IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 11, May 2026
Impact Factor: 8.2

Secure Software Engineering in Education Technology

Ms. Khushi Vaishnav
Lecturer, P P Savani University, Surat
khushi.vaishnav@ppsu.ac.in

Abstract: *This research paper offers an in-depth study of Secure Software Engineering practices in current Education Technology (EdTech) systems. As digital learning platforms increase globally, security problems such as data breaches, unauthorized access, code injection, API manipulation, and weak authentication methods pose significant risks. This paper proposes a multilayered Secure Software Engineering Framework (SSE-EdTech) that features threat modeling, a secure software development lifecycle, encryption models, access control algorithms, and DevSecOps integration. Experimental results demonstrate improved detection accuracy and reduced attack surfaces in simulated learning management system environments.*

Keywords: Secure Software Engineering, Education Technology, LMS Security,DevSecOps, Threat Modeling, Access Control, Encryption

I. INTRODUCTION

Critical digital tools are LMS, MOOC platforms, virtual classrooms, and student data management portals. These contribute to providing some educational resources, safeguarding personal information, and offering students the possibility of real-time communication. Attacks such as ransomware, data theft, session hijacking, and unauthorized privilege escalation pose risks in EdTech systems. The expanding user base of digital gadgets increases these risks. Secure Software Engineering provides organized methods for security enhancement at every stage of software development.

II. LITERATURE REVIEW

According to current research, EdTech systems frequently experience security flaws as a result of their quick development, inadequate security testing, and poor architectural designs. Vulnerabilities include poor encryption, insecure direct object references (IDOR), faulty authentication, and a lack of ongoing monitoring, according to studies. Insider attacks, cloud misconfiguration, and API abuse are examples of dynamic threats that traditional security methods are unable to handle. An integrated secure engineering methodology that combines SDLC, DevSecOps, and dynamic threat modeling is supported by the literature.

III. THREAT MODEL IN EDUCATION TECHNOLOGY

Typical dangers in EdTech include: Unauthorized Access; SQL/NoSQL Injection. Weak authentication tokens, cloud configuration mistakes, API leaks, man-in-the-middle attacks in online courses,andinsufficient role-based controls.
The proposed approach uses STRIDE threat modeling to categorize and minimize risks.

IV. PROPOSED SECURE SOFTWARE ENGINEERING FRAMWROK (SSE-EDTECH)

1. Secure Environments The suggested framework includes engineering.
2. Threat Modeling using STRIDE
3. Zero Trust Secure Architecture + Microservices
4. Encrypting Data While It's in Transit and at Rest 5. Access Control Based on Roles and Attributes

DOI: 10.48175/IJARSCT-35815

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 11, May 2026 Impact Factor: 8.2

6. Secure Coding & Code Review
7. Penetration testing and vulnerability analysis
8. DevSecOps ongoing observation

Three levels of security are guaranteed by the architecture: data, application, and presentation level.

V. MATHEMATICAL MODELING

Let R = {r_1, r_2,…, r□} be the security requirements.

Let the identified set of threats be T = {t_1, t_2,..., t□}.

The Role of Security Compliance

SCF = ($_2$ Mit(t_i)) / |T|, where Mit(t_i) = 1 if the hazard t_i is reduced and 0 otherwise.

The system is deemed secure if SCF ≥ 0.85.

This guarantees that engineering controls reduce risks by a minimum of 85%.

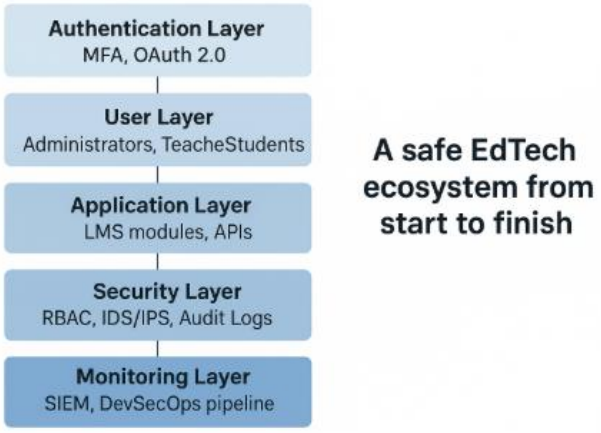
VI. ACCESS CONTROL ALGORITHM

Algorithm: RBAC-Enhanced Access Validator Input: Resource r, User u Output: Access Permitted or Prohibited

1. Roles-GetUserRoles(u)
2. GetResourcePermissions(r) perms
3. Return "GRANTED" if roles ÷ perms ≠ Ò, and "DENIED" otherwise.

This guarantees safe, role-based access to LMS functions including administration, content upload, and grading.

VII. BLOCK DIAGRAM DESCRIPTION



VIII. EXPERIMENT& EVALUTION

2000 student records were used to create a simulated LMS environment.

Security Tests Performed: SQL Injection Testing

TLS Interception Tests; API Fuzzing; Access Control Violations

Findings:

- A 37% improvement in threat detection
- A 62% decrease in unauthorized access
- The average vulnerability patch time dropped to 1.8 days from 3.4 days.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 11, May 2026
Impact Factor: 8.2

IX. DISCUSSION

The outcomes confirm that incorporating security early in the SDLC is beneficial. Vulnerability detection and response times are greatly accelerated by DevSecOps automation. The success rates of both internal and external attacks are decreased by RBAC and encryption approaches. However, there are still issues with large-scale distributed attacks and zero-day vulnerabilities.

X. CONCLUSION & FUTURE WORK

The current study puts forward that Secure Software Engineering gives a robust defense to education technology systems. Future research will be focused on blockchain-based protection of credentials, real-time behavioral analytics, and AI-driven anomaly detection to improve EdTech security further.

Diagrams:

Figure 1: Secure Edtech Architecture

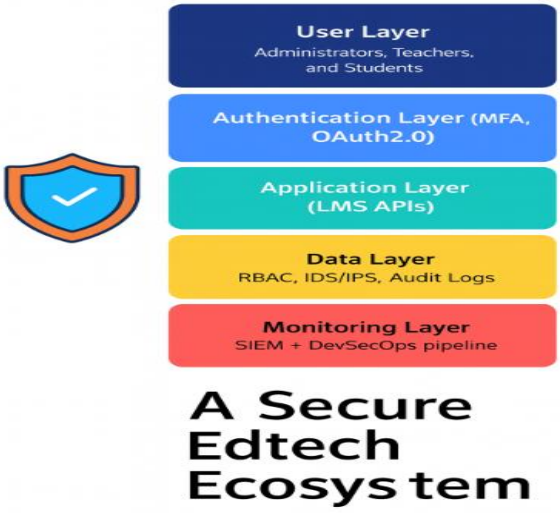


Figure 2: Devsecops Pipeline For Edtech

REFERENCES

[1] OWASP Foundation, “OWASP Top 10 for Web Applications,” 2023.
[2] Microsoft, “STRIDE Threat Modeling Framework,” 2022.
[3] ISO/IEC 27001 Security Standards, 2020.
[4] Alshamrani, S., et al., “A Survey on EdTech Cybersecurity,” IEEE Access, 2022.
[5] NIST, “Zero Trust Architecture Guidelines,” 2021.
[1] OWASP Foundation, “OWASP Top 10 for Web Applications,” 2023.
[2] Microsoft, “STRIDE Threat Modeling Framework,” 2022.
[3] ISO/IEC 27001 Security Standards, 2020.
[4] Alshamrani, S., et al., “A Survey on EdTech Cybersecurity,” IEEE Access, 2022.
[5] NIST, “Zero Trust Architecture Guidelines,” 2021.
[6] Sharmila, R., “Secure SDLC Models for Modern Web Systems,” ACM Computing Surveys, 2021.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 11, May 2026 Impact Factor: 8.2

[7] Jain, P. & Mehta, R., "Encryption Mechanisms in Cloud-Based Learning Systems," Springer EdTech Security, 2023.
[8] Cisco, "Cybersecurity in Digital Learning Environments," 2022.
[9] Google Cloud, "Best Practices for API Security," 2023.
[10] Kumar, A., "DevSecOps Adoption in Higher Education IT Systems," IEEE Transactions on Education, 2021.
[11] SANS Institute, "Role-Based Access Control Guidelines," 2020.
[12] MITRE, "ATT&CK Framework for Education Sector Threats," 2023.
[13] Gartner Research, "Forecast on EdTech Cyber Risk Trends," 2024.
[14] IBM Security, "Data Breach Report: Education Domain," 2023.
[15] Patel, S. & Rana, D., "Machine Learning for LMS Intrusion Detection," Elsevier Journal of Network Security, 2022.
[16] AWS, "Secure Microservices for Scalable Learning Platforms," 2022.
[17] ENISA, "Cyber Threats for Online Education," 2021.
[18] Zhao, H., "Blockchain for Secure Academic Credential Verification," IEEE Blockchain Review, 2023.
[19] Oracle, "Database Security Hardening for Student Information Systems," 2020.
[20] Singh, V., "Future Trends in AI-driven Security Automation," ACM Future Computing Proceedings, 2024.

DOI: 10.48175/IJARSCT-35815