RESEARCH ARTICLE OPEN ACCESS

Simulated Bug Bounty Platform for Ethical Hacking Training

Piramu Chendu S*, Sathya M**
*(Student, Computer Science and Engineering, Francis Xavier Engineering College, Tirunelveli
Email:piramuchendus.ug22.cs@francisxavier.ac.in)
** (Assistant Professor, Computer Science and Engineering, Francis Xavier Engineering College, Tirunelveli
Email: sathyam@francisxavier.ac.in)

----------**********----------

Abstract:

The need for accessible, hands-on training environments that go beyond standard theoretical learning is increasing due to the rapid evolution of cybersecurity threats. The Simulated Bug Bounty Platform for Ethical Hacking Training, a web-based system intended to offer a secure, organized, and interactive setting for vulnerability assessment and reporting, is presented in this paper. By enabling users to explore programs, engage with simulated vulnerable apps, and submit structured bug reports within specified scopes, the platform mimics real-world bug bounty operations. It incorporates essential modules including a Company Interface for program design, scope specification, and report validation, and a Hacker Interface for involvement. Practice labs containing popular vulnerabilities like SQL Injection and Cross-Site Scripting are incorporated into the system, allowing users to obtain practical experience without running the risk of legal or ethical issues. Vulnerability severity is also assessed using a CVSS-based grading system, and gamification elements like leaderboards and rankings boost user incentive and engagement. Efficiency and consistency are increased by automated procedures for performance monitoring, feedback, and report validation. The platform's potential as a scalable and legally compliant solution for ethical hacking training is validated by experimental results, which show that it successfully bridges the gap between theoretical knowledge and real cybersecurity skills.

Keywords — Web security, practice labs, vulnerability assessment, CVSS scoring, ethical hacking, cybersecurity training, gamification, and simulated bug bounty platforms

----------**********----------

I. INTRODUCTION

The complexity and frequency of cybersecurity threats have increased dramatically over the last ten years, with attackers using methods including web application exploits, injection attacks, and authentication bypass mechanisms to infiltrate systems. Conventional cybersecurity education approaches sometimes fall short of adequately training students to deal with real-world security issues since they mostly rely on theoretical learning and little hands-on experience. Because of the legal dangers, lack of advice, and intense competition, true bug bounty programs are not appropriate for novices, even though they offer opportunity for practical experience. Researchers have investigated simulation-based learning environments that offer safe and regulated platforms for cybersecurity training in order to overcome these constraints. These methods allow users to conduct ethical hacking and vulnerability evaluation without impacting actual systems or going against the law. The Simulated Bug Bounty Platform for Ethical Hacking Training, a web-based system created to mimic actual bug bounty procedures in a secure and organized setting, is presented in this paper. Users can engage with simulated vulnerable apps, examine programs, and submit structured bug reports based on predetermined scopes using this platform. It includes practice labs with popular vulnerabilities like SQL Injection and Cross-Site Scripting, allowing users to get practical experience in a safe

environment. The system also incorporates gamification elements like leaderboards and rankings to increase incentive and engagement, as well as a CVSS-based scoring methodology for vulnerability assessment. Through the integration of automatic feedback systems, organized workflows, and simulation, the platform offers a holistic learning ecosystem that connects theoretical understanding with real-world application. In addition to preparing users for involvement in real-world bug bounty programs, the suggested method shows that simulation-based bug bounty environments can function as efficient, scalable, and legally compliant cybersecurity training solutions.

II. OBJECTIVE

The main goal of the Simulated Bug Bounty Platform for Ethical Hacking Training is to provide a safe, interactive web-based system that allows users to practice ethical hacking and vulnerability assessment in a regulated, legally compliant setting. This platform seeks to improve practical cybersecurity abilities by emulating real-world bug bounty workflows, in contrast to typical learning techniques that mostly concentrate on theory.Creating a realistic and organized bug bounty environment where users may investigate programs, comprehend specified scopes, and conduct vulnerability testing on simulated apps is one of the main goals. The solution guarantees that users are given hands-on experience in discovering and analyzing common security vulnerabilities like SQL Injection, Cross-Site Scripting, and authentication problems by simulating real-world scenarios. Establishing a thorough bug reporting system that enables people to submit organized vulnerability reports is another crucial goal. The system makes sure that reports adhere to professional standards, which include impact analysis, reproduction procedures, and thorough descriptions. This enhances users' reporting abilities and helps them comprehend the significance of responsible disclosure.The project also focuses on creating a corporate interface that allows companies to define scope, payment tiers, and vulnerability categories in order to design and manage bug bounty programs. This module offers a realistic view of how bug bounty programs function on actual platforms while guaranteeing regulated participation.In order to assess vulnerability severity and replicate incentive distribution, the system also intends to incorporate a CVSS-based scoring algorithm. This guarantees that users acquire industry-standard techniques for vulnerability prioritization and risk assessment. Using gamification elements like points, rankings, badges, and leaderboards to increase user engagement is another goal. These tools encourage users to take an active role, develop their abilities, and monitor their advancement over time.The creation of simulated practice laboratories, where users may securely conduct security testing without interfering with actual systems, is another focus of the project. These labs guarantee ethical compliance by offering a risk-free setting for learning and experimentation.Additionally, the system seeks to offer real-time performance tracking and feedback, enabling users to track their development, pinpoint areas for growth, and consistently improve their abilities.Lastly, by bridging the gap between theoretical understanding and real-world application, the project aims to support cybersecurity education. The platform offers a scalable and efficient way to train future ethical hackers by integrating simulation, organized processes, and interactive elements.Along with these objectives, the platform's modular and scalable architecture enables future improvements including cloud-based deployment, automatic report validation, and AI-based vulnerability recommendations. This guarantees that the system will continue to offer a productive learning environment and be flexible enough to meet changing cybersecurity requirements.

III. METHODOLOGY

The suggested Simulated Bug Bounty Platform for Ethical Hacking Training's entire processing pipeline unifies performance evaluation, structured reporting, vulnerability testing, and user interaction into a single framework. Users register, investigate bug bounty

programs, test in simulated environments, and submit vulnerability reports as part of the system's sequential workflow. Following validation, these reports are rated using established methods and utilized to update user performance metrics, ranks, and awards. This end-to-end process converts user behaviors into valuable cybersecurity insights, facilitating learning and assessment in a safe and moral setting. Fig. 1 shows the system architecture of the suggested platform and depicts the entire workflow and interaction between system components.

A. **System Architecture:** The proposed platform's system design starts with the User Module, where users are classified as either companies or hackers. Through program browsing, access requests, practice lab participation, and bug reports, hackers engage with the system. By establishing scope, compensation levels, and vulnerability categories, companies, on the other hand, design and oversee bug bounty programs.The technology gives hackers access to Practice Labs, which mimic vulnerable real-world applications, once they have chosen a program. Within predetermined parameters, these laboratories enable users to securely test vulnerabilities like SQL Injection, Cross-Site Scripting, and authentication problems. Once a vulnerability has been discovered, the hacker uses the BugReport Module to submit a comprehensive report that includes all relevant details, including reproduction methods, expected output, actual result, and proof of concept.The filed report is evaluated for validity and is linked to both the chosen program and the hacker. Using a CVSS Calculator Module, which examines factors including attack vector, complexity, and impact, the system assesses the report and determines a severity rating. The report is classified as valid, invalid, or duplicate based on this assessment.The Payment Module simulates payment distribution based on severity levels if the report is valid. In order to ensure real-time tracking of user accomplishments, the Leaderboard Module simultaneously updates the hacker's performance metrics, such as points, rank, and reputation.Additionally, the system has a HackerRequest Module that handles hackers' requests to join programs, giving businesses the option to approve or deny participation.A Notification Module also guarantees real-time communication by updating users on program updates, incentives, and report status.A centralized backend oversees all system interactions and efficiently stores and processes user, report, program, payment, and notification data. Each component of the architecture functions independently while seamlessly integrating with other modules thanks to its modular design.In addition to facilitating future improvements like AI-based report validation and intelligent vulnerability suggestions, our modular and scalable design guarantees effective handling of numerous users and reports. The solution offers a comprehensive simulation of real-world bug bounty platforms by fusing organized workflows, automated evaluation, and interactive elements, improving learning and useful cybersecurity abilities.

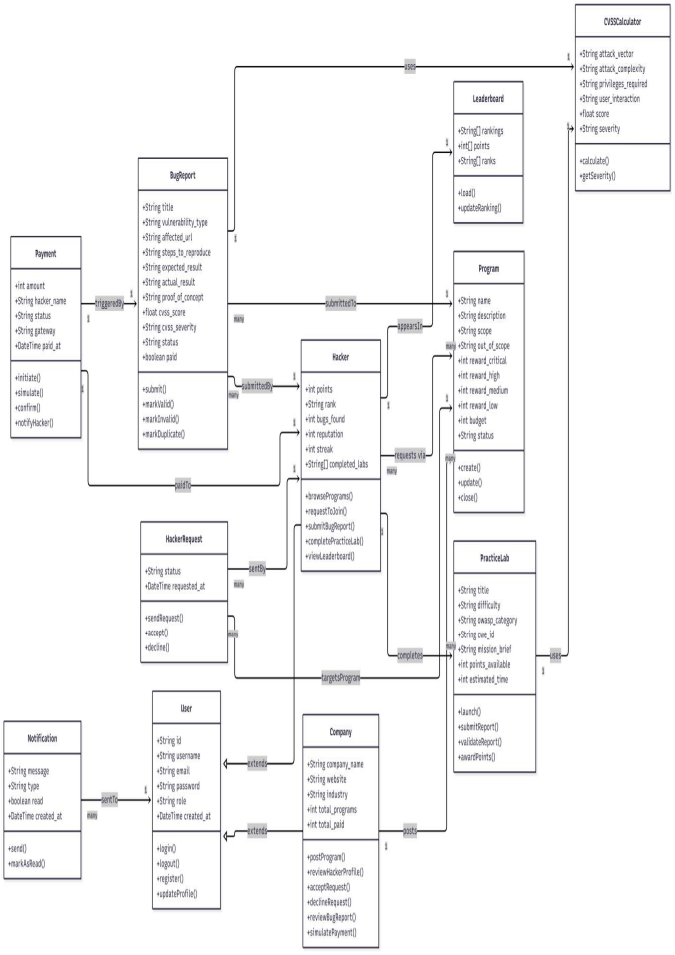


***Fig. 1**. Diagram of the Proposed BugHunters Platform's System Architecture*

B. **User Interaction and Program Access Layer*:*** The BugHunters platform's entrance point, the User Interaction Layer, facilitates smooth communication between hackers and businesses. In order to guarantee that users can safely access the system according to their roles, this layer controls user registration, authentication, and role-based access. While businesses can build and manage programs by specifying scope, payouts, and vulnerability categories, hackers can browse available bug bounty programs, request participation, and access practice labs.By verifying user requests and upholding restricted program access, the system guarantees organized participation. This layer's user-friendly interface and effective backend connectivity enable instantaneous real-time engagement. It guarantees that users may simply traverse the platform and participate in ethical hacking activities within predetermined limitations by offering a clear and coordinated workflow.

C. **Vulnerability Testing and Practice Lab Module:** Users can conduct security testing on simulated apps in a controlled environment thanks to the Vulnerability Testing Module. Users are redirected to practice labs that mimic real-world vulnerabilities like SQL Injection, Cross-Site Scripting (XSS), and authentication errors rather than engaging with actual systems.Based on criteria like the OWASP Top 10, each lab is created with predetermined goals, degrees of complexity, and vulnerability classifications. In order to avoid any unintentional effects on actual systems, the system makes sure that all testing operations take place in a secure and isolated setting. Users can experiment, study attack strategies, and comprehend system behavior with this module without running the danger of legal repercussions.The module also monitors user progress and interactions, making sure that every activity supports learning objectives and performance assessment.

1) **Bug Reporting and Validation Module:** Users can record vulnerabilities in an organized and expert manner with the help of the Bug Reporting Module. Users must supply comprehensive details, including the type of vulnerability, impacted components, reproduction instructions, expected and actual outcomes, and proof of concept.Following submission, the report is examined and classified as legitimate, invalid, or duplicate using a validation mechanism. By ensuring that only meaningful and accurate reports are accepted, this validation process helps users appreciate the significance of appropriate documentation and responsible disclosure.Additionally, the module keeps track of report history and feedback, which enables users to gradually enhance their reporting abilities. The system gets users ready for real-world bug bounty programs by imposing organized reporting requirements.

D. **Severity Analysis and CVSS Scoring Engine:** Using a uniform grading methodology, the Severity Analysis Module is in charge of assessing the effect of vulnerabilities that have been disclosed. A CVSS-based calculator that evaluates elements including attack vector, attack complexity, privileges needed, user engagement, and impact on confidentiality, integrity, and availability is included into the platform.The method classifies vulnerabilities into levels like Low, Medium, High, and Critical and creates a severity score based on these factors. This guarantees evaluation consistency and synchronizes the platform with industry norms.The scoring process is an essential part of the system since it not only aids in comprehending the impact of vulnerabilities but also plays a critical role in reward distribution and ranking updates.

E. **Reward Management and Leaderboard System:** Reward distribution for legitimate bug reports is managed by the Reward Management Module. The method allocates prizes according to predetermined severity levels established by the business when a vulnerability has been verified and rated. The platform simulates payments to mimic actual bug bounty procedures.The Leaderboard System simultaneously modifies user reputation scores, rankings, and points

according to their performance. Metrics including problems discovered, legitimate reports, and participation consistency are monitored by this module.The platform fosters a competitive and inspiring atmosphere that promotes ongoing learning and active engagement by fusing reward distribution with ranking systems.

F. **Notification and Feedback System:** Real-time communication between users and the system is guaranteed via the Notification Module. It offers information on program approvals, prize distribution, report status, and other significant events.Users get feedback on their reports, which enables them to identify errors and make better submissions in the future. This ongoing feedback loop keeps users interested in the platform and improves the learning process.In order to provide transparency and action traceability, the system additionally logs all notifications and activity.

G. **Data Management and System Integration Layer:** The platform's core is the Data Management Layer, which manages the processing and storage of all system data, including users, applications, bug reports, payments, and alerts.Each component of the system, including reporting, scoring, and rewards, functions independently while preserving smooth integration thanks to its modular architecture. Scalability, dependability, and effective system performance are thus guaranteed.Future improvements like automatic validation, enhanced analytics, and AI-based vulnerability recommendations are also supported by the architecture. The platform guarantees seamless operation and a consistent user experience by combining all modules into a single system.

IV. RESULTS AND DISCUSSION

The effectiveness of the BugHunters platform in offering an organized, safe, and engaging setting for ethical hacking exercise was assessed experimentally. The review concentrated on important elements such interface usability, bug report validation accuracy, the efficacy of the CVSS-based severity ranking system, and overall user involvement through gamification features. To guarantee dependable and seamless system operation, the integration of several modules, including practice labs, programs, reporting, and notifications, was also investigated.

1. BugHunters Landing Page

The main interface via which users engage with the platform is the landing page. It makes the system's goal—ethical hacking education and bug bounty simulation—clearly apparent and offers simple access to modules like Learning, Practice Labs, Programs, and Leaderboard. Both novice and seasoned users may easily navigate the interface thanks to its simple and intuitive design. Quick-action buttons like "Start Learning" and "View Labs" increase user engagement and simplify navigation.

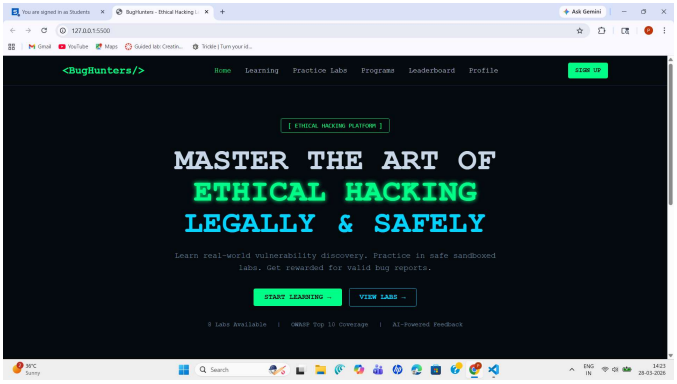


Fig.2. BugHunters Landing Page – Ethical Hacking Platform Interface

2. Company Dashboard

The Programs Dashboard verifies that the system can effectively handle and display several bug bounty programs. Users can investigate and seek participation by seeing program characteristics including scope, reward structure, and status.Real-time data aggregation is demonstrated by the inclusion of metrics such as total programs, reports, active hackers, and total rewards. This improves openness and gives users information about platform performance and activity.

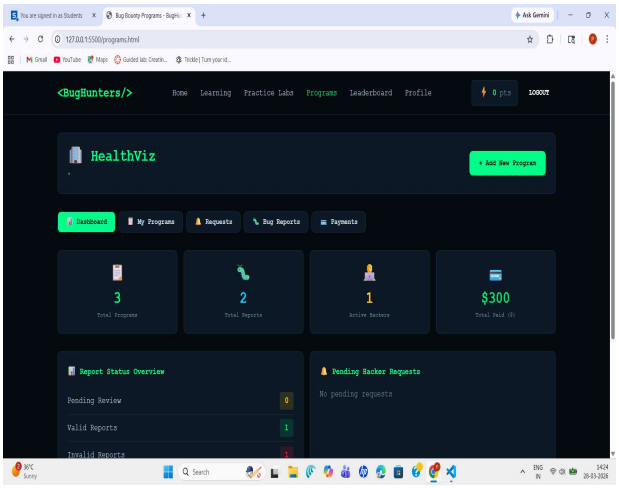


Fig.3. Company Dashboard

3. Bug Reports and Validation Result

The system's capacity to manage structured vulnerability submissions is demonstrated by the Bug Reports module. Only significant findings are rewarded since reports are clearly classified as legitimate or invalid.The system adheres to normal reporting procedures, as seen by the presentation of vulnerability type, CVSS score, and impacted URLs. The efficiency of the validation process in preserving report quality and avoiding duplicate or inaccurate submissions is demonstrated in this module.

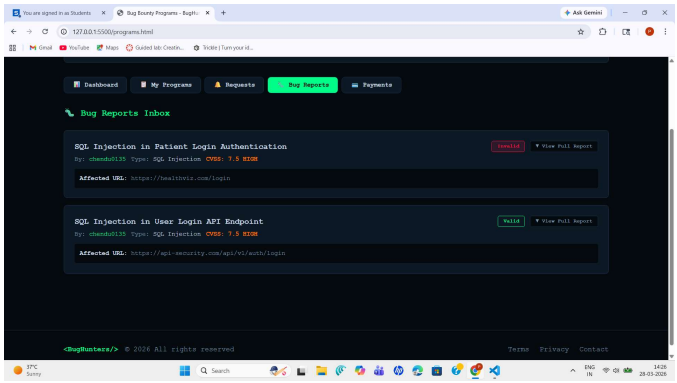


FIG. 4. BUG REPORTS – VALIDATION AND CLASSIFICATION

4. Learning Module Interface

The Learning Module attests to the platform's capacity to offer both theoretical information and hands-on instruction. Along with practical examples and attack scenarios, it provides organized explanations of vulnerabilities like SQL Injection.By ensuring that learners acquire conceptual understanding prior to undertaking practical labs, this integration of learning content enhances overall skill development and learning efficiency.

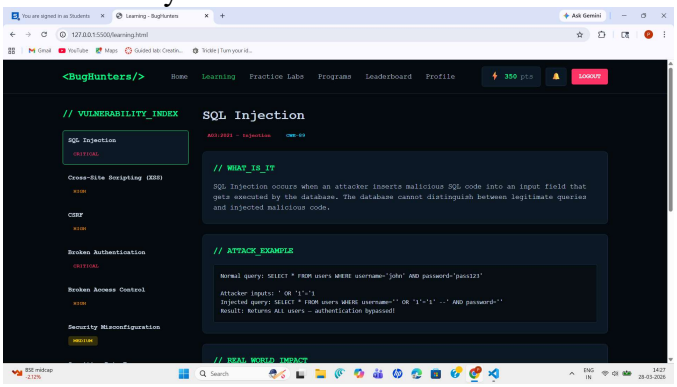
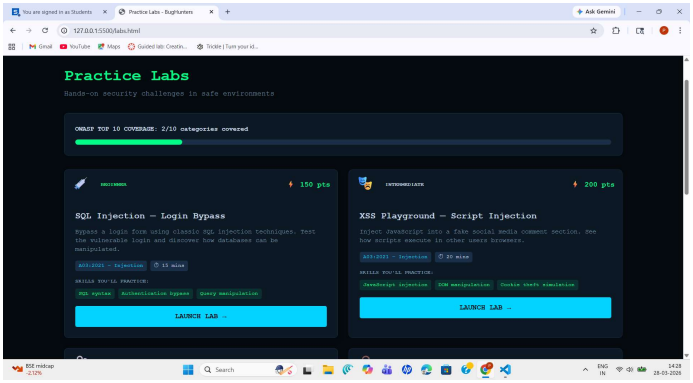


Fig. 5. Learning Module – Vulnerability Explanation Interface

5. Practice Labs Execution



The Practice Labs module offers practical training opportunities in a secure setting. Labs like SQL Injection and XSS, which mimic real-world vulnerabilities without impacting real systems, can be started by users. Structured learning progression is ensured by the inclusion of skill tags, projected time, and difficulty levels. The system's capacity to connect theoretical understanding with real-world application is confirmed by this module.

Fig. 6: Practice Labs – Hands-on Security Testing

IV. V. REFERENCES

[1] Allodi, L., & Massacci, F. (2017). "Security Events and Vulnerability Data for Cybersecurity Risk Estimation." *IEEE Transactions on Dependable and Secure Computing.*

[2] Beuran, R., Tang, D., Pham, C., Chinen, K., Tan, Y., & Shinoda, Y. (2018). "Integrated Framework for Hands-On Cybersecurity Training: CyTrONE." *IEEE Access.*

[3] Feng, Y., Li, Y., & Xu, H. (2021). "ContractSafeguard: Practical Bug Bounty Platform for Smart Contracts with Intel SGX." *IEEE Transactions on Information Forensics and Security.*

[4] Hamari, J., Koivisto, J., & Sarsa, H. (2014). "Does Gamification Work? A Literature Review of Empirical Studies on Gamification." *Proceedings of the 47th Hawaii International Conference on System Sciences (HICSS), IEEE.*

[5] Mirkovic, J., & Peterson, P. (2019). "Class Capture-the-Flag Exercises." *IEEE Security and Privacy.*

[6] Zhao, M., Liao, X., & Bao, T. (2022). "An Agent-Based Modeling Approach to Designing and Optimizing Bug Bounty Programs for Cybersecurity in Developing Countries." *IEEE Transactions on Network and Service Management.*

[7] OWASP Foundation (2021). *OWASP Top 10: The Ten Most Critical Web Application Security Risks.*

[8] NIST (National Institute of Standards and Technology) (2018). *Framework for Improving Critical Infrastructure Cybersecurity.*

[9] Votipka, D., Rabinowitz, M., Mazurek, M. L., & Shacham, H. (2020). "An Observational Investigation of Reverse Engineers' Processes." *IEEE Symposium on Security and Privacy.*

[10] Ruohonen, J., & Leppänen, V. (2017). "Toward Validation of the Common Vulnerability Scoring System." *IEEE Transactions on Dependable and Secure Computing.*

[11] ENISA (European Union Agency for Cybersecurity) (2020). *Cybersecurity Skills Development in the EU.*

[12] Grossman, J. (2017). *The Bugcrowd Vulnerability Rating Taxonomy.* Bugcrowd Inc.

[13] Sharma, T., & Dash, S. (2020). "A Survey on Web Application Vulnerabilities (SQL Injection, XSS) and Prevention Techniques." *IEEE International Conference on Computational Intelligence and Computing Research.*

[14] Pope, N., & Bodeau, D. (2018). "Principles of Secure Software Development." *IEEE Security & Privacy.*

[15] Bacudio, A. G., Yuan, X., Chu, B., & Jones, M. (2011). "An Overview of Penetration Testing." *International Journal of Network Security & Its Applications.*

[16] Kumar, S., & Carley, K. M. (2019). "Cybersecurity Training Using Simulation and Gamification." *IEEE Conference on Intelligence and Security Informatics.*

[17] Sabottke, C., Suciu, O., & Dumitras, T. (2015). "Vulnerability Disclosure in the Age of Social Media: Exploiting Twitter for Predicting Real-World Exploits." *USENIX Security Symposium.*

[18] Alshamrani, A., Myneni, S., Chowdhary, A., & Huang, D. (2019). "A Survey on Advanced Persistent Threats: Techniques, Solutions, Challenges, and Research Opportunities." *IEEE Communications Surveys & Tutorials.*

