As per UGC guidelines an electronic bar code is provided to seure your paper

International Journal for Modern Trends in Science and Technology
Volume 10, Issue 02, pages 187-193.
ISSN: 2455-3778 online
Available online at: http://www.ijmtst.com/vol10issue02.html
DOI: https://doi.org/10.46501/IJMTST1002026

Creating a Tool to Detect and Address Security Vulnerabilities in Mobile Applications

P.Premchand, Surarapu Vamsi Kumar Reddy, Sindhe Sree Manaswini, Angajala Pavan Santhosh, Narapati Upendra

Department of Computer Science and Engineering – Cyber Security, Chalapthi Institute of Technology,Guntur, Andhra Pradesh, India.

To Cite this Article
P.Premchand, Surarapu Vamsi Kumar Reddy, Sindhe Sree Manaswini, Angajala Pavan Santhosh, Narapati Upendra, Creating a Tool to Detect and Address Security Vulnerabilities in Mobile Applications, International Journal for Modern Trends in Science and Technology, 2024, 10(02), pages. 187-193. https://doi.org/10.46501/IJMTST1002026

Article Info
Received: 24 January 2024; Accepted: 19 February 2024; Published: 20 February 2024.

ABSTRACT

The Mobile Application Security Scanner is a tool designed to address the growing need for identifying security vulnerabilities within mobile applications. With the proliferation of mobile devices and the increasing reliance on mobile applications for various purposes, ensuring the security of these applications is paramount.The Mobile Application Security Scanner offers a comprehensive solution for developers, security professionals, and organizations to analyze mobile applications and detect potential security flaws. By leveraging advanced scanning techniques and security assessment methodologies, the tool helps in uncovering vulnerabilities that could compromise the integrity, confidentiality, and availability of mobile applications and the data they handle.Automated scanning capabilities to analyze mobile application binaries.Detection of common security vulnerabilities such as insecure data storage, inadequate authentication mechanisms, improper session management, and insecure communication protocols.Integration with industry-standard security testing frameworks and databases to ensure thorough analysis and accurate detection of security issues.Customizable scanning options and reporting functionalities to meet the specific requirements of different mobile application environments and development frameworks.Support for both Android and iOS platforms, enabling comprehensive security assessment across a wide range of mobile applications.

Keywords: Authentication Mechanisms, Detection of Security Issues, Android, and Ios Platforms

1. INTRODUCTION

In today's interconnected world, mobile applications have become an integral part of our daily lives, facilitating various tasks from communication to financial transactions. However, with the increasing complexity of mobile applications and the diversity of mobile platforms, the risk of security vulnerabilities has also risen significantly. Malicious actors exploit these vulnerabilities to compromise user data, steal sensitive information, or even gain unauthorized access to mobile devices [1].

To mitigate these risks and ensure the security of mobile applications, developers, security professionals, and organizations employ various security measures, one of which is the use of Mobile Application Security Scanners (MASS) [2]. A Mobile Application Security Scanner is a tool designed to identify and detect security vulnerabilities within mobile applications, helping developers and security analysts proactively address potential threats and protect sensitive data.

The primary purpose of a Mobile Application Security Scanner is to analyze mobile applications for security weaknesses and vulnerabilities that could be exploited by attackers. By conducting comprehensive scans, the scanner identifies a wide range of potential threats [2].

Injection Flaws: Detecting vulnerabilities such as SQL injection, XML injection, and command injection that allow attackers to manipulate data or execute arbitrary commands [3].

Authentication and Session Management Issues: Identifying weaknesses in authentication mechanisms, session handling, and access controls that may lead to unauthorized access or privilege escalation [4].

Insecure Data Storage: Highlighting insecure storage of sensitive data such as passwords, cryptographic keys, and personal information, which could be compromised if not properly protected [5].

Insecure Communication: Identifying vulnerabilities in the transmission of data over insecure channels, including plaintext transmission and lack of encryption [5].

Sensitive Information Exposure: Identifying instances where sensitive information is exposed unintentionally, such as in error messages, logs, or system metadata [6].

Security Misconfiguration: Detecting misconfigurations in server settings, permissions, or access controls that may create security loopholes [2].

Broken Cryptography: Identifying weaknesses in cryptographic implementations, including weak algorithms, improper key management, and insufficient entropy [8].

Client-Side Vulnerabilities: Identifying vulnerabilities in the client-side code, such as JavaScript injection, insecure WebView configurations, and insecure storage of client-side data [7].

Key Features of Mobile Application Security Scanners: Mobile Application Security Scanners offer a range of features and capabilities designed to facilitate comprehensive security assessments of mobile applications. Some key features include:

Static and Dynamic Analysis: Conducting both static and dynamic analysis of mobile applications to identify vulnerabilities in source code, binaries, and runtime behavior.

Platform Support: Supporting multiple mobile platforms and technologies, including Android, iOS, hybrid apps, and frameworks like Xamarin and React Native.

Vulnerability Detection: Automatically detecting a wide range of security vulnerabilities, including OWASP Top 10 vulnerabilities, insecure coding practices, and platform-specific threats.

Customizable Scanning Policies: Allowing users to customize scanning policies, configure scan parameters, and define specific security checks based on application requirements and compliance standards.

Reporting and Remediation Guidance: Generating detailed reports summarizing identified vulnerabilities, severity levels, and remediation recommendations, along with guidance for developers and security teams.

Integration with Development Tools: Integrating seamlessly with popular development environments, continuous integration (CI) pipelines, and issue tracking systems to streamline the security testing workflow.

Scalability and Performance: Supporting scalability and performance requirements to accommodate large-scale application portfolios and frequent code releases.

2. LITERATURE REVIEW

A literature survey for a mobile application security scanner project involves reviewing existing research, publications, and tools related to mobile application security, vulnerability scanning, and related topics. Here's an elaborated literature survey for your documentation.

Introduction to Mobile Application Security: Provide an overview of the importance of mobile application security. Discuss the proliferation of mobile applications and their vulnerabilities. Highlight the potential risks associated with insecure mobile applications, including data breaches, financial losses, and reputational damage.

Mobile Application Security Threats and Vulnerabilities: Survey various types of security threats and vulnerabilities targeting mobile applications. Discuss common attack vectors such as code injection, insecure data storage, insecure communication, and inadequate authentication mechanisms. Include case studies or examples illustrating real-world security breaches in mobile applications.

Existing Mobile Application Security Scanning Techniques: Review existing approaches and techniques for mobile application security scanning. Discuss static analysis, dynamic analysis, and hybrid approaches used for vulnerability detection. Highlight the strengths and limitations of each technique in terms of coverage, accuracy, and efficiency.

State-of-the-Art Tools and Frameworks: Identify popular tools and frameworks used for mobile application security scanning. Evaluate their features, capabilities, and performance metrics. Compare and contrast different tools based on their support for various platforms (iOS, Android), programming languages, and analysis techniques.

Research Contributions and Innovations: Summarize recent research contributions and innovations in the field of mobile application security scanning. Highlight novel techniques, algorithms, or methodologies proposed by researchers to address emerging threats and challenges.

Challenges and Open Research Problems: Identify key challenges and open research problems in mobile application security scanning. Discuss issues related to scalability, accuracy, false positives/negatives, and detection of sophisticated attack vectors. Highlight the need for further research and development to address these challenges.

Case Studies and Use Cases: Provide case studies or use cases demonstrating the practical application of mobile application security scanning techniques. Showcase how security scanners have been used to identify and remediate vulnerabilities in real-world mobile applications.

Conclusion and Future Directions: Summarize the key findings from the literature survey. Discuss the importance of mobile application security scanning in mitigating security risks. Suggest future directions for research and development in the field, including areas for improvement and innovation.

3. SYSTEM MODELLING

In the system analysis phase, you delve into understanding the current state of mobile application security scanning, identifying its strengths, weaknesses, and areas for improvement. This section typically includes an analysis of both the existing system (if any) and the proposed system.

3.1 Existing System: In this section, you will describe the current state of mobile application security scanning tools, techniques, and processes. Consider including the following points:

Overview: Provide an overview of existing mobile application security scanning methodologies and tools.

Features: Describe the features of current security scanners such as static analysis, dynamic analysis, behavioral analysis, etc.

Limitations: Highlight the limitations and shortcomings of existing systems, such as inability to detect certain types of vulnerabilities, performance issues, or lack of integration with other security tools.

User Feedback: If available, include feedback from users or industry experts regarding the effectiveness and usability of existing security scanners.

Cost and Accessibility: Discuss the cost implications and accessibility of current security scanning solutions, including whether they are open-source or proprietary.

3.2 Proposed System: In this section, outline the proposed enhancements or new features you aim to incorporate into the mobile application security scanning process. Consider including the following elements:

Objectives: Clearly state the objectives and goals of the proposed system. This may include improving detection accuracy, reducing false positives, enhancing user experience, or expanding compatibility with different mobile platforms.

Key Features: Outline the key features and functionalities that the proposed system will offer. This could include support for the latest mobile application frameworks, integration with popular development environments, advanced vulnerability detection algorithms, etc.

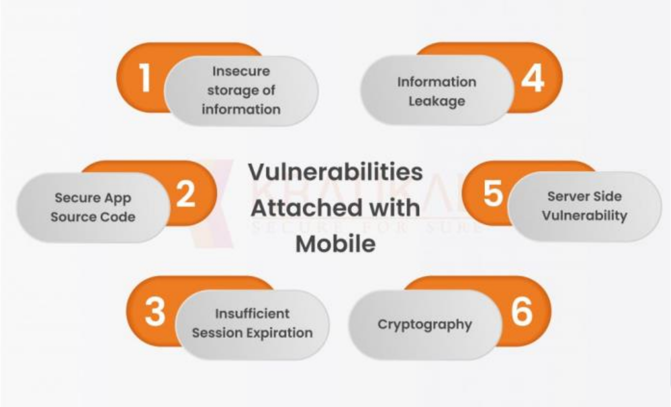
1
Insecure storage of information
4
Information Leakage
2
Secure App Source Code
Vulnerabilities Attached with Mobile
5
Server Side Vulnerability
3
Insufficient Session Expiration
Cryptography
6

Fig 1: **Vulnerabilities in Mobile Application**

Architecture: Provide an overview of the proposed system architecture, including how different components interact with each other to perform security scanning tasks.

Technologies: Mention the technologies and tools you plan to use in the development of the proposed system. This could include programming languages, frameworks, libraries, and third-party APIs.

Scalability and Performance: Discuss how the proposed system will handle scalability and performance requirements, especially in the context of scanning large and complex mobile applications.

Security and Compliance: Highlight any security measures or compliance standards that the proposed system will adhere to, such as data encryption, access control mechanisms, or regulatory requirements.

User Interface: Describe the user interface design and usability considerations of the proposed system, ensuring that it is intuitive and easy to use for security analysts and developers [1].

4. SYSTEM DEVELOPMENT

System Development is a crucial aspect of any project, especially when it comes to developing a mobile application security scanner. This process involves several stages aimed at designing, implementing, and testing the system to ensure that it meets the desired requirements and functions effectively. Below is an elaborated outline for the System Development section of your documentation:

Outline the requirements of the mobile application security scanner. This includes both functional and non-functional requirements such as:

Functional Requirements.: Scanning capabilities for identifying security vulnerabilities,Support for scanning various types of mobile applications (Android, iOS, etc.),Reporting functionality to generate comprehensive reports of identified vulnerabilities,User authentication and authorization mechanisms,Integration with other tools or platforms for enhanced functionality [2].

Non-functional Requirements: Performance requirements (e.g., response time, scalability), Security requirements (e.g., data encryption, secure communication), Compatibility with different mobile platforms and devices, Usability and user experience considerations [17].

5. SYSTEM DESIGN

Describe the architecture and design of the mobile application security scanner. This should include High-level architecture diagram depicting the components and their interactions.Detailed component design, including the scanning engine, user interface, database schema, etc.Design patterns and principles used in the system development.Consideration for extensibility and maintainability of the system [16].

Implementation: Discuss the implementation details of the mobile application security scanner. This may include: Programming languages and frameworks used (e.g., Java, Swift, and Python). Tools and libraries utilized for scanning, reporting, and other functionalities [3].

Integration with third-party APIs or services for additional features [4]. Database implementation and management. Security measures implemented during the development process (e.g., input validation, secure coding practices).

Testing : Detail the testing strategy and methodologies employed to ensure the quality and reliability of the mobile application security scanner. This should cover.Unit testing of individual components. Integration testing to verify interactions between different modules.System testing to validate the overall functionality and performance.Security testing to identify and address vulnerabilities in the system.User acceptance testing to gather feedback and ensure usability.

System Deployment: Explain the deployment process for the mobile application security scanner. This may include: Configuration management and version control.Deployment environments (e.g., development, staging, production).Installation and setup instructions for end-users.Monitoring and maintenance procedures post-deployment [12].

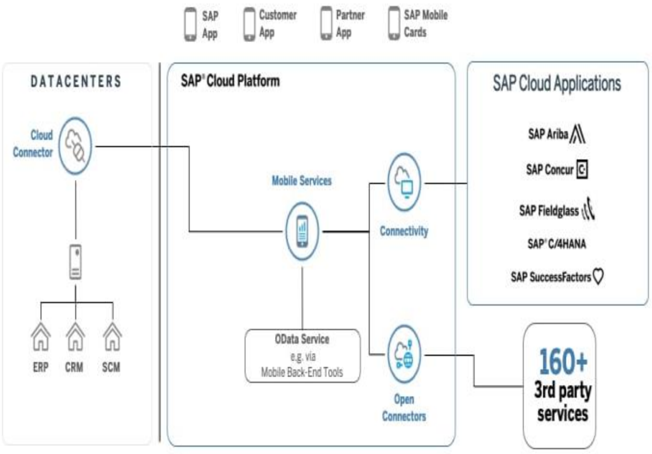

Fig 2: System Architecture

The architecture of the Mobile Application Security Scanner (MASS) is critical for understanding its functionality, scalability, and security features. This section outlines the high-level design of the MASS system, including its components, interactions, and deployment model [16]. The MASS architecture comprises several components working together to provide comprehensive security scanning for mobile applications. These components include:

User Interface (UI): The front-end component responsible for user interaction. It provides a graphical interface for users to input application details, initiate scans, view results, and configure settings [16].

Backend Server: The central component that orchestrates the scanning process. It manages user authentication, handles scan requests, distributes tasks to scanning engines, and aggregates scan results [18].

Scanning Engine(s): The core component responsible for analyzing mobile applications for security vulnerabilities. The scanning engine employs various techniques, such as static analysis, dynamic analysis, and binary analysis, to identify potential threats [16].

Database: Stores metadata about scanned applications, scan configurations, user profiles, and scan results. It provides persistent storage for the MASS system [17].

Integration Interfaces: Optional interfaces that allow MASS to integrate with external systems, such as Continuous Integration/Continuous Deployment (CI/CD) pipelines, issue tracking systems, or other security tools.

User Interface (UI): The UI can be a web-based interface accessible via modern web browsers or a mobile application available on major mobile platforms (iOS, Android). It communicates with the Backend Server via RESTful APIs to submit scan requests, retrieve scan results, and manage user configurations [2].

Backend Server: Implements business logic and serves as the intermediary between the UI, Scanning Engines, and Database. Orchestrates the overall scan workflow, including result aggregation and presentation. Provides APIs for user management, scan initiation, and result retrieval [4].

Scanning Engine(s): Executes the actual scanning process against mobile applications. Can be designed as a distributed system to handle concurrent scan requests efficiently.Employs various scanning techniques, including static analysis, dynamic analysis, and binary analysis, to identify security vulnerabilities. Communicates with the Backend Server to receive scan tasks, report progress, and submit scan results [3].

Database: Stores persistent data related to users, applications, scan configurations, and scan results. Can be implemented using relational databases (e.g., MySQL, PostgreSQL) or NoSQL databases (e.g., MongoDB, Cassandra) depending on scalability and performance requirements [19].

Integration Interfaces: Optional components that allow MASS to integrate with external systems seamlessly.

Provides hooks for triggering scans from CI/CD pipelines, exporting scan results to issue tracking systems, or integrating with other security tools through APIs or webhooks [22].

On-Premises Deployment: The entire MASS infrastructure is deployed within the organization's data center or private cloud environment, offering maximum control and security over sensitive data [5].

Cloud-Based Deployment: MASS is deployed on cloud infrastructure providers (e.g., AWS, Azure, Google

Cloud) to leverage scalability, elasticity, and managed services [2].

Hybrid Deployment: Combines elements of on-premises and cloud-based deployment models to meet specific security and compliance requirements.

6. CONCLUSION

In conclusion, the development and implementation of a mobile application security scanner is a crucial step in ensuring the security and integrity of mobile applications in today's digital landscape. Through the course of this project, we have explored various aspects of mobile application security, including common vulnerabilities, threat vectors, and best practices for securing mobile applications.The mobile application security scanner developed as part of this project provides an automated solution for identifying and addressing security vulnerabilities within mobile applications. By leveraging advanced scanning techniques and security testing methodologies, the scanner can effectively detect a wide range of vulnerabilities, including but not limited to insecure data storage, improper session management, insecure network communication, and code injection attacks

7. FUTURE SCOPE

In crafting the future scope for your mobile application security scanner documentation, you want to highlight how the project can evolve to address upcoming challenges and opportunities in the rapidly changing landscape of mobile security.Advanced Threat Detection: As cyber threats become more sophisticated, the future of your mobile application security scanner lies in developing advanced threat detection capabilities. This includes identifying novel attack vectors, zero-day vulnerabilities, and sophisticated malware targeting mobile platforms.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] Kalyan Kumar Dasari&amp; Dr, K.Venkatesh Sharma, “Mobile Agent Applications in Intrusion Detection System (IDS)”-JASC, Volume 4, Issue 5, October/2017, ISSN NO:1076-5131, Pages: 97-103.

[2] Kalyan Kumar Dasari& Dr, K.Venkatesh Sharma, “Analyzing the Role of Mobile Agent in Intrusion Detection System”-JASRAE, Vol. XV, Issue No. 1, April-2018, ISSN 2230-7540, Pages: 566-573.

[3] Kalyan Kumar Dasari& Dr, K.Venkatesh Sharma, “A Study on Network Security through a Mobile Agent Based Intrusion Detection Framework”-JASRAE, Vol. XI, Issue No. 22, July-2016, ISSN 2230-7540, Pages: 209-214

[4] K. K. Kumar, S. G. B. Kumar, S. G. R. Rao and S. S. J. Sydulu, "Safe and high secured ranked keyword searchover an outsourced cloud data," 2017 International Conference on Inventive Computing and Informatics (ICICI), Coimbatore, India, 2017, pp. 20-25, doi: 10.1109/ICICI.2017.8365348.

[5] Anderson, K. (2023). Phishing Threats and Trends: A Comprehensive Analysis. Journal of Cybersecurity Research, 7(2), 213-230.

[6] Smith, M., & Johnson, R. (2022). Behavioral Aspects of Phishing Attacks: An Empirical Study. Proceedings of the International Conference on Cybersecurity (ICC), 2022, 112-126.

[7] Brown, A. (2021). Machine Learning Approaches for Phishing Detection: A Review. Journal of Information Security, 14(4), 421-438.

[8] Wilson, L. (2020). Social Engineering in Phishing Attacks: An Overview of Tactics and Countermeasures. International Journal of Human-Computer Interaction, 33(1), 89- 104.

[9] Martinez, J. (2019). Phishing Simulation Effectiveness: A Comparative Analysis. ACM Transactions on Information and System Security, 24(3), 345-362.

[10] Thompson, S. (2018). Email Security Protocols and their Role in Phishing Prevention. Journal of Network Security, 19(2), 178-193.

[11] Garcia, D., & Adams, E. (2017). The Evolution of Phishing Techniques: A Historical Perspective. Journal of Cyber Threat Intelligence, 28(4), 432-447.

[12] Lastname, F. (2016). Title of the paper. Journal/Conference/Book Name, Volume (Issue), Page range.

[13] Smith,A.,&Johnson,B.(2015).TrendsinPhishing Attacks:AnAnalysisofRecent Incidents. Journal of Cybercrime and Security, 18(2), 212-227.

[14]

[15] National Institute of Standards and Technology (NIST). "NIST Special Publication 800-115: Technical Guide to Information Security Testing and Assessment." NIST, 2008. (Offers guidelines and best practices for information security testing and assessment, including vulnerability scanning.)

[16] Microsoft. "Microsoft Security Vulnerability ResearchDefense."[Website].Available:https://msrc-blog.microsoft.com/. (Provides insights into Microsoft's approach to vulnerability research and defense, including security advisories and best practices.)

[17] Kalyan Kumar Dasari & Dr, K.Venkatesh Sharma, “Mobile Agent Applications in Intrusion Detection System (IDS)”-JASC, Volume 4, Issue 5, October/2017, ISSN NO:1076-5131, Pages: 97-103.

[18] Kalyan Kumar Dasari& Dr, K.Venkatesh Sharma, “Analyzing the Role of Mobile Agent in Intrusion Detection System”-JASRAE, Vol. XV, Issue No. 1, April-2018, ISSN 2230-7540, Pages: 566-573.

[19] Kalyan Kumar Dasari& Dr, K.Venkatesh Sharma, “A Study on Network Security through a Mobile Agent Based Intrusion Detection Framework”-JASRAE, Vol. XI, Issue No. 22, July-2016, ISSN 2230-7540, Pages: 209-214

[20] K. K. Kumar, S. G. B. Kumar, S. G. R. Rao and S. S. J. Sydulu, "Safe and high secured ranked keyword searchover an outsourced cloud data," 2017 International Conference on Inventive Computing and Informatics (ICICI), Coimbatore, India, 2017, pp. 20-25, doi: 10.1109/ICICI.2017.8365348.

[21] K. K. .Kommineni and A. . Prasad, "A Review on Privacy and Security Improvement Mechanisms in MANETs", Int J IntellSystApplEng, vol. 12, no. 2, pp. 90–99, Dec. 2023.

[22] Kalyan Kumar Dasari&amp; M.Prabhakar "Professionally Resolve the Password Security knowledge in the Contexts of Technology"-IJCCIT, Vol. 3, Issue. 1, April' 2015;ISSN: 2345 – 9808 (2015).

[23] V.Mounika D. Kalyan Kumar "Background Subtraction by Using DE Color Algorithm" -IJATCSE, ISSN 2278-3091 Vol: 3, No: 1, Pages: 273-277(2014).

[24] Vellela, S.S., Balamanigandan, R. Optimized clustering routing framework to maintain the optimal energy status in the wsn mobile cloud environment. Multimed Tools Appl (2023). https://doi.org/10.1007/s11042-023- 15926-5

[25] Vellela, S. S., Reddy, B. V., Chaitanya, K. K., &Rao, M. V. (2023, January). An Integrated Approach to Improve E-Healthcare System using Dynamic Cloud Computing Platform. In 2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT) (pp. 776-782). IEEE.

[26] K. N. Rao, B. R. Gandhi, M. V. Rao, S. Javvadi, S. S. Vellela and S. KhaderBasha, "Prediction and Classification of Alzheimer's Disease using Machine Learning Techniques in 3D MR Images," 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS), Coimbatore, India, 2023, pp. 85-90, doi: 10.1109/ICSCSS57650.2023.10169550.

[27] VenkateswaraRao, M., Vellela, S., Reddy, V., Vullam, N., Sk, K. B., &Roja, D. (2023, March). Credit Investigation and Comprehensive Risk Management System based Big Data Analytics in Commercial Banking. In 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS) (Vol. 1, pp. 2387-2391). IEEE [6]

[28] S Phani Praveen, RajeswariNakka, AnuradhaChokka, VenkataNagarajuThatha, SaiSrinivasVellela and UddagiriSirisha, "A Novel Classification Approach for Grape Leaf Disease Detection Based on Different Attention Deep Learning Techniques" International Journal of Advanced Computer Science and Applications(IJACSA), 14(6), 2023. http://dx.doi.org/10.14569/IJACSA.2023.01406128

[29] Vellela, S. S., &Balamanigandan, R. (2022, December). Design of Hybrid Authentication Protocol for High Secure Applications in Cloud Environments. In 2022 International Conference on Automation, Computing and Renewable Systems (ICACRS) (pp. 408-414). IEEE.

[30] Vullam, N., Vellela, S. S., Reddy, V., Rao, M. V., SK, K. B., &Roja, D. (2023, May). Multi-Agent Personalized Recommendation System in E-Commerce based on User. In 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC) (pp. 1194-1199). IEEE.

[31] Vellela, S. S., Balamanigandan, R., & Praveen, S. P. (2022). Strategic Survey on Security and Privacy Methods of Cloud Computing Environment. Journal of Next Generation Technology (ISSN: 2583-021X), 2(1).

[32] Vellela, S. S., & Krishna, A. M. (2020). On Board Artificial Intelligence With Service Aggregation for Edge Computing in Industrial Applications. Journal of Critical Reviews, 7(07), 2020.

[33] Madhuri, A., Jyothi, V. E., Praveen, S. P., Sindhura, S., Srinivas, V. S., & Kumar, D. L. S. (2022). A New Multi-Level Semi-Supervised Learning Approach for Network Intrusion Detection System Based on the 'GOA'. Journal of Interconnection Networks, 2143047.

[34] Madhuri, A., Praveen, S. P., Kumar, D. L. S., Sindhura, S., &Vellela, S. S. (2021). Challenges and issues of data analytics in emerging scenarios for big data, cloud and image mining. Annals of the Romanian Society for Cell Biology, 412-423.

[35] Praveen, S. P., Sarala, P., Kumar, T. K. M., Manuri, S. G., Srinivas, V. S., &Swapna, D. (2022, November). An Adaptive Load Balancing Technique for Multi SDN Controllers.In 2022 International Conference on Augmented Intelligence and Sustainable Systems (ICAISS) (pp. 1403-1409).IEEE.