International Journal on Perceptive and Cognitive Computing (IJPCC) Vol 10, Issue 1 (2024)
https://doi.org/10.31436/ijpcc.v10i1.458

Streams of Data Flow in Transmission Control Protocol (TCP) Request-Response Cycle Efficiency

Adamu Abubakar*, Zulkefli Muhammed Yusof
Department of Computer Science, KICT, International Islamic University Malaysia.

*Corresponding author: adamu@iium.edu.my
(Received: 2nd April 2023; Accepted: 12th January 2024; Published on-line: 28th January 2024)

Abstract— This study examines the complexities of data transmission in Transmission Control Protocol (TCP) Request-Response cycles, with the goal of improving overall efficiency. The research aims to enhance the efficiency of these cycles by studying the dynamic characteristics of data streams. An experimental analysis was carried out utilising a thorough examination of TCP streams of HTTP request-response cycle, with a focus on the complex interaction between client requests and server responses. This study utilises BlazeMeter and JMeter to analyse the effectiveness of TCP request-response cycles, specifically examining the dynamics of data flow. Significant variances were seen in performance indicators across a range of different settings. Analysis of certain experimental request-response scenarios reveals that consistently high response times leads to a persistent server load as well as resource limitations, which negatively affect the overall user experience. In contrast, certain request-response scenarios demonstrate a greater throughput, suggesting a more efficient data transfer capacity, whilst lower throughput scenarios in the experimental conditions indicate the possible bottlenecks as well as network problems. The analysis encompasses various thread groups, providing insights into error rates, response times, and throughputs. These findings enhance the understanding of the efficiency of the TCP request-response cycle and emphasise the elements that affect the flow of data during transmission sessions. Consequently, the research concludes that the data flow within TCP request-response cycles lacks a discernible pattern that can be utilised for training purposes in the field of Artificial Intelligence.

Keywords— Transmission Control Protocol (TCP), Request-Response Cycle, Data Flow, Efficiency, Performance Optimization.

I. Introduction

What is meant by the term "streams of data flow" as operationally defined by this research is the continuous movement or transfer of data that occurs within a system. Transmission Control Protocol, sometimes known as TCP, is a highly popular networking protocol that guarantees the transfer of data in a dependable and organised manner between devices that are connected to a network. Request-Response Cycle Efficiency is a measure that indicates how effective and optimised the process is. This process involves one entity, which is generally a client, sending a request to another entity, which is frequently a server, and the recipient responding in accordance with the request. Essentially, this implies that the emphasis should be placed on gaining an understanding of how data moves within the context of TCP, more precisely inside the request-response cycle, with a particular focus on efficiency.

In order to evaluate the effectiveness of the communication between devices, it necessary to comprehend how TCP manages communication between devices. It begins with the establishment of a connection via a three-way handshake, the implementation of a system that would monitor and enforce policies about the entire traffic within TCP, specifically, HTTP traffic is crucial. One of the goals is to analyse and assess HTTP traffic over TCP transmission session with the intention of recognizing and distinguishing a number of different types of attacks [1]. Similarly, policy enforcement is crucial to monitoring whether or not the stream of data flow is being followed [2].

The problem(s) or issue associated with the HTTP Transmission Session lies request-response cycle uncertainty. It has been proved through simulation tests that use real-world web access requests that the HTTP-session model is successful in defining the behaviour of web access and detecting attacks that are based on the web [3]. Another research problem lies with the fact that regarding web servers and websites, attackers might initiate DDoS attacks by inundating the target with a substantial volume of HTTP requests, with the aim of depleting the server's resources and bandwidth [4]. This might result in a server overload, resulting in its unresponsiveness and denial of service to genuine users. Although HTTP-based DDoS assaults are prevalent, attackers may employ a blend of

https://doi.org/10.31436/ijpcc.v10i1.458

these techniques to generate a more powerful and diverse assault.

The opportunities that streams of data flow in TCP request-response cycle efficiency provides lies with the prospects for enhancement. Analysing and comprehending the data flow patterns to identify opportunities for optimizing the TCP request-response cycle, improves the overall performance and responsiveness of online applications. Examining the data transmission within TCP streams can offer valuable information on the use of network and server resources. This information can provide guidance for enhancing resource allocation and utilisation to achieve greater efficiency. Having a comprehensive understanding of the data flow in the TCP Request-Response cycle is crucial for optimising performance, improving user experience, maximising resource utilisation, and addressing security concerns. This awareness ultimately enhances the overall effectiveness and efficiency of web applications and networked systems. That is why this research aims to examine streams of data flow in TCP request-response cycle efficiency.

In addition to this section, which provides an overview of the research, the rest of the paper is structured as follows: Section 2 provides an overview of the relevant research and studies. Section 3 presents the research methodology. Section 4 provides an analysis and presentation of the results. Section 5 provides the conclusions of the paper.

II. RELATED WORK

TCP is a crucial protocol for ensuring dependable data transport in computer networks. The efficacy of TCP in handling Request-Response cycles, specifically in the context of data transmission, has attracted significant attention in recent scholarly works.

Liang et al. [3] presents a novel framework for actively detecting HTTP assaults by using the HTTP-session model and evaluating the behaviour linked to such attacks. The framework analyses data in the form of HTTP requests and calculates the likelihood of anomalies for both specific session attributes and the entire session. The probabilities that have been calculated are given weights and combined to provide a definitive probability. This probability is then used to determine whether an HTTP session is considered an attack or not. The effectiveness of the proposed approaches is demonstrated in simulated tests using real online access records, showing high detection rates with little false positives.

Taking into account even brief interruptions in web-based applications can result in significant financial losses and damage to one's reputation in the market [4]. Machine learning methods have demonstrated potential in accurately identifying DDoS assaults with little false alarms and high rates of detection [5]. A test was conducted on TCP and HTTP characteristics of business intelligence transaction on a live network, which provides statistics and detailed analysis based on multiple traces and conversation [6].

A comprehensive analysis of web application session management approaches and HTTP session hijacking concerns has been conducted in the previously published literature. The session management system has been found to be susceptible to a number of vulnerabilities and exploits. One of the most common types of attacks is known as session hijacking, which occurs when an unauthorized person gains unauthorized entry into a user's session. The usage of default session management IDs and websites that are not constructed appropriately are the primary factors that lead to vulnerabilities in session hijacking [7-8]. It is recommended to use secure session management solutions in order to prevent session hijacking. Some examples of these tactics include the utilization of secure HTTPS connections, the utilization of HttpOnly and Secure cookie characteristics, and the utilization of one-time cookies [9-10]. By protecting the generation, destruction, and transmission of session tokens, these methods improve the security of session tokens and make it more difficult for attackers to hijack sessions. The implementation of these security measures is very necessary for online applications in order to protect user sessions and reduce the likelihood of illegal access [11].

Shafique et al. [12] presents a comprehensive examination of security weaknesses in Multipath TCP throughout the process of establishing a connection. These vulnerabilities include Man-in-the-middle attacks, ADD-Address assaults, Denial of Service (DoS) attacks, SYN floods, and other cyber-attacks. The research also analyses the existing remedies that have been developed to address these security issues. In addition, the study evaluates a compilation of recently implemented Multipath TCP security measures aimed at enhancing the protocol's efficiency.

Chen et al. [13] suggest a security method to protect against TCP session hijacking by incorporating encryption techniques inside the TCP protocol. These algorithms include RSA-based cryptography technology, RSA-based signature technology, DH key exchange algorithm, and HAMC-SHA1 integrity verification technology. The proposed security technique seeks to efficiently protect against TCP session hijacking, a prevalent network attack that leverages inherent flaws in the TCP protocol. The study proposes the integration of these encryption methods into the TCP protocol as a means to get safe authentication, hence resolving the existing deficiency of secure authentication in the protocol.

Cao et al. [14] investigate the impact of packet loss on the impression of performance by end users. Even a slight

https://doi.org/10.31436/ijpcc.v10i1.458

occurrence of packet loss has a substantial impact on the noticeable performance of a website, highlighting the cruciality of preventing packet loss rather than focusing on increasing capacity. The transfer time experiences exponential growth as the packet loss rate increases, although the impact of varying queue sizes is insignificant in comparison to packet loss. Enhancing the amount of accessible bandwidth yields diminishing benefits for regular web browsing activities. The data set has a strong fit with an exponential curve, consistently observed in all recorded data sets. For extremely low payload quantities, the impact of available bandwidth on the results is negligible.

The digital ecosystem and Internet infrastructure in India could both use an Internet measurement network. [15]. Hoque et al. [16] studied the influence of system improvements on the performance of networks and applications, as well as the impact of VPN-based solutions on network and application performance. Research indicates that VPN-based technologies, including Lumen, PrivacyGuard, and Video Optimizer, can result in unclear network performance measures and reduce application performance. Packet losses can have diverse impacts on metrics of averaged neural signals. Simulations demonstrated that long-term signals were generally resistant to the impact of packet losses, with the exception of timing offsets [17]. In contrast, signals with a short timescale exhibited substantial dispersion and weakening as a result of packet losses. These findings indicate that the capacity to distinguish minor discrepancies associated with various task conditions may be impaired when handling short-duration signals that are influenced by packet losses.

III. RESEARCH METHODOLOGY

In order to shed light on the complexities of TCP stream dynamics during transmission sessions, this study methodology makes use of a strategy that combines quantitative simulations with qualitative analysis. It is guaranteed that an in-depth understanding of the research goals support the use of the research methodological approach selected. Furthermore, through the utilization and the incorporation of Blaze Metre and JMeter as simulation tools justify the feasibility of the selected research methodology.

A. Experimental Scenarios

The Gombak Campus of the International Islamic University Malaysia (IIUM) was the location where the experiment was carried out. The purpose of this experiment is to investigate the interaction that takes place between a server that is situated in Portland, Oregon, United States, and source of the request (see Figure 1). For the sake of this experiment, the server that was selected in Portland functioned as the target.

Figure 1 illustrates the path followed by the transmission session, presenting the sequence of request-response cycles involved. The figure emphasizes the complex process of communication, demonstrating how the data passes via numerous hubs before finally reaching the server. Although the large chain of hubs adds complexity, this means that, as far as different hubs are involved, then transmission issues can be expected.

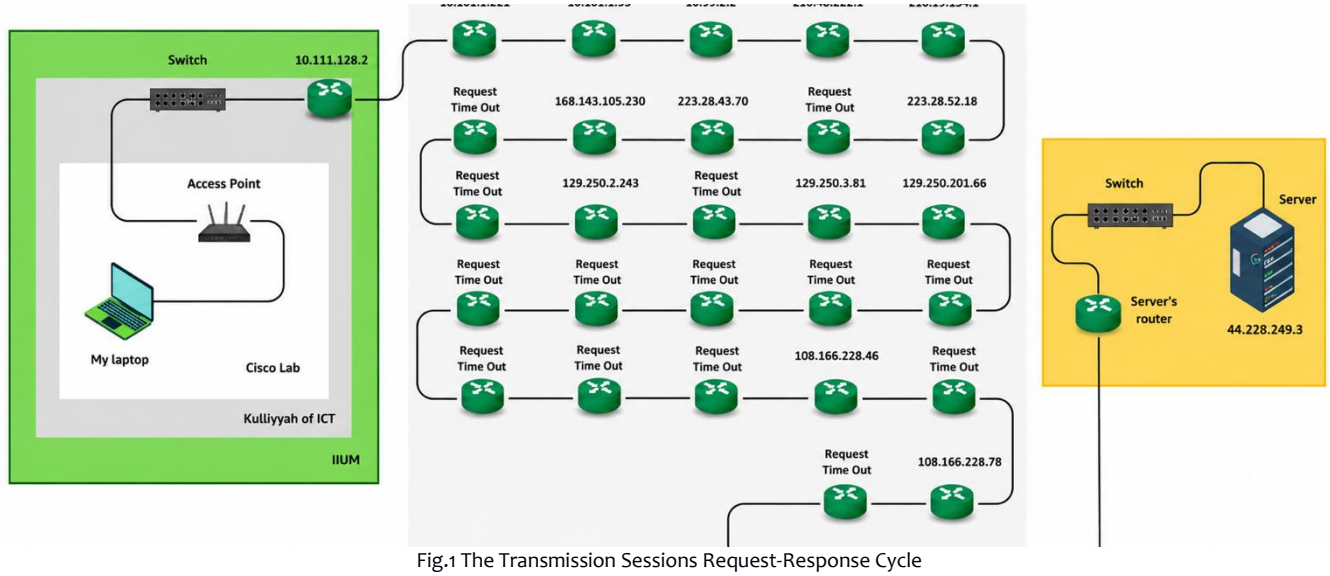

Fig.1 The Transmission Sessions Request-Response Cycle

The use of BlazeMeter was implemented in order to record the specifics of the communication that took place between the IIUM Gombak Campus and the server located in Portland. The tool known as BlazeMeter is designed to make it easier to record request-response pairs that occur during interactions with a server. Users are able to evaluate

https://doi.org/10.31436/ijpcc.v10i1.458

the effectiveness of the communication process when it comes to performance, behaviour, and efficiency.

JMeter was utilized to analyse the transmission sessions that were acquired. The JMeter tool is a performance testing tool that offers in-depth analysis of a variety of parameters pertaining to network communication. In addition to supporting both HTTP and HTTPS protocols, Apache JMeter is a popular open-source application that is used for performance testing and load testing. An analysis of the request-response interactions that were recorded by BlazeMeter was carried out with its assistance in this particular scenario. The utilisation of JMeter made it possible to conduct an exhaustive analysis of the data, which assisted in determining the effectiveness and efficiency of the communication that took place between the IIUM Gombak Campus and the server that was situated in Portland, Oregon of the United States.

In a nutshell, the experiment consisted of visiting a server located in Portland from Kuala Lumpur. BlazeMeter was used to record the interactions between the request and the response, and JMeter was utilised to do an in-depth analysis of the transmission sessions that were recorded. The objective of the experiment was to acquire a better understanding of the performance of the communication link between the sites that were being described.

B. Experimental Parameters

The key parameters of the experiment included the locations (the server in Portland and the IIUM Gombak Campus). Furthermore, the purpose of evaluating the performance of communication, the tools that were used (BlazeMeter and JMeter), the precise specifications of the server, and the emphasis placed on analysing the recorded transmission sessions for the purpose of performance evaluation. Therefore, the following are the parameters used:

Label: The identifier or name of the sampler (such as HTTP request or JDBC request) that was run.

Samples: The cumulative quantity of samples dispatched throughout the experiment.

Average: The mean duration of all samples' response times.

Median: The response time that falls exactly in the centre when all samples are arranged in ascending order. The median is the central value when all reaction times are arranged in ascending order.

90% Line: The response time below which 90% of the samples fall. This is also known as the 90th percentile.

The 95% Line is a representation of the 95th percentile, similar to the 90% line.

99% Line: Represents the 99th percentile, indicating the response time below which 99% of the samples fall.

Min: The minimum response time observed during the test.

Max: The maximum response time observed during the test.

Error%: The percentage of samples that resulted in an error.

Throughput refers to the rate at which the server processes requests, measured in the number of requests per unit of time, typically expressed as requests per second.

Received KB/sec refers to the rate at which data is being received from the server, measured in kilobytes per second.

Sent KB/sec refers to the rate at which data is transmitted to the server, measured in kilobytes per second.

Response Time Metrics: Analyse the average, median, and percentiles to gain insight into the distribution of response times. A significantly elevated average or median could suggest the presence of performance deficiencies.

C. Experimental Analysis

Within the vast realm of the internet, web applications present a wide array of experiences, with each navigation route showcasing distinct aspects of functionality and design. This experiment examines the web application hosted at "testphp.vulnweb.com" (See Figure 2).

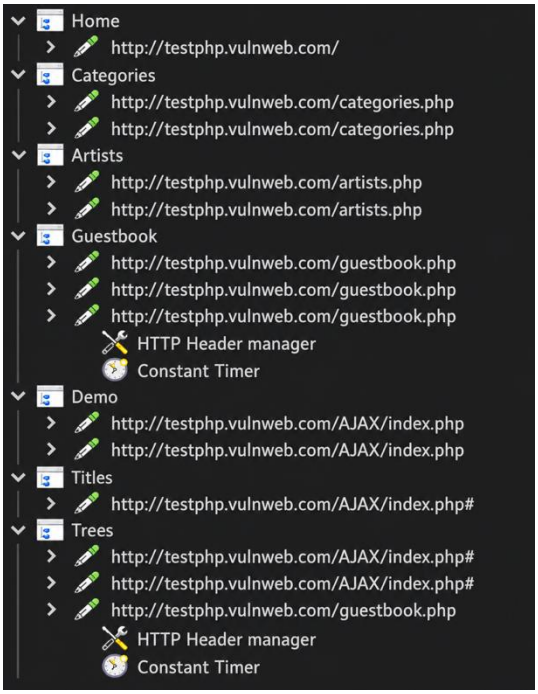

Fig .2 list of URLs visited over the Transmission Session

The provided list of URLs serves as a guide that is utilised in the experiment that is being conducted. As part of this experiment, these URLs were selected as labels and then visited in order to collect data on request-response interactions. The navigation path that is available within

https://doi.org/10.31436/ijpcc.v10i1.458

"testphp.vulnweb.com" is a reflection of the user's exploration of various areas of the web application. During this voyage, the complicated and varied nature of web applications is brought to light. Each URL serves as a gateway to a different universe of material and functionality, and this journey illustrates such a nature. It begins with a secure introduction on the homepage, and then moves on to provide dynamic experiences in the "Demo" area. The request-response cycle, on the other hand, is the most important part of this experiment since it enables the measurement of transmission sessions.

Performance testing is essential for guaranteeing the dependability and effectiveness of web applications and the transmission of data between requests and responses. This section presents the outcomes of the experimental analysis conducted on "1 thread (user)" and "5 thread (user)". The JMeter Aggregate Report yields data on the performance of different components inside a web application. The result of the test scenario on 1 thread (user) executing a sequence of HTTP queries against a susceptible web application is presented in Figure 3 and the details parameter in Appendix 1.

.

IV. PRESENTATION OF EXPERIMENTAL RESULT AND DISCUSSION

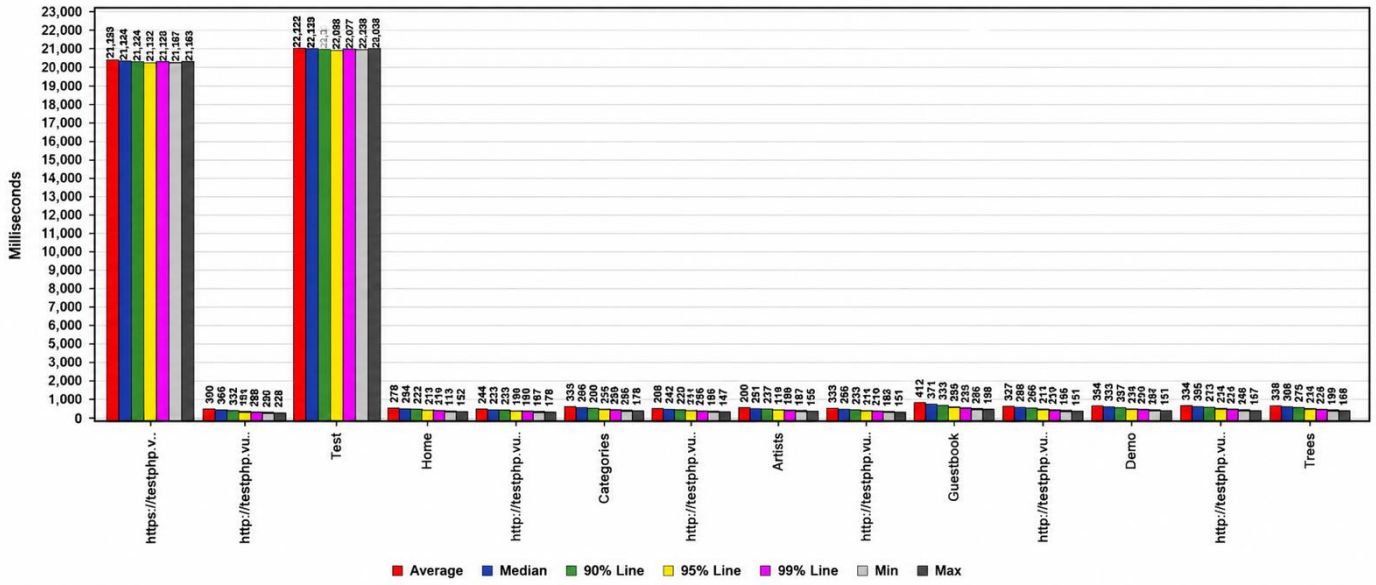

Fig. 3 Aggregate results on 1thread (user)

The initial label, "https://testphp.vulnweb.com/," stands out because of its unusual outcomes. Upon analysing a single sample, it is evident that the response times display a concerning level of consistency, with the average, median, and various percentiles (90%, 95%, 99%) all measuring 21459 milliseconds (see Figure 4). This gives rise to concerns over the performance of the application, indicating a possible bottleneck or problem with the server. The lack of errors is remarkable, however, the unreasonable response time of 21459 milliseconds and the poor throughput of 0.0466 requests per second indicate a significant performance issue.

In contrast, the website "http://testphp.vulnweb.com/" provides more logical and sensible outcomes. After conducting two samples, the reaction times exhibit a moderate average and median, and no errors have been recorded. The throughput of 0.03218 requests per second is deemed satisfactory, indicating that this particular aspect of the programme is functioning well given the current workload.

The "Test" label exhibits another occurrence of exceptionally rapid response times, measuring 22030 milliseconds, without any errors. The throughput is moderate, suggesting that the component has the capacity to handle a certain degree of workload but still needs optimisation in order to attain satisfactory response times.

The "Home" designation indicates a low response time of 438 milliseconds, without any errors, and an acceptable throughput of 0.01624 requests per second. These findings indicate that the home page of the programme is efficiently optimised and capable of responding well to the current level of usage. When navigating to particular pages inside the application, the "http://testphp.vulnweb.com/categories.php" URL shows satisfactory response times without any problems and a high throughput rate of 0.33875 requests per second. This suggests that the categories page is effectively managing the workload with satisfactory response times. Comparable trends may be identified in the "http://testphp.vulnweb.com/artists.php" and "http://testp

https://doi.org/10.31436/ijpcc.v10i1.458

hp.vulnweb.com/AJAX/index.php" URLs, where reasonable response times, absence of errors, and high throughputs suggest acceptable performance. However, the "http://testphp.vulnweb.com/guestbook.php" label offers a respectable response time with no problems but a considerably lower throughput (0.01876 requests per second). Further analysis and optimisation in this area could enhance the overall performance.

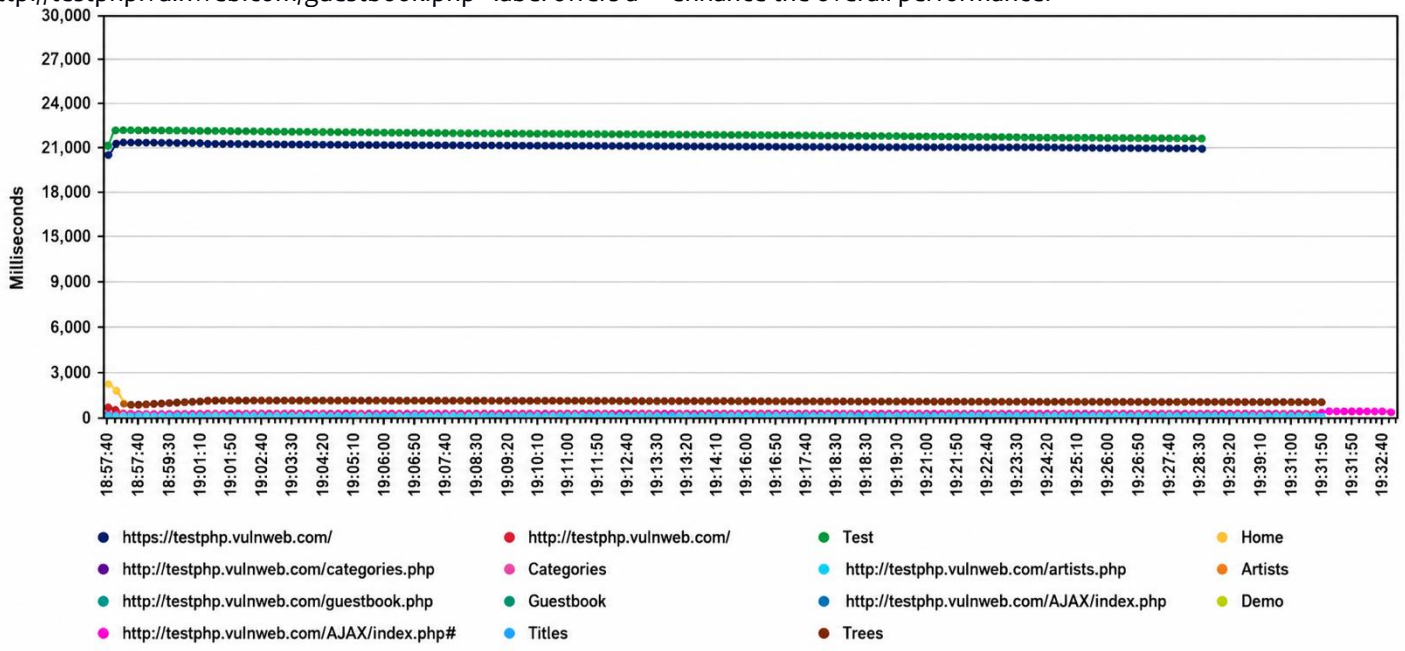

Fig. 4 The Response time on 1thread (user)

The outcome of the test scenario involving 5 threads (users) doing a series of HTTP queries on a vulnerable web application is provided in Figure 5 and the details is presented in Appendix 2.

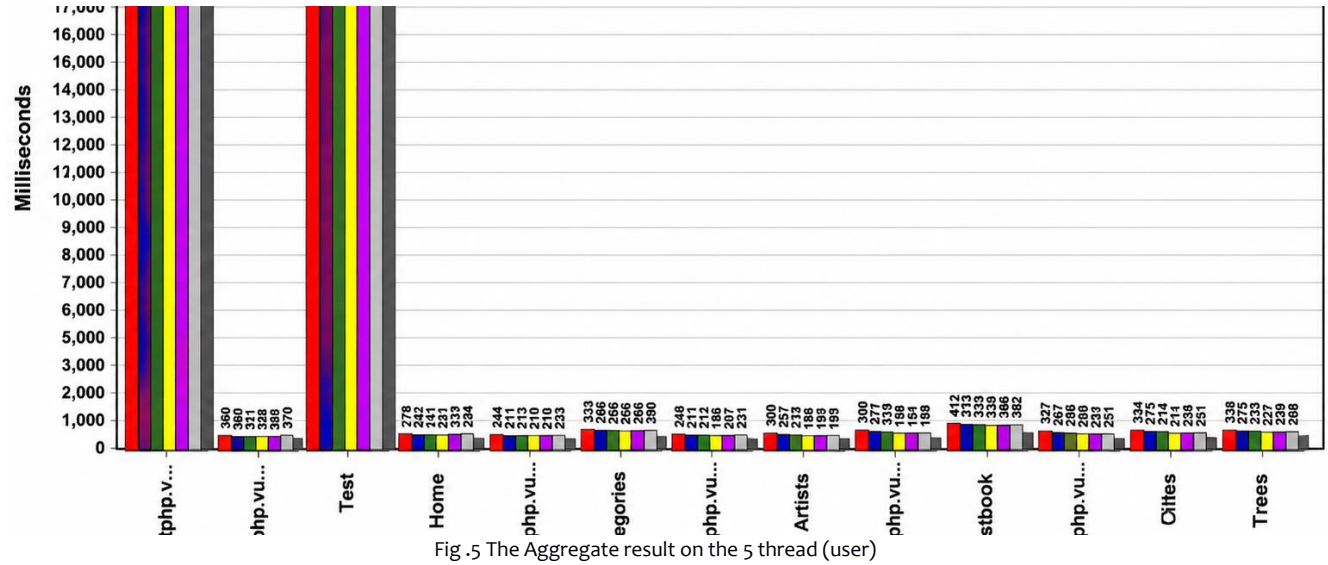

Fig .5 The Aggregate result on the 5 thread (user)

The initial element, designated as "https://testphp.vulnweb.com/," produces worrisome outcomes. Based on a single sample, the response times for all measures (average, median, 90%, 95%, 99%) are exceptionally high, measuring 21459 milliseconds (see Figure 6). The consistent lack of diversity in this context indicates a significant problem with performance, potentially caused by a bottleneck or inefficiency inside the application. A 100% error rate highlights the application's inability to react within an

https://doi.org/10.31436/ijpcc.v10i1.458

acceptable timeframe. The throughput is significantly low, measuring at 0.0466 requests per second, suggesting the presence of scalability challenges or limitations in available resources. In contrast, the component labelled "http://testphp.vulnweb.com/" exhibits more encouraging outcomes. The average and median response times, obtained from two samples, are 504 and 438 milliseconds, respectively, indicating a moderate level. No faults were detected, and the throughput is satisfactory, with a rate of 0.03218 requests per second. This implies that this particular element of the application is more efficiently optimised to manage the specified workload.

Another component, identified as "Test," demonstrates comparable issues to the initial label, displaying excessively long reaction times of 22030 milliseconds and a 100% error rate. The throughput is moderate, with a rate of 0.03877 requests per second, suggesting the presence of possible scalability difficulties. On the other hand, the "Home" component demonstrates positive outcomes with a swift response time of 438 milliseconds, absence of reported faults, and a satisfactory throughput of 0.01624 requests per second. This suggests a highly optimised homepage that is capable of efficiently managing the given workload.

Regarding individual pages inside the application, the label "http://testphp.vulnweb.com/categories.php" exhibits satisfactory response times with a significant amount of data processed per unit of time (0.33875 requests per second) and no documented problems. These findings indicate that the categories page effectively manages the workload while maintaining satisfactory response times. The "http://testphp.vulnweb.com/artists.php" and "http://testphp.vulnweb.com/AJAX/index.php" labels exhibit comparable patterns, characterised by reasonable response times, absence of errors, and high throughputs, indicating excellent performance of these components.

Nevertheless, the label "http://testphp.vulnweb.com/guestbook.php" demonstrates fluctuations in response times among the four examples. Although no mistakes were detected, the throughput is quite moderate, with a rate of 0.01876 requests per second, indicating the presence of potential optimisation possibilities.

The "TOTAL" row presents a comprehensive summary of the entire test. The average response time is moderately high, measuring at 2198 milliseconds. However, the presence of an 8.33% error rate indicates that a portion of the requests experienced problems during the test. Additional examination of the specific pages that are causing errors and performance problems is necessary. The overall throughput is satisfactory, with a rate of 0.05716 requests per second.

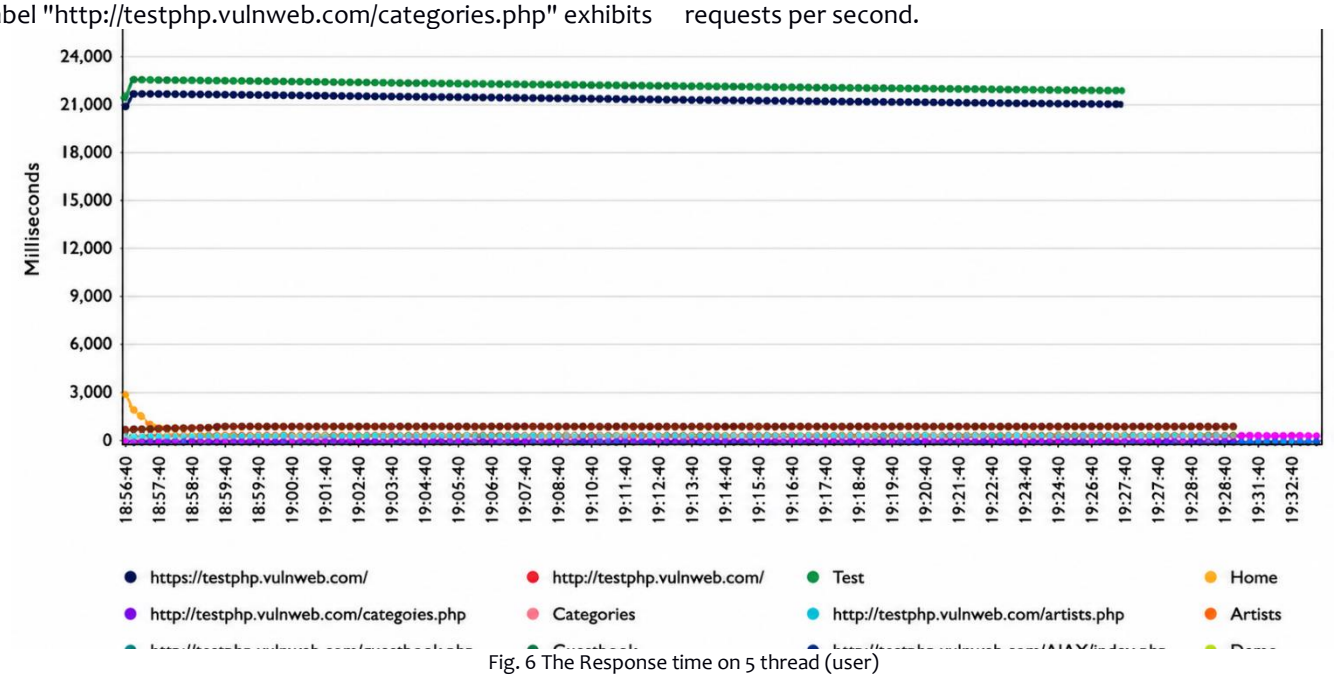

Fig. 6 The Response time on 5 thread (user)

To conclude, a web application's performance characteristics are better understood when the user scenario is multi-threaded. These comparison assessments help make informed application architectural, scalability, and performance decisions. JMeter Aggregate Report shows that performance testing is vital for designing dependable and efficient online apps that meet user expectations in diverse conditions.

V. CONCLUSION

This study examines the efficiency of data flow streams in TCP request-response cycle scenarios. The crucial finding pertains to the users' threads during the transmission

session. Adopting a five-threaded user scenario resulted in a more intricate simulation of simultaneous user activities. Although the specific outcomes for each labelled component may differ, it is possible to define some broad expectations and probable findings. It was expected that the average response times and percentiles would increase because of the increased number of users. The throughput is expected to indicate the application's ability to handle several requests at the same time. Components that faced difficulties in the 1 thread user scenario, such as extended response times or failures, may encounter exacerbated problems when subjected to the higher pressure of 5 thread users. In contrast, well-optimized components were anticipated to demonstrate proportional improvements in throughput, indicating their scalability The comparison of the two scenarios acts as a great instrument for optimising performance and evaluating scalability. It is essential to address bottlenecks and performance concerns that have been detected when using only one thread. This is necessary in order to improve the overall efficiency of the application. Furthermore, it is advisable to expand the testing scenarios in order to assess the scalability in the face of different user loads. This entails doing a comprehensive examination of the application's response to heightened concurrent users, determining performance thresholds, and guaranteeing a uniform and satisfactory user experience.

Acknowledgment

This research is made possible and supported by UMP-IIUM Sustainable Research Collaboration 2022 Research Grant (IUMP-SRCG22-014-0014).

Conflict of Interest

The authors declare that there is no conflict of interest.

References

[1] N. Muraleedharan, A. Thomas, S. Indu, B.S. Bindhumadhava. A Traffic Monitoring and Policy Enforcement Framework for HTTP. In2020 Third ISEA Conference on Security and Privacy (ISEA-ISAP) 2020 Feb 27 (pp. 81-86). IEEE.

[2] X. Zhou, Z. Zong, Rui T, Zhu J, inventors; ZTE Corp, assignee. Method for selecting policy and charging rules function. United States patent US 8,326,263. 2012 Dec 4.

[3] J. Liang, J.W. Sun. Active Detection of Application Layer Attacks Based on Analysis of HTTP-Session. Advanced Materials Research. 2011 Aug 2;268:1253-8.

[4] N.V Patil, C. Rama Krishna, . Kuma. Distributed frameworks for detecting distributed denial of service attacks: a comprehensive review, challenges and future directions. Concurrency and Computation: Practice and Experience. 2021 May 25;33(10): e6197.

[5] M. Najafimehr, S. Zarifzadeh, S. Mostafavi. DDoS attacks and machine - learning - based detection methods: A survey and taxonomy. Engineering Reports. 2023:e12697.

[6] A.S. Ahmadzai Q. Yazdani, A. Abubakar. Analysis of Business Intelligence Systems Transmission Session. Journal of Science and Technology. 2022 Dec 6;27(1):16-28.

[7] S. Wedman A. Tetmeyer H. Saiedian. An analytical study of web application session management mechanisms and HTTP session hijacking attacks. Information Security Journal: A Global Perspective. 2013 Mar 4;22(2):55-67.

[8] B.C. Nathani, E. Adi. Website vulnerability to session fixation attacks. J. Information Eng. App. II. 2012;7:32-6.

[9] E. Hoxha., I. Tafa., K. Ndoni., I. Tahiraj., A. Muco. Session hijacking vulnerabilities and prevention algorithms in the use of internet. (2022). doi: 10.18844/gjcs.v12i1.7449.

[10] D. Yadav, D. Gupta, D. Singh, D. Kumar, U. Sharma. Vulnerabilities and security of web applications. In2018 4th International Conference on Computing Communication and Automation (ICCCA) 2018 Dec 14 (pp. 1-5).

[11] P. Namitha P. Keerthijith. A Survey on Session Management Vulnerabilities in Web Application. In2018 International Conference on Control, Power, Communication and Computing Technologies (ICCPCCT) 2018 Mar 23 (pp. 528-532). IEEE.

[12] F. Shafique, S. Fatima, F.Y. Khuhawar, Z.A Arain. An Analysis of Multipath TCP Security Vulnerabilities: A Research Study. In2022 IEEE 19th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET) 2022 Dec 19 (pp. 172-177)..

[13] M. Chen, F. Dai, B. Yan, J. Cheng. Encryption Algorithm for TCP Session Hijacking. InArtificial Intelligence and Security: 6th International Conference, ICAIS 2020, Hohhot, China, July 17–20, 2020, Proceedings, Part II 6 2020 (pp. 191-202). Springer International Publishing.

[14] Y. Cao, J. Nejati, A. Balasubramanian, A. Gandhi. Econ: Modeling the network to improve application performance. InProceedings of the internet measurement conference 2019 Oct 21 (pp. 365-378).

[15] A. Raje, A. Agrawal, T. Santhosh T, Sachan S, Nayak M, Sinha S, De I, Haq J, Maitra S. The Internet Measurement Network (AIORI-IMN). In2023 4th International Conference on Computing and Communication Systems (I3CS) 2023 Mar 16 (pp. 1-8). IEEE.

[16] M.A. Hoque, A. Rao, S. Tarkoma. Network and application performance measurement challenges on android devices. ACM SIGMETRICS Performance Evaluation Review. 2021 Mar 5;48(3):6-11.

[17] E.M. Dastin-van Rijn, N.R. Provenza, M.T. Harrison D.A. Borton. How do packet losses affect measures of averaged neural signals*f*. In2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC) 2021 Nov 1 (pp. 941-944).

https://doi.org/10.31436/ijpcc.v10i1.458

Appendix 1: The Result of the Test Scenario Involving 1 threads (users)

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Max	Error %	Throughput	Received KB/sec	Sent KB/sec
https://testphp.vulnweb.com/	1	21459	21459	21459	21459	21459	21459	21459	100.00%	0.0466	0.13	0
http://testphp.vulnweb.com/	2	504	438	571	571	571	438	571	0.00%	0.03218	0.16	0.01
Test	1	22030	22030	22030	22030	22030	22030	22030	100.00%	0.03877	0.31	0.01
Home	1	438	438	438	438	438	438	438	0.00%	0.01624	0.08	0.01
http://testphp.vulnweb.com/categories.php	2	241	237	246	246	246	237	246	0.00%	0.33875	2.1	0.13
Categories	1	483	483	483	483	483	483	483	0.00%	0.03347	0.42	0.03
http://testphp.vulnweb.com/artists.php	2	344	244	444	444	444	244	444	0.00%	0.46501	2.53	0.17
Artists	1	688	688	688	688	688	688	688	0.00%	0.03164	0.34	0.02
http://testphp.vulnweb.com/guestbook.php	4	290	240	436	436	436	236	436	0.00%	0.01876	0.1	0.01
Guestbook	1	921	921	921	921	921	921	921	0.00%	0.01239	0.21	0.02
http://testphp.vulnweb.com/AJAX/index.php	2	231	221	242	242	242	221	242	0.00%	0.41246	1.8	0.15
Demo	1	463	463	463	463	463	463	463	0.00%	0.05523	0.48	0.04
http://testphp.vulnweb.com/AJAX/index.php#	3	374	439	442	442	442	243	442	0.00%	0.02134	0.09	0.01
Titles	1	439	439	439	439	439	439	439	0.00%	0.04234	0.19	0.02
Trees	1	925	925	925	925	925	925	925	0.00%	0.00673	0.1	0.01
TOTAL	24	2198	438	925	21459	22030	221	22030	8.33%	0.05716	0.38	0.03

https://doi.org/10.31436/ijpcc.v10i1.458

Appendix 2: The Result of the Test Scenario Involving 1 threads (users).

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Max	Error %	Throughput	Received KB/sec	Sent KB/sec
https://testphp.vulnweb.com/	1	21459	21459	21459	21459	21459	21459	21459	100.00%	0.0466	0.13	0
http://testphp.vulnweb.com/	2	504	438	571	571	571	438	571	0.00%	0.03218	0.16	0.01
Test	1	22030	22030	22030	22030	22030	22030	22030	100.00%	0.03877	0.31	0.01
Home	1	438	438	438	438	438	438	438	0.00%	0.01624	0.08	0.01
http://testphp.vulnweb.com/categories.php	2	241	237	246	246	246	237	246	0.00%	0.33875	2.1	0.13
Categories	1	483	483	483	483	483	483	483	0.00%	0.03347	0.42	0.03
http://testphp.vulnweb.com/artists.php	2	344	244	444	444	444	244	444	0.00%	0.46501	2.53	0.17
Artists	1	688	688	688	688	688	688	688	0.00%	0.03164	0.34	0.02
http://testphp.vulnweb.com/guestbook.php	4	290	240	436	436	436	236	436	0.00%	0.01876	0.1	0.01
Guestbook	1	921	921	921	921	921	921	921	0.00%	0.01239	0.21	0.02
http://testphp.vulnweb.com/AJAX/index.php	2	231	221	242	242	242	221	242	0.00%	0.41246	1.8	0.15
Demo	1	463	463	463	463	463	463	463	0.00%	0.05523	0.48	0.04
http://testphp.vulnweb.com/AJAX/index.php#	3	374	439	442	442	442	243	442	0.00%	0.02134	0.09	0.01
Titles	1	439	439	439	439	439	439	439	0.00%	0.04234	0.19	0.02
Trees	1	925	925	925	925	925	925	925	0.00%	0.00673	0.1	0.01
Thread Group:https://testphp.vulnweb.com/	5	21030	21029	21036	21047	21047	21013	21047	100.00%	0.22959	0.64	0
Thread Group:http://testphp.vulnweb.com/	10	438	439	447	447	449	428	449	0.00%	0.15933	0.81	0.06
Thread Group:Test	5	21468	21469	21475	21475	21475	21454	21475	100.00%	0.19786	1.56	0.07
Thread Group:Home	5	440	440	441	449	449	435	449	0.00%	0.08024	0.41	0.03
Thread Group:http://testphp.vulnweb.com/categories.php	10	228	224	242	242	248	222	248	0.00%	1.51538	9.41	0.57
Thread Group:Categories	5	457	450	456	490	490	445	490	0.00%	0.16356	2.03	0.12
Thread Group:http://testphp.vulnweb.com/artists.php	10	352	231	491	491	505	221	505	0.00%	1.99084	10.84	0.74
Thread Group:Artists	5	704	700	712	730	730	683	730	0.00%	0.15467	1.68	0.11
Thread Group:http://testphp.vulnweb.com/guestbook.php	20	282	234	436	441	443	222	443	0.00%	0.09346	0.52	0.04
Thread Group:Guestbook	5	903	900	910	917	917	891	917	0.00%	0.0613	1.01	0.08

https://doi.org/10.31436/ijpcc.v10i1.458

Thread Group:http://testphp.vulnweb.com/AJAX/index.php	10	229	223	240	240	246	218	246	0.00%	1.76647	7.73	0.66
Thread Group:Demo	5	458	454	462	469	469	452	469	0.00%	0.26424	2.31	0.2
Thread Group:http://testphp.vulnweb.com/AJAX/index.php#	15	367	437	442	444	447	222	447	0.00%	0.10612	0.46	0.04
Thread Group:Titles	5	436	437	439	442	442	431	442	0.00%	0.20477	0.9	0.08
Thread Group:Trees	5	895	897	898	905	905	885	905	0.00%	0.03345	0.48	0.04
TOTAL	144	2155	437	917	21047	21475	218	22030	8.33%	0.0651	0.44	0.03