iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** XII **Month of publication:** December 2023

DOI: https://doi.org/10.22214/ijraset.2023.57785

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue XII Dec 2023- Available at www.ijraset.com

CodeTogether: Bridging Autonomy and Collaboration in Online Judging through Virtual Rooms

Shahedhadeennisa S[1], Prasham Vipul Prabhakar[2], Priyanshu Gupta[3], Nishant Mohan[4], Sanjida Yasmin[5]
Dayananda Sagar College of Engineering, Shavige Malleshwara Hills, 91st Main Rd, 1st Stage, Kumaraswamy Layout, Bengaluru, Karnataka-560078

***Abstract:** This research presents an innovative online judge system, merging features from platforms like LeetCode with a unique twist. Users can create and share problem statements, creating an immersive coding experience. The platform introduces virtual rooms, facilitating both solo coding and collaborative, pseudo-contest environments with real-time chat and leaderboards. Built for an engaging coding experience, the system aims to redefine how programmers approach challenges. Emphasizing a balance between individual skill development and competitive coding experiences, this paper explores the design's impact on skill enhancement. The model outlined serves as a valuable resource for developers looking to create dynamic coding environments, encouraging a blend of solo and collaborative coding practices.*

I. INTRODUCTION

There exists a plethora of online platforms to facilitate coding practice for interviews and programming contests alike. Courses are taught both online as well as in traditional university settings for solving problems based on datastructures and algorithms. However, both of these methods have their shortcomings.

The rise of online coding platforms not only serve as a space for people to refine their coding abilities, but also foster a community-driven approach to problem solving. Recognizing the demand for competitive and collaborative coding experiences, this research introduces an innovative online judge system aiming to offer a mix of individual and competitive coding environments.

The evolution of online coding platforms such as LeetCode and HackerRank has influenced how individuals tackle algorithmic problems. While these platforms provide a vast number of problems to practice in isolation, there's a noticeable gap in collaborating coding experiences. This research addresses that gap by introducing an online judge system which allows users to create and share problem statements, and participate in virtual rooms that support both solo coding and pseudo-contests. In these rooms, participants can code alongside their peers and engage in real-time discussions through the integrated chat feature. Such a platform not only serves as a tool for self-improvement, but also fosters a competitive environment at the user's convenience.

II. RELATED WORK

The concept of online coding platforms has received considerable attention in the literature. Platforms such as LeetCode, HackerEarth, and HackerRank have laid the foundation for individual skill enhancement through carefully curated problem sets. However, the collaborative aspect, especially within a pseudo-contest environment, has seen less exploration.

Recent research by Smith et al. [4] sheds light on the benefits of collaborative coding, emphasizing its positive impact on skill diversity and problem-solving approaches. Our work builds upon these foundations, introducing a unique model that combines individual and collaborative coding experiences, contributing to the broader discourse on effective programming education.

III. LITERATURE SURVEY

Yutaka Watanobe et al., present the functional and non-functional requirements of Online Judge System (OJS), a tool used to evaluate program correctness. OJSs have been developed and operated for over a decade, evaluating over six million solutions. The core element of an OJS is the judge system, which performs load balancing, evaluation, and notification processes. However, the construction theory for OJS development has not been sufficiently discussed. The paper aims to address this gap by presenting real-world experiences and challenges encountered during the OJS's long journey.

©IJRASET: All Rights are Reserved

Hua Zhang et al., developed an online judging-based experimental teaching platform to address the complexity of code validation in programming courses.

The platform functions as an online arrangement experiment for teachers, online coding, and automatic evaluation for students. This approach not only improves students' problem-solving abilities but also assists teachers in reviewing experiments quickly and correctly, reducing teacher workload and improving the quality of experimental teaching. Research has shown that the application of online judge technology in experimental teaching can significantly improve students' problem-solving abilities and reduce teacher workload.

MU¨ MI˙NE KAYA et al., integrate an online compiler and a source code plagiarism detection tool into the Moodle distance education system for easy assessment of programming assignments. The Moodle system is extended with the GCC compiler and the Moss source code plagiarism detection tool. The study found that using the online compiler and the plagiarism detection tool reduces time and effort needed for assessment, prevents students from plagiarism, and increases their success in programming-based Data Structures courses.

The integration of these tools into the Moodle distance education system is a significant step towards improving distance education.

Michal Fori˘sek discusses the suitability of programming tasks for automated evaluation, focusing on contests like the International Olympiad in Informatics (IOI) and the ACM International Collegiate Programming Contest (ACM ICPC). The main goals of the automated evaluation procedure are to find and implement correct algorithms, determine the number of points for correct algorithms based on their asymptotic time complexity, and deduct points for small mistakes. The article also discusses ways to recognize such tasks and discusses other possibilities for evaluation, which can be applied to any programming task, including school assignments.

Panagiotis Fotaris et al., explore the application of gamification techniques to computer programming classes. Video games are successful in sustaining high levels of motivation and engagement, allowing students to solve complex problems without feeling fatigue or frustration. The researchers applied a gamified, multi-dimensional, problem-based learning approach to an entry-level Python programming course, using the "Kahoot!" Classroom Response System (CRS), the classroom version of TV game show "Who Wants To Be A Millionaire?", and Codecademy's interactive platform. The results showed that the gamified approach was motivating and enriching for both students and instructors, proving the potential of technology-enhanced learning in higher education.

Cheedoong Drung et al., introduce the affinity algorithm to improve the precision of user programs' processing time and use queuing theory to analyze and improve the system's performance indicators. By implementing these mechanisms, the average capacity of simultaneously judging tasks will increase, judging results will become more precise, and the total number of computers in an OJ system will decrease, yielding lower hardware costs. The authors also discuss the challenges of measuring the process time of a given task in modern computers, as the OS' process/thread allocation and scheduling are transparent to users and programs.

IV. SYSTEM ARCHITECTURE

The system consists of several database schemas, namely, user, contests, submissions, problem, and judge. The user schema stores typical user data such as an id, name, email password, etc. as well as a method getSession to get an instance of the current user session. The contests schema records room data such as roomID, participants, problems, duration, etc. and provides methods getRank as well as getProblems. The submissions contain records of submission id, language used, id of submitter, code, input, expected output and contestID. The problem schema consists of problemID, title, description, timeLimit, input, expected output, etc. along with methods such as addTestCase, removeTestCase and updateDescription.

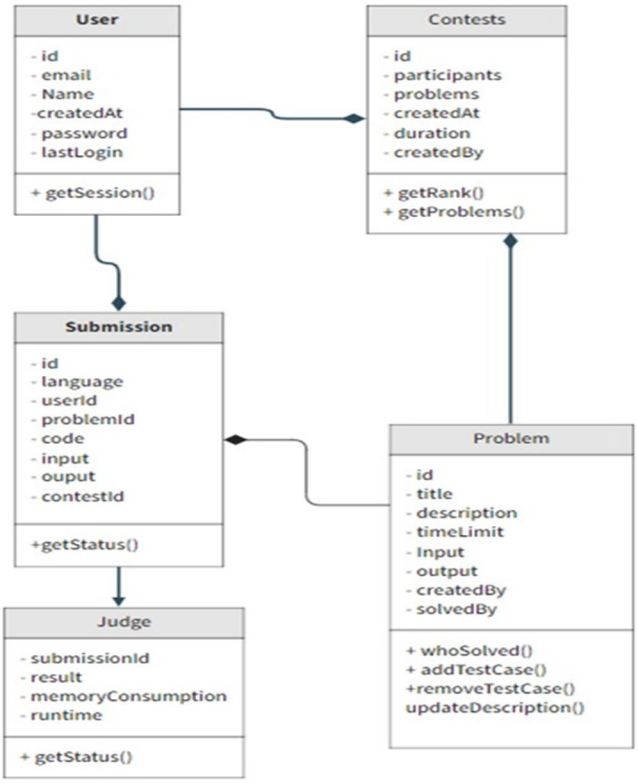
V. IMPLEMENTATION DETAILS

he comprehensive system comprises four integral components, each encapsulated within a docker container to ensure optimal modularity and seamless deployment. The frontend, crafted with React and Redux, serves as the user-facing interface, offering an intuitive and interactive space for user engagement. Leveraging the power of React for dynamic user interfaces and Redux for state management, this component establishes a robust endpoint, facilitating a responsive and feature-rich user experience. On the backend, the system employs Node.js to handle a myriad of tasks, including processing user requests, managing authentication protocols, and enabling real-time communication within virtual rooms. Node.js, known for its non-blocking, event-driven architecture, ensures efficient handling of concurrent connections, enhancing the system's scalability and responsiveness. This backend architecture forms the backbone of the entire system, seamlessly orchestrating user interactions and data flow. For data persistence, MongoDB serves as the database, storing a diverse range of information crucial for system functionality.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

User data, problem statements, submission history, and other relevant details find a structured home within MongoDB's document-oriented database.

The choice of MongoDB aligns with its flexibility, scalability, and ability to handle large volumes of unstructured data efficiently, contributing to the system's overall robustness.

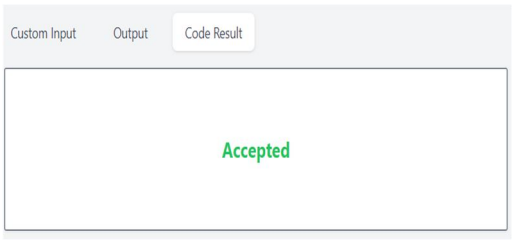


A unique and crucial aspect of the system lies in its containerized code execution environment. Each time a user submits code, a dynamic container is instantiated, creating a dedicated and isolated space for code execution. This dynamic container creation is meticulously orchestrated to optimize resource utilization within an isolated environment. This not only ensures efficient management of computing resources but also fortifies the system against potential security threats posed by malicious code submissions. In essence, the technology stack, programming languages, and containerization strategy employed in the system underscore a thoughtful and strategic approach to software architecture. By utilizing React, Redux, Node.js, MongoDB, and Docker containers, the system achieves a harmonious blend of user-centric frontend design, robust backend functionality, efficient data management, and secure code execution practices.

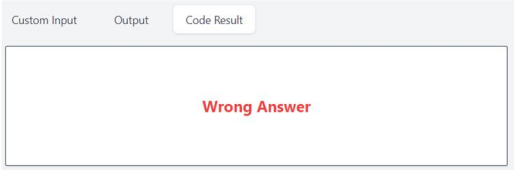
VI. FEATURES AND FUNCTIONALITIES

Much like a typical online judge, a user can choose a problem they would like to solve and submit their code. On submission, one of five verdicts – Accepted, Wrong Answer, Time Limited Exceeded, Compilation Error or Runtime Error are returned.

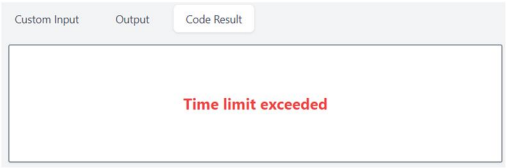
1) Accepted Verdict:



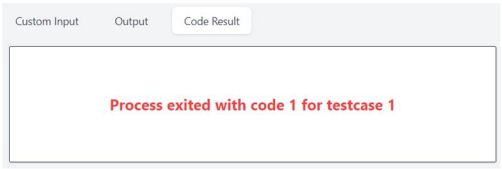
2) *Wrong Answer Verdict:*



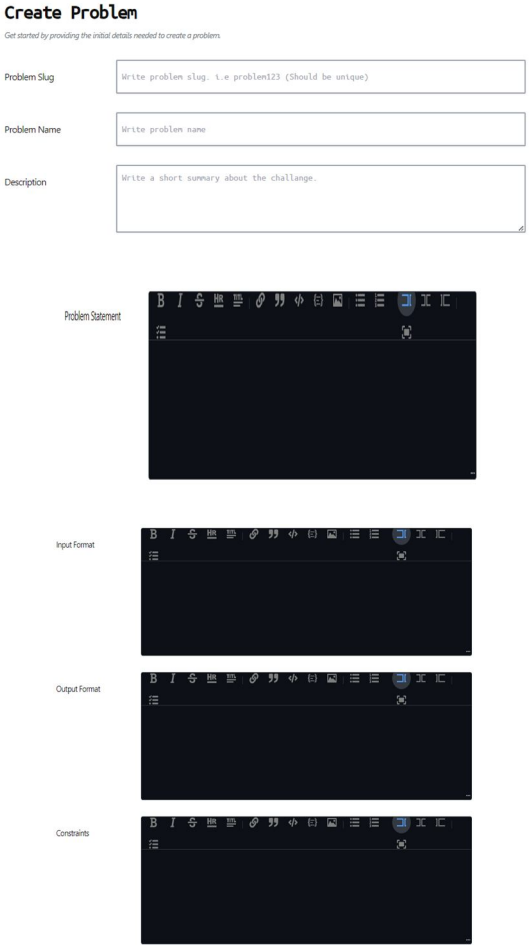
3) *Time Limit Exceeded Verdict*:

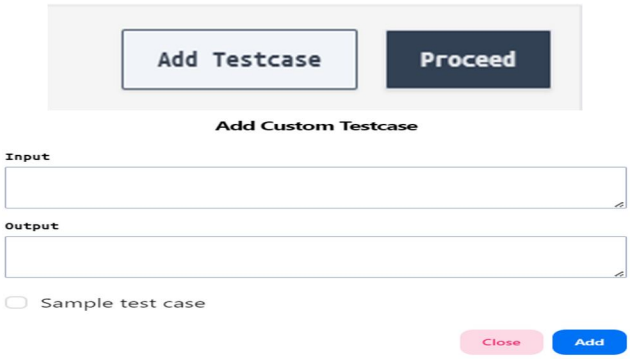


4) *Compilation or Runtime Error:*



Users can also add their own problems by providing the problem statement, constraints, input and output format along with testcases and expected outputs.





Furthermore, any user can create a virtual room and invite people with an invite link to join their pseudo-contest. The virtual room consists of a problem statement on the left, a code editor in the middle and a combined chat and scoreboard on the right

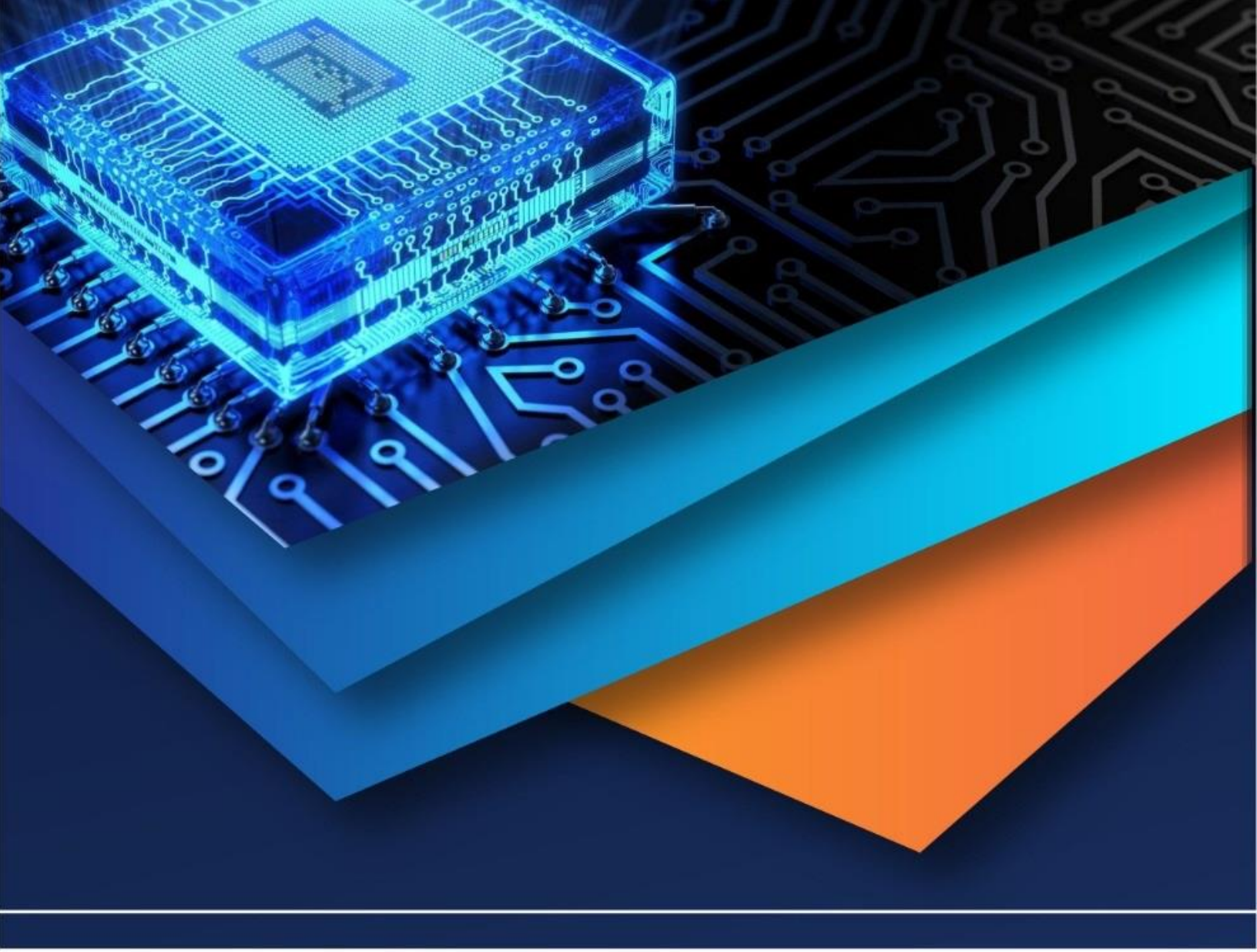
VII. CONCLUSION

In conclusion, our online judge system redefines the coding practice landscape by seamlessly integrating solo and collaborative coding environments. The research contributes to the ongoing evolution of programming education by introducing a model that fosters individual skill development alongside collaborative coding experiences.

REFERENCES

[1] D. Peeren, "RIPS Technologies Blog," 22 December 2016. [Online].
[2] M. M. Group, "Internet world Stats," 31 December 2017. [Online].
[3] "W3Techs web Technology Surveys," 5 April 2018. [Online].
[4] T. Farah, D. Alam, M. A. Kabir and T. Bhuiyan, "SQLi penetration testing of financial Web applications: Investigation of Bangladesh region," 2015 World Congress on Internet Security (WorldCIS), Dublin, 2015, pp. 146-151.
[5] Z. Su and G. Wassermann, "The essence of command injection attacks in web applications," ACM SIGPLAN Notices, vol. 41, no. 1, pp. 372-382. ACM, 2006.
[6] C. F. James, O. Vitaly, B. Nish, and H. Niels, "Buffer Overflow Attacks: Detect, Exploit, Prevent," (2005): 1-932266.
[7] A. Shrivastava, S. Choudhary and A. Kumar, "XSS vulnerability assessment and prevention in web application," 2016 2nd International Conference on Next Generation Computing Technologies (NGCT), Dehradun, 2016, pp. 850-853.
[8] D. Huluka and O. Popov, "Root cause analysis of session management and broken authentication vulnerabilities," World Congress on Internet Security (WorldCIS-2012), Guelph, ON, 2012, pp. 82-86.
[9] Y. Takamatsu, Y. Kosuga and K. Kono, "Automated detection of session management vulnerabilities in web applications," 2012 Tenth Annual International Conference on Privacy, Security and Trust, Paris, 2012, pp. 112-119.
[10] M. A. Obaida, E Nelson, J. E. Rene V, I. Jahan, and S. Z. Sajal. "Interactive Sensitive Data Exposure Detection Through Static Analysis.", 2017.
[11] Q. H. Mahmoud, D. Kauling and S. Zanin, "Hidden android permissions: Remote code execution and shell access using a live wallpaper," 2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, 2017, pp. 599-600.
[12] S. Mohammad and S. Pourdavar, "Penetration test: A case study on remote command execution security hole," 2010 Fifth International Conference on Digital Information Management (ICDIM), Thunder Bay, ON, 2010, pp. 412-416.
[13] L. Zhang, H. Zhang, X. Zhang and L. Chen, "A New Mechanism for Trusted Code Remote Execution," 2007 International Conference on Computational Intelligence and Security Workshops (CISW 2007), Heilongjiang, 2007, pp. 574-578.
[14] R.H. and B. Joyce., "Purify: Fast detection of memory leaks and access errors," Winter USENIX Conferenc, San Francisco, California, USA, January: 1992, p. 125--136.
[15] N. Dor, "Cleanness checking of string manipulations in C programs via integer analysis," Static Analysis, 2001.
[16] N. Frykholm, "Countermeasures against buffer overflow attacks," RSA Tech Note, 2000, pp. 1-9.
[17] S.D. Xenitellis, "Identifying security vulnerabilities through input flow tracing and analysis," Information Management & Computer Security, vol. 11, 2003, pp. 195-199.
[18] U. Erlingsson, Low-level Software Security : Attacks and Defenses Low-level Software Security : Attacks and Defenses, Redmond, WA, USA: 2007.
[19] J. Wilander and M. Kamkar, "A comparison of publicly available tools for dynamic buffer overflow prevention," Proceedings of the 10th Network and Distributed System Security Symposium, Citeseer, 2003, p. 149–162.
[20] H. Shacham, M. Page, B. Pfaff, and E. Goh, "On the effectiveness of address-space randomization," ACM conference on, 2004, p. 298.
[21] P. Mell, K. Scarfone, and S. Romanosky, "A Complete Guide to the Common Vulnerability Scoring System Version 2.0," System, 2007, pp. 1-23.
[22] K. Geers, "Live Fire Exercise: Preparing for Cyber War," 2010.
[23] H. Holm, T. Sommestad, J. Almroth, and M. Persson, "A quantitative evaluation of vulnerability scanning," Information Management & Computer Security.
[24] P.H. Garthwaite, J.B. Kadane, and A. O'Hagan, "Statistical methods for eliciting probability distributions," Journal of the American Statistical Association, vol. 100, 2005, pp. 680-701.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)