses. This model uses the strengths of the two analysis approaches i.e. Dynamic analysis, though accurate, but requires lots of test cases. In our model, we need fewer test cases for dynamic analysis because the coverage area is reduced. Only suspicious functions are handled by the dynamic analyzer. On the other hand, static analysis, though fast and has no runtime overhead, are not thorough enough in their analysis.

Therefore those functions which can be handled by static analyzers with higher precision are passed to them for analysis. This paper focused only on the buffer overflow vulnerability and still the idea of integrating the two analysis approaches can be used for detecting other vulnerabilities also like race conditions, integer errors, etc.

The IT assets will comprise of containerized resources for the applications, such as Docker, Google Kubernetes Clusters etc. The last decade has observed a dramatic increase in the development and use of virtualization technologies. Hence, the demand for an efficient and secure virtualization solution has also increased. We have also seen a large number of such solutions emerged which can be classified into two major classes' i.e. hypervisor-based virtualization and container-based virtualization which is also known as OS-level virtualization.

Container technologies have been around for a very long time but Docker is a relatively new and the most dominant candidate among all the other technologies. Along with so many advantages, it has a few disadvantages as well in which its security is the primary and the most crucial concern.

In this paper (A. Tomar, 2020) authors proposed a threat model for Docker with all the possible attack scenarios in Docker-based host systems. It has many advantages like speed, portability, density, and rapid delivery. Along with this, the paper gives a detailed classification of the several

attack scenarios that can be derived for docker security. The DoS attack has also been performed in the Docker environment (using nmap and hping3) with different case studies analyzing the data.

The main focus of the work has been on intrusion detection for container-based systems. The exploration of the capacity of state-of-the-art intrusion detection algorithms to generate stable and complete profiles of containers running applications and evaluation of their effectiveness in the container-based system was carried out. The preliminary results focused on studying the convergence capacity of the algorithms Sequence Time-Delaying Embedding (STIDE) and Bags of System Calls (BoSC), commonly used for intrusion detection. The further work focused on defining a methodology to evaluate and compare different intrusion detection algorithms in a representative and fair manner. The production of meaningful and representative datasets in the context of container-based applications, which were not available.

In this work (W. Findlay, D. Barrera, and A. Somayaji, 2021), showed various vulnerabilities or soft belly in the context of Container Security. The Linux containers have become the preferred unit of application management in the cloud, forming the foundation of Docker, Kubernetes, Snap, Flatpak, and others. This includes just the binaries, libraries, and configuration files needed by an application, containers enabling simplified deployment of vendor-packaged applications, rapid horizontal application scaling, and direct developer-to-production DevOps workflows, all without the overhead of hypervisor-based virtual machines (HVMs). This research also provides background on the classic and extended Berkeley Packet Filters (BPF and eBPF) and discusses the motivation behind a new container-focused security enforcement mechanism

Hence our study is a vital part to know while doing the static and dynamic scan of our IT assets, does it impact the performance of the very applications the IT assets supports.

IV. EXPERIMENTAL SETUP TO STUDY APPLICATION RESPONSE TIME

The test web application is hosted in a Google Cloud®, which was containerized using Docker containers with replica sets. This container platform comes in to provide the security, monitoring and logging, enterprise storage and networking. It is one of the fastest approaches when it comes to containers.

The following sections gives an in depth insight regarding the experimental setup involving the network connectivity, quantification of application response time and, the Transmit/Receive (Tx/Rx) of HTTP data for application response time analysis.

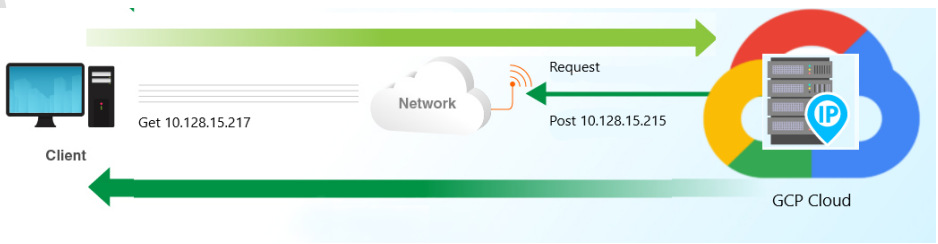


Figure 3: Continuous query exchange between the GCP host and the requesting client

The Transmit/Receive (Tx/Rx) of the HTTP data takes place using the HTTP get/post,

between the IP interfaces. These are HTTP request and response methods enable the communication between the clients and servers.

This is made possible by the Hypertext Transfer Protocol (HTTP) which is designed to enable communications between clients and servers and it works in the form of request-response protocol. The four methods commonly used are GET, POST, DELETE and PUT. One can execute the HTTP requests using a curl command and get a response back to the requests thus creating a Tx/Rx of HTTP data response.

The application hosted, takes a file via HTTP POST, so the test bench created is to continuously upload 300Mb file to the application, and one can also do HTTP GET, we have endpoints for HTTP GET and HTTP POST in our application.

V. DATA MODELING OF APPLICATION RESPONSE TIMES

The Data modelling comparison is executed between three scenarios, viz

1. Idle Policy Scenario vs Minimal Policy Scenario,

2. Idle Policy Scenario vs HardEnd Scenario, and

3. Minimal Policy Scenario vs HardEnd Scenario,

Where the Idle Scenario is the control environment tested against, the other anomalous scenarios. Here the test exemplars are used as load where the scans are running. The second scenario is the Minimum Policy setup with scan enabled with a defender while the test exemplars are used while the defender is running. The third scenario is the HardEnd policy in which scans with the defender runs while the test exemplars are used.

Under this modeling, the various tests can be applied once the normality of the distribution is confirmed. The next section explains in brief about the test for normality and various other Statistical and Hypothesis tests conducted.

VERIFICATION OF APPLICATION RESPONSE TIME AS NORMALLY DISTRIBUTED

It is important to first understand the collected response time dataset and its type of distribution. To do that, we perform the Q-Q plotting. It is a type of statistical measure which graphically analyzes and compares two probability distributions by plotting their quantiles against each other.

The Q-Q plots comes into picture when one wants to know whether the distribution is normal or not, and can tell the type of distribution just by looking at the plot.

A normal distribution is one in which the values are evenly distributed both above and below the mean, and it can be described by two values: the mean and the standard deviation.

Our dataset is passed through the Python code for the Q-Q plot, and the following conclusions can be made. As seen in the image below, we obtain a straight line of data-points depicting that the data is normally distributed but proportionally skewed.

JES
Journal of Engineering Sciences
Vol 13, Issue 05, MAY/2022
ISSN NO:0377-9254

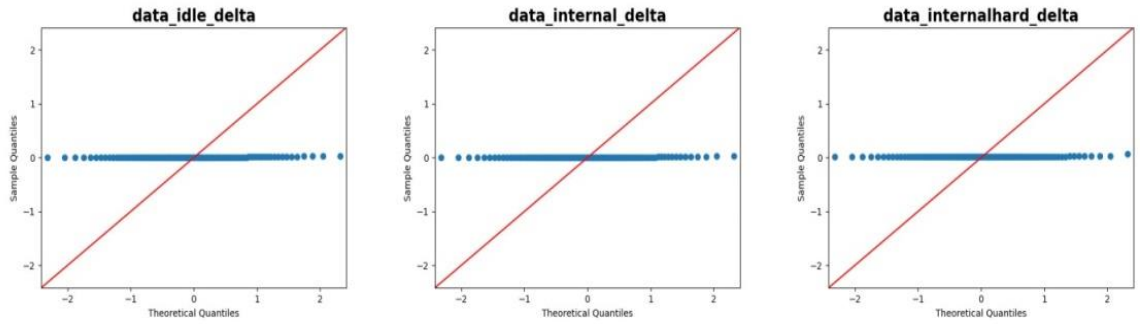


Figure 4: Q-Q plots for columnar data

APPLICATION RESPONSE TIMES ANALYSIS USING BASIC STATISTICS

Statistics is basically a science that involves data collection, its interpretation and finally data validation. Statistical data analysis is the process of performing various statistical operations to get the desired output.

Many basic statistical measures were performed on the data such as (a) Mean which is the average of a given data set, (b) Standard Deviation which is the amount of variation or dispersion of a data set of values, (c) Kurtosis which is the measure of the peakedness or flatness of the distribution, (d) Skewness which is the measure of how much the probability distribution of a random variable deviates from the normal distribution and (e) Confidence interval which is the measure the degree of uncertainty or certainty in a sampling.

Conclusion on Quantitative Analysis: No Definitive Answer

Hence, we will investigate the data collected using Test of Hypothesis (ToH) and Analysis of Variance (ANOVA).

ANALYSIS OF APPLICATION RESPONSE TIME: USING TOH & ANOVA

Test of Hypothesis

T-test compares the means of two independent groups in order to determine whether there is statistical evidence that the associated population means are significantly different. This test works well when dealing with two groups, but when we compare more than two groups at the same time this test may result in increasing the chances of false positives.

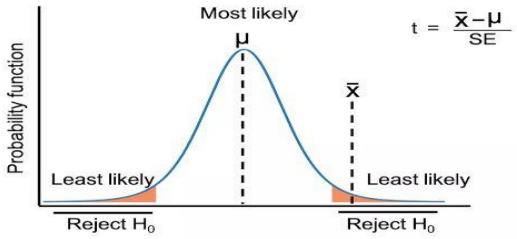


Figure 5: T-test

Analysis of Variance (ANOVA)

The analysis of variance or ANOVA is a statistical inference test that lets you compare multiple groups at the same time. It checks

for one or more factors by comparing the means of different samples.

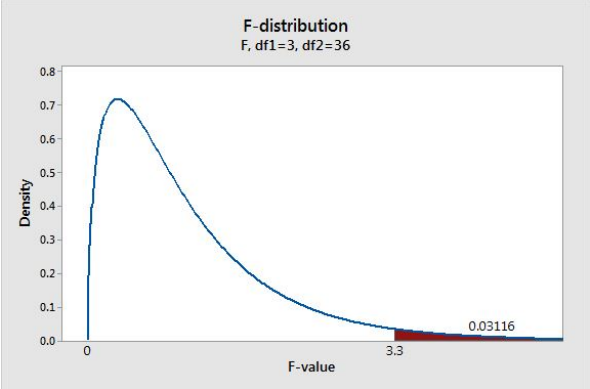


Figure 6: F-test (ANOVA)

We make the use of ToH & ANOVA (T test 95% confidence, 2 Tail test) and obtain the following results.

Result:

Idle Policy Scenario vs Minimal Policy Scenario	No Significant Difference
Idle Policy Scanrio vs HardEnd Scenario	Significant Difference that the means different
Minimal Policy Scenario vs HardEnd Scenario	Significant Difference that the means different

Table 1: Conclusion for ToH & ANOVA

Conclusion on Analysis using ToH & ANOVA

We can conclude that the TCP/IP 3-way handshake times are statistically different when the Idle policy is compared with HardEnd policy and when Minimal Policy is compared with Hard End policy, see table 1.

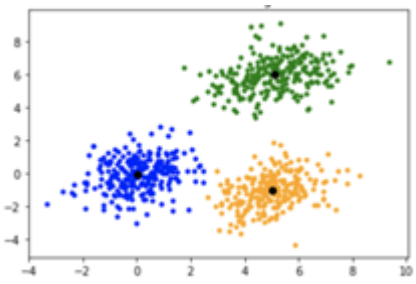
ANALYSIS OF APPLICATION RESPONSE TIME USING CLUSTERING METHODS

In order to draw valid conclusions using the collected data, we use the clustering algorithms. These algorithms take the data and uses similarity metrics; they group the data which can be grouped to interpret valid information. And are used to partition the data points based on certain similarities, after which they can be analyzed and valid conclusions can be drawn from it.

We use the three scenarios as explained previously (5.1, Table-1), and apply the clustering techniques on it, to study the observed changes in application response time and gather concrete results.

ANALYSIS OF APPLICATION RESPONSE TIME COMPARING THE DATA SETS: USING K-MEANS CLUSTERING

Vol 13, Issue 05, MAY/2022
ISSN NO:0377-9254



The main objective of the **K-Means algorithm** is to minimize the sum of distances between the mean or standard deviation and their respective cluster centroids. In simpler words we can say that in this study we find the number of cluster centroids and we can find out how the data is clustered around it and using it we can figure out how far or close are the anomalies centered around their mean.

When it comes to the K-Means algorithm, its performance is usually not as competitive, compared to other clustering techniques because of the slight variations in the data could lead to high variance. Moreover, the clusters are assumed to be spherical and evenly sized i.e. to be of the same size, which may sometimes reduce the accuracy of the K-Means clustering Python results.

But the major issue with K-Means which remains constant is that, the number of clusters, i.e. the selection of the value of 'k'. As this is a pre-defined selection of number of clusters algorithm, this value must be decided at the beginning of the processing, and considering an ideal scenario, one might not always be aware of how many clusters to be considered right from the beginning.

As far as time complexity is considered, K-Means is linear in the number of data objects i.e. O(n), where n is the number of data objects.

Analysis of K-Means Clustering:

This analysis results in graphs which visualizes the K-Means clusters. These are clustered into two for each scenario. Upon inspection of our produced graphs, we can detect points distanced far away than usual, these are referred as outliers. The K-Means clustering algorithm is sensitive to them as their presence gives rise to erroneous results. As a result, the outliers must be detected and removed for optimal results.

This Analysis on the data sets of Idle Scenario vs HardEnd Scenario, and the Minimal Policy Scenario Vs Hard End scenario was conducted, which yields the following outputs.

Conclusion of the K-Mean Clustering Analysis:

The following conclusions were made, a. While running upload of files (300Mb), the measure of TCP/IP handshake increases significantly when WAAS is used and compared against Control Policy to an amount of .047 seconds. b. One can also say that the WAAS becomes effective when the upload size is quite large when compared against Control and Minimal Policies. It results in the following graphs and distances.

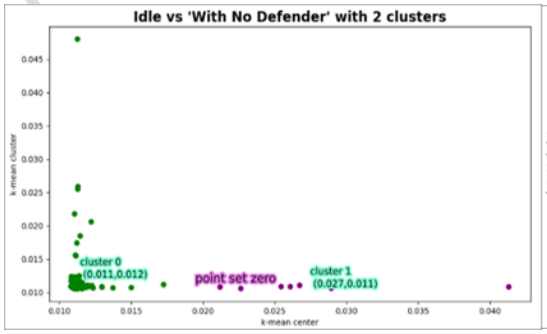


Figure 7: K-Means Clusters

Result:

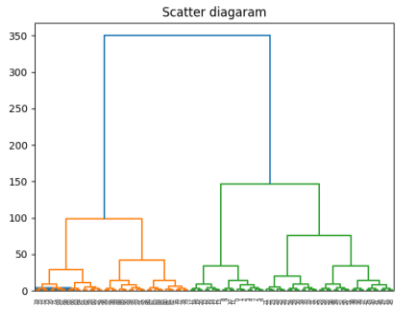
Cases for 300Mb Upload	Distance between Cluster centers
Idle Scenario vs HardEnd Scenario	0.0568 seconds
Idle Scenario vs Minimal policy	0.0161 seconds
Minimal Policy vs HardEnd Scenario	0.568 seconds

ANALYSIS OF APPLICATION RESPONSE TIME COMPARING THE DATA SETS: USING HIERARCHICAL CLUSTERING

Hierarchical Clustering algorithm is one as the name suggests, builds a hierarchy of clusters. It starts with all the data points assigned to a cluster of their own. Then two nearest clusters are merged into the same cluster, until one mega cluster is obtained. And can be best represented using dendrogram, which depicts that, merging of two-two clusters.

One can easily say that the gap caused in k-means clustering technique is bridged using the hierarchical clustering technique.

On the other hand, the K-Means is only applicable on only numeric data, while Hierarchical does not even require a distance, any measure can be used. The time complexity of the hierarchical clustering algorithm is quadratic i.e. O(n2). One can easily say, as the number of records increases, the performance of hierarchical algorithm goes decreasing and time for execution increases.



Analysis of Hierarchical Clustering: But at times, visually speaking hierarchical clustering will not always deliver concrete output when applied to large datasets. This technique works well if the number of clusters is two only. The most important feature of this technique is that it gives insight about the outliers. These are the data points that are very far away from other data points and these could be errors in the data recording or some special data points with very different values. To understand that in depth, we perform a process of outlier detection and then their removal. This is explained in brief

in the following sections.

Conclusion of the Hierarchical Clustering Analysis:

The output of the Hierarchical Clustering, is a plot formed using dendrograms. These dendrograms depict the hierarchy of clusters, based on a metric of Euclidean distance as shown in the figure below.

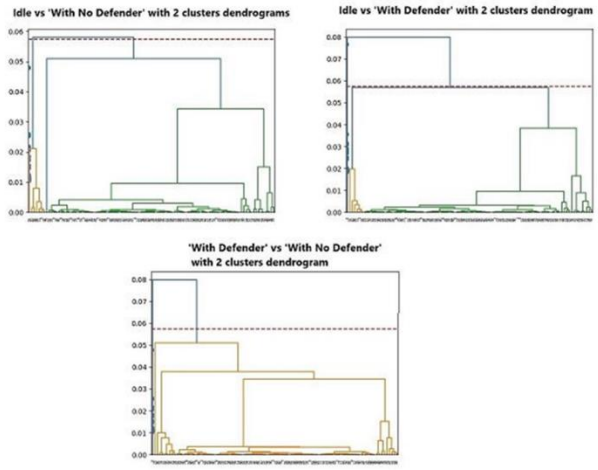


Figure 8: Hierarchical Dendrograms

VI. ANALYSIS OF APPLICATION RESPONSE TIME DATA USING 3-SIGMA RULE

The drawback of the K-Means clustering technique is that it is sensitive to Outliers, can be defined as noisy data or anomalies within our data that could lead to errors in the data analysis.

In-order to get optimal outputs it's really important to understand the number of outliers, remove them and then carry out the process again to get better outputs.

This can be done by first, the data points can be monitored and then eliminate them eventually after some iterations. Upon the confirmation in the previous section [6.1], stating that our data is normally distributed. We then perform outlier removal using 3-sigma so 99.7 percent of data perfectly fits into the normal distribution.

RESPONSE TIME ANALYSIS USING K-MEANS CLUSTERING POST OUTLIER REMOVAL

Post the outlier removal from the data using 3-sigma rule and new data being computed. The distance of the means via the K-means were computed again and found the following, along with the new graphs.

Conclusion on K-Means Cluster Analysis post Outlier removal:

Vol 13, Issue 05, MAY/2022
ISSN NO:0377-9254

Cases for 300Mb Upload	Distance between Cluster centers
Idle Scenario vs HardEnd Scenario	0.0567 seconds
Idle Scenario vs Minimal policy	0.0105 seconds
Minimal Policy vs HardEnd Scenario	0.0567 seconds

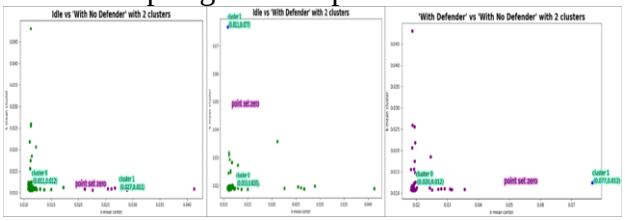
COMPARISON BETWEEN THE CHANGES OBSERVED IN THE APPLICATION RESPONSE TIME, PRE AND POST OUTLIER REMOVAL

The cluster analysis conducted to study the changes in the application response times, as observed can be categorized into two scenarios, viz. change in response time pre and post the outlier removal. The differences can be observed in terms of graphs and the distances between the cluster centers. These are shown in the following table.

1) Impact of Running Scans without Removing Outliers is

Cases	Distance between Cluster centers
Idle Scenario vs HardEnd Scenario	0.0568 seconds
Idle Scenario vs Minimal policy	0.0161 seconds
Minimal Policy vs HardEnd Scenario	0.0568 seconds

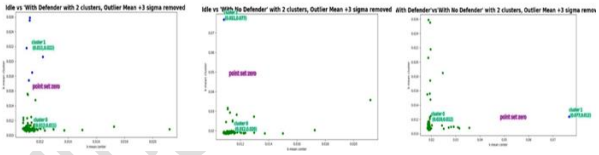
➢ Graphs generated pre outlier removal



2) Impact of Running Scans after Removing Outliers is

Cases	Distance between Cluster centers
Idle Scenario vs HardEnd Scenario	0.0567 seconds
Idle Scenario vs Minimal policy	0.0105 seconds
Minimal Policy vs HardEnd Scenario	0.0567 seconds

➢ Graphs generated pre outlier removal



Conclusion: We can observe that post the outlier removal the clusters formed are free from outliers and the distance calculated between the cluster centers, has also been decreased. More significant change has been observed in the second case of Idle Scenario vs Minimal Policy. The other cases have comparatively less change in their response times.

VII. CONCLUSION

When it comes to the security of the data, we conducted this study to understand the impact of running the vulnerabilities scans in the IT assets of a company and its impact on the performance of the applications which uses those IT assets.

In that context we used sample data from an application using curl commands and collected data while uploading of files 300Mb to the application, when the IT assets are in one of the three setups as mentioned previously. While doing so we collected TCP/IP three-way Handshake data and collected the TCP/IP Delta times.

We further used Data Analytical methods to analyze the result. Upon finding significant difference in the outputs, we used

Vol 13, Issue 05, MAY/2022
ISSN NO:0377-9254

Clustering Techniques. The K-Means clustering technique helped us in estimating the distance between the centroids. While the Hierarchical's dendrograms, validates our choice of the number of cluster to be chosen. But it fails in giving substantail evidence when exceeding more than two clusters. But it delivers required information revolving the outliers.

To make concrete findings based on our data we monitor our data and firstly produce Q-Q plots of the dataset to make sure that the data is normally distributed. This resulted in the confirmation of data being normal yet proportionally skewed. Following which the outliers were monitored and removed based on 3-sigma removal, so that 99.7 percent of the data lies within the distribution. Passing the cleaned data again via K-Means clustering resulted in graphs without the anomalies. And there was an overall impact on the application response times, when analyzed pre and post the outlier detection which was tabulated.

We can also conclude that running Minimal cans has no significant impact on the application performance, in turn the security of the application is enhanced when one runs HardEnd or Minimal Policy Scans, and the difference between running those scans is also quite minimal.

FUTURE ENHANCEMENT

Further work is required in this area to collect and analyze data while running this experiment in a multi-server distributed environment, where microservices run the applications. Then see the overall impact of running a vulnerability scan on the IT assets and its impact on the application responsiveness.

REFERENCES

1. L. Sthle and S. Wold, “Analysis of variance (anova),” Chemometrics and Intelligent Laboratory Systems, vol. 6, no. 4, pp. 259–272, 1989. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0169743989800954
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 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence). IEEE, 2020, pp. 150–155.
3. J. Flora, “Improving the security of microservice systems by detecting and tolerating intrusions,” in 2020 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW). IEEE, 2020, pp. 131–134.
4. W. Findlay, D. Barrera, and A. Somayaji, “Bpfcontain: Fixing the soft underbelly of container security,” arXiv preprint arXiv:2102.06972, 2021.
5. I. Vurdelja, I. Bla˘zi´c, D. Boji´c, and D. Dra˘skovi´c, “A framework for automated dynamic malware analysis for linux,” in 2020 28th Telecommunications Forum (TELFOR). IEEE, 2020, pp. 1–4.