Unauthorized Access and Privilege Escalation

[1]Ms. M. BUVANA, [2]LATHIKA SRI.B, [3]MAGIZHOVIYA.S, [4]MANISHA.K, [5]POOJITHA.M

1,2,3,4,57Student, Information Technology, Sri Shakthi Institute of Engineering and Technology, Coimbatore, Tamil Nadu, India.

---***---

Abstract - Traditional access control systems store user roles in centralized databases, making them vulnerable to hacking and insider attacks. Secure Auth uses blockchain-based Role-Based Access Control (RBAC) to store user roles securely on the blockchain. Smart contracts enforce access rules, preventing unauthorized changes or privilege escalations. This ensures tamper-proof, transparent, and decentralized access control. Users can verify their roles anytime, and only authorized actions are allowed. This approach enhances security, transparency, and trust in authentication systems.

***Key Words*:** Access Control, Cybersecurity, RBAC, Blockchain, Smart Contracts, Unauthorized Access, Privilege Escalation, Secure Auth.

1. INTRODUCTION

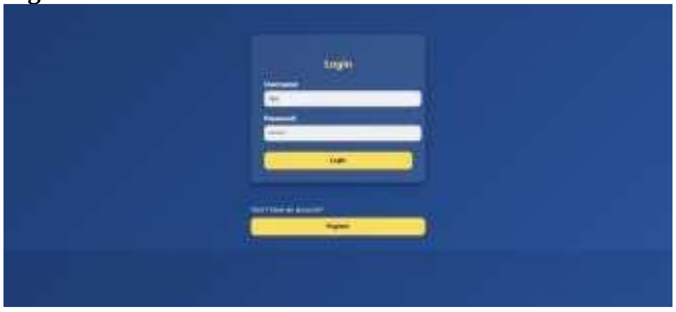
Traditional access control systems are prone to attacks due to centralized storage of user roles. Secure Auth uses blockchain-based RBAC and smart contracts to provide secure, transparent, and tamper-proof access control. This approach prevents unauthorized access and privilege escalation effectively.

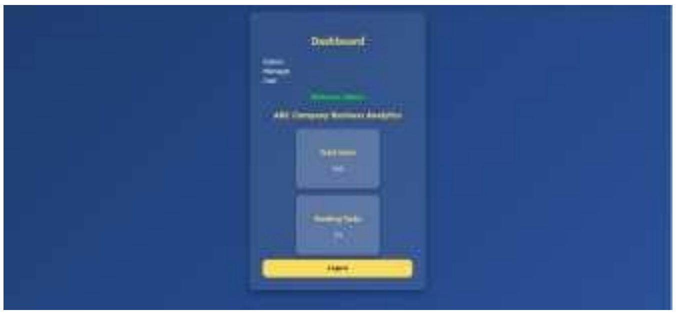
2. Body of Paper

In traditional authentication systems, user credentials, roles, and permissions are stored in centralized databases. While widely adopted, this model is prone to critical vulnerabilities such as unauthorized access, privilege escalation, insider attacks, and data breaches. Malicious users can exploit insecure backend logic or improperly managed access controls to gain elevated privileges and perform unauthorized actions. Furthermore, centralized Role-Based Access Control (RBAC) systems are susceptible to tampering, as administrators or attackers with elevated access can modify roles or permissions, compromising system security.

Our project, Secure Auth, introduces a blockchain-based RBAC system to prevent such attacks. By leveraging Ethereum smart contracts, user roles and permissions are stored in a tamper-proof and decentralized manner. Authentication is done via MetaMask, which uses cryptographic signatures instead of passwords. This not only eliminates password-based vulnerabilities like phishing and brute-force attacks but also ensures that control is enforced strictly through immutable smart contract logic.

Fig -1:Front-end





Furthermore, Secure Auth enhances data security by integrating IPFS (Inter Planetary File System) to store encrypted user data off-chain. Only the rightful user can decrypt and access their data, ensuring privacy even if external databases are compromised. This combination of blockchain, smart contracts, and decentralized storage offers a transparent, secure, and scalable solution to protect against unauthorized access and privilege escalation.

Fig -2: Role-based Access

Fig -3:Database.

3. CONCLUSIONS

Our project prevents unauthorized access and privilege escalation by using blockchain-based Role-Based Access Control (RBAC). User roles are securely stored on the Ethereum blockchain, and smart contracts enforce access rules, ensuring tamper-proof and decentralized management. By integrating MetaMask authentication, this system enhances security, transparency, and trust, making access control more reliable and resistant to attacks.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our project guide and faculty members for their invaluable guidance, support, and encouragement throughout the development of this research work. Their expertise and feedback were instrumental in shaping the direction and quality of this project.
We also extend our thanks to our institution for providing the necessary resources, facilities, and a conducive environment for research and innovation. Finally, we are grateful to our peers, friends, and family members who supported and motivated us during this journey.
Let me know if you'd like to mention any specific professor, lab, or department!

REFERENCES

1. Khan, M., & Verma, P. (2020). "Enhancing Security in Node.js RESTful APIs using JSON Web Tokens (JWT)." International Journal of Innovative Research in Technology, 7(3), 66-70.
2. Brad traversy (2021). MERN Stack Front To Back: Full Stack React, Redux & Node.js. Independently published. – A practical guide to building full-stack applications using the MERN stack, including authentication and role-based access.
3. Michael Hartl. (2022). Node.js and Express: Build Full-Stack Web Apps with JavaScript. Learn Enough. – Covers building secure backend systems with Node.js and Express, including login and access control.
4. Choudhary, A., & Shah, M. (2022). "Secure Login and Registration System using React, Express and MongoDB." International Journal of Information and Computation Technology, 12(3), 105-110.