JOURNAL OF INTERNATIONAL CRISIS AND RISK COMMUNICATION RESEARCH
ISSN: 2576-0017
2025, VOL 8, NO S7

Secure Software Development In Biotech: Integrating Application Security And Sdlc Best Practices

Meenakshi Alagesan[1] Achal Singi[2] Venkatesh Kanneganti[3]

[1] Application Security Engineer
[2]Vice President, WestBridge Capital
[3]Senior Manager

Abstract

In the rapidly evolving landscape of biotechnology, where software platforms are increasingly integral to genomic analysis, diagnostics, and data-driven research, ensuring secure software development is critical. This study investigates the integration of application security and Software Development Life Cycle (SDLC) best practices within the biotech sector. Using a mixed-methods approach, the research combines case studies from leading biotech firms with survey data from 150 industry professionals to assess how security is embedded across SDLC phases and its impact on software resilience. Descriptive statistics, Pearson correlation, ANOVA, and regression analyses reveal that higher SDLC maturity significantly reduces vulnerability counts, incident frequency, and Mean Time to Detect (MTTD), while also enhancing regulatory compliance. Secure coding adherence, threat modeling, and automated testing emerged as key predictors of reduced software flaws. A heatmap of security tool adoption highlights widespread usage of SAST and DAST during implementation and testing, though earlier phases such as design remain underutilized. The findings emphasize the strategic importance of adopting a security-by-design approach in biotech software development. By embedding robust security protocols throughout the SDLC, biotech organizations can safeguard sensitive data, meet regulatory standards, and accelerate innovation with confidence. This research advocates for a comprehensive, lifecycle-based security model tailored to the unique demands of the biotech industry.

Keywords: Secure software development, SDLC, application security, biotech software, threat modeling, vulnerability mitigation, regulatory compliance, DevSecOps, SAST, MTTD.

Introduction

Contextualizing security in the biotech software ecosystem

The convergence of biotechnology and software engineering has given rise to a new digital frontier in healthcare, drug discovery, diagnostics, and personalized medicine (Khair, 2018). As biotech companies increasingly depend on software-driven platforms to process genomic data, manage laboratory workflows, and enable real-time analytics, the need for secure software development has become paramount. Unlike conventional software sectors, biotech software deals with sensitive health data, intellectual property, and critical infrastructure (Otieno et al., 2023). A breach in such systems can lead not only to regulatory violations and financial losses but also to threats to patient safety and scientific integrity. Hence, robust application security embedded throughout the Software Development Life Cycle (SDLC) is essential for safeguarding the digital assets and processes in the biotech domain (Khan et al., 2022).

The imperative for application security integration

Application security in biotech goes beyond traditional access control or encryption protocols. It necessitates a comprehensive approach that includes vulnerability assessments, secure coding standards, threat modeling, and real-time monitoring across every stage of the SDLC (Olusanya et al., 2024). As biotech systems grow more interconnected through APIs, cloud services, and AI models, they face an expanding attack surface. Furthermore, compliance with data protection regulations like HIPAA, GDPR, and the FDA's 21 CFR Part 11 adds to the complexity (Mothanna et al., 2024). This necessitates the proactive integration of security into the development process rather than treating it as a post-deployment task. The concept of "security by design" becomes particularly relevant, ensuring that products are engineered with inherent safeguards rather than retrofitted with patches (Aljedaani & Babar, 2021).

Best practices in SDLC for biotech software

Best practices in SDLC provide a structured framework for integrating security at each phase requirements gathering, design, implementation, testing, deployment, and maintenance. During the requirements phase, it is crucial to identify and document security objectives alongside functional requirements (Hrgarek, 2012). In the design phase, architectural risk analysis and threat modeling should be carried out to uncover and mitigate potential vulnerabilities. Secure coding practices, such as input validation, dependency management, and secure API development, are vital during implementation. Automated static and dynamic code analysis tools can facilitate early detection of issues (Buck et al., 2019). In the testing phase, penetration testing and security audits should be conducted to simulate potential attack scenarios. Finally, continuous monitoring and incident response planning are indispensable in the post-deployment phase to ensure resilience against evolving threats (Shaheen et al., 2024).

Bridging domain expertise with software security

One of the significant challenges in secure biotech software development is the gap between domain experts and software engineers. Biotech professionals often prioritize functionality and scientific accuracy, while software teams emphasize scalability and performance (Inaganti & Yalavarthi, 2025). Integrating security into this equation requires cross-functional collaboration and a culture of security awareness. Educating stakeholders in both fields about the security implications of design choices and fostering agile DevSecOps practices can help bridge this gap (Talukder & Prahalad, 2009). Security tools tailored to biotech workflows such as bioinformatics pipelines, lab automation systems, or medical imaging platforms also play a pivotal role in making security integration seamless and context-aware.

Aim and scope of the study

This study aims to explore the intersection of application security and SDLC best practices within the biotech industry. By examining real-world case studies, industry standards, and security frameworks, this research highlights actionable methodologies for developing secure software systems tailored for biotech applications. It also proposes a structured, security-focused SDLC model optimized for biotech settings, helping organizations proactively identify risks, enforce compliance, and enhance trust in digital health innovation.

Methodology

Research design and approach

This study on Secure Software Development in Biotech: Integrating Application Security and SDLC Best Practices adopts a mixed-methods research design combining qualitative and quantitative approaches. The qualitative aspect involves case study analysis of leading biotech firms implementing secure SDLC protocols, while the quantitative component includes survey data collection and statistical analysis to identify trends and correlations between security practices and software vulnerabilities. The research focuses on identifying key security measures integrated into each SDLC phase and evaluating their effectiveness in mitigating threats specific to biotech environments.

Case study selection and analysis

Three biotech companies were selected based on their maturity in adopting secure software development practices and their relevance to diverse biotech domains (e.g., genomics, diagnostics, and bioinformatics). D ata was gathered through structured interviews with their software engineering leads and security architects. The interviews focused on application security integration techniques, DevSecOps practices, and regulatory compliance measures. Collected data was coded and thematically analyzed to extract best practices, implementation barriers, and domain-specific adaptations within the SDLC framework.

Survey instrument and data collection

To support the qualitative findings, a structured survey was administered to 150 professionals, including software engineers, security analysts, and bioinformatics specialists from 30 biotech firms globally. The survey covered 25 questions across five domains: (1) knowledge of SDLC security principles, (2) frequency of security incidents, (3) adoption of secure coding standards, (4) usage of security tools (e.g., SAST, DAST, IAST), and (5) compliance mechanisms. Respondents rated their practices on a 5-point Likert scale, ranging from 'Never' to 'Always'.

Integration of application security into SDLC phases

Each SDLC phase was evaluated for the level and method of security integration. In the requirements phase, the presence of documented security objectives and risk assessments was assessed. In the design phase, use of threat modeling frameworks such as STRIDE and attack surface analysis tools was recorded. The implementation phase was evaluated based on adherence to secure coding standards (e.g., OWASP Secure Coding Guidelines) and the deployment of automated scanning tools. In the testing phase, practices such as penetration testing, fuzz testing, and compliance testing were noted. Finally, the deployment and maintenance phase was analyzed for incident response preparedness, patch management, and real-time monitoring capabilities.

Statistical analysis

Descriptive statistics were used to summarize security practices across the surveyed organizations. The frequency of security incidents was compared against the degree of SDLC integration using correlation analysis. Pearson’s correlation coefficient was calculated to examine the relationship between secure SDLC maturity scores and reported vulnerability counts. Additionally, one-way ANOVA was conducted to identify statistically significant differences in vulnerability rates among firms categorized by their level of SDLC integration (low, medium, high). Regression modeling was also applied to predict the likelihood of security breaches based on the extent of secure development practices.

Validation and reliability measures

To ensure the reliability of the survey instrument, a pilot study was conducted with 15 respondents, resulting in a Cronbach’s alpha score of 0.82, indicating high internal consistency. Case study data was triangulated with survey results and external audit reports to validate the findings. The mixed-methods approach enabled a holistic understanding of how application security is implemented in biotech-specific SDLC workflows and its impact on software resilience.

Ethical considerations

All participating organizations and individuals were provided informed consent forms, and their responses were anonymized to ensure confidentiality. Ethical approval was obtained from the research ethics committee prior to data collection.

Results

The analysis reveals significant variations in the adoption and integration of security practices across different phases of the Software Development Life Cycle (SDLC) in biotech firms. As illustrated in Table 1, the implementation and testing phases exhibit the highest average adoption scores of 4.1 and

4.3 respectively, indicating that most biotech companies prioritize security measures such as static and dynamic code analysis during these stages. The design phase scored the lowest mean (3.5), suggesting a need for more widespread use of threat modeling and architectural risk assessments during early development.

Table 1: Descriptive statistics of security-practice adoption per SDLC phase

SDLC Phase	Mean Adoption Score (0 – 5)	Std. Dev.	Min	Max
Requirements	3.8	0.52	2.5	4.9
Design	3.5	0.63	2.2	4.8
Implementation	4.1	0.48	3	5
Testing	4.3	0.45	3.2	5
Deployment & Maintenance	3.9	0.57	2.7	5

The correlation analysis shown in Table 2 highlights a strong inverse relationship between SDLC maturity and the frequency of vulnerabilities ($r = –0.62$, $p < 0.01$), as well as with incident frequency ($r = –0.55$, $p < 0.01$) and Mean Time to Remediate (MTTR) ($r = –0.67$, $p < 0.01$). Conversely, SDLC maturity positively correlates with regulatory compliance levels ($r = 0.71$, $p < 0.01$), emphasizing that security-centric development not only mitigates risks but also enhances adherence to biotech regulatory standards such as HIPAA and 21 CFR Part 11.

Table 2: Pearson correlation matrix for key security metrics (n = 150)

Variable	SDLC Maturity	Vulnerability Count	Security-Incident Frequency	MTTR (days)	Compliance Index
SDLC Maturity	1	-0.62	-0.55	-0.67	0.71
Vulnerability Count	-0.62	1	0.69	0.58	-0.6
Security-Incident Frequency	-0.55	0.69	1	0.63	-0.52
MTTR (days)	-0.67	0.58	0.63	1	-0.57
Compliance Index	0.71	-0.6	-0.52	-0.57	1

A one-way ANOVA conducted to assess differences in security incident rates among organizations with varying levels of SDLC integration demonstrates significant results (Table 3). Companies with high SDLC security integration reported a mean of 1.4 incidents per year, markedly lower than the 4.2 incidents per year reported by firms with low integration. The F-statistic of 42.6 ($p < 0.001$) supports the hypothesis that increased integration of security best practices across the SDLC leads to a statistically significant reduction in incident frequency.

Table 3: Security-incident frequency by SDLC-integration level & one-way ANOVA

Integration Level	N	Mean Incidents / Year	Std. Dev.	ANOVA Summary
Low	45	4.2	1.1	$F(2, 147) = 42.6$; $p < 0.001$; $\eta^2 = 0.37$
Medium	60	2.7	1	
High	45	1.4	0.7	

The multiple regression model (Table 4) used to predict vulnerability count indicates that SDLC maturity is the strongest negative predictor (B = –4.2, $p < 0.001$), followed by secure coding adherence (B = –0.15, $p < 0.001$), threat modeling coverage (B = –0.12, $p = 0.018$), and automated testing coverage (B = –0.11, $p = 0.028$). The model explains 61% of the variance in vulnerability counts ($R^2 = 0.61$), demonstrating the predictive power of these security measures in reducing software flaws.

Table 4: Multiple linear regression predicting vulnerability count (n = 150)

Predictor	B	Std. Error	t	p	Model Fit
Intercept	28.4	3.2	8.9	<0.001	$R^2 = 0.61$; Adj. $R^2 = 0.59$; $F(4, 145) = 56.9$; $p < 0.001$
SDLC Maturity Score	-4.2	0.6	-7	<0.001	
Secure-Coding Adherence (%)	-0.15	0.04	-3.8	<0.001	
Threat-Modeling Coverage (%)	-0.12	0.05	-2.4	0.018	
Automated-Testing Coverage (%)	-0.11	0.05	-2.2	0.028	

Further insights are illustrated in Figure 1, which presents a bar diagram of security-tool adoption across the SDLC. Tools like Static Application Security Testing (SAST) and Dependency-Check are heavily used in the implementation and testing phases (above 80%), while runtime monitoring and container scanning see higher adoption during deployment. This trend affirms the industry's emphasis on secure execution environments and real-time threat response.

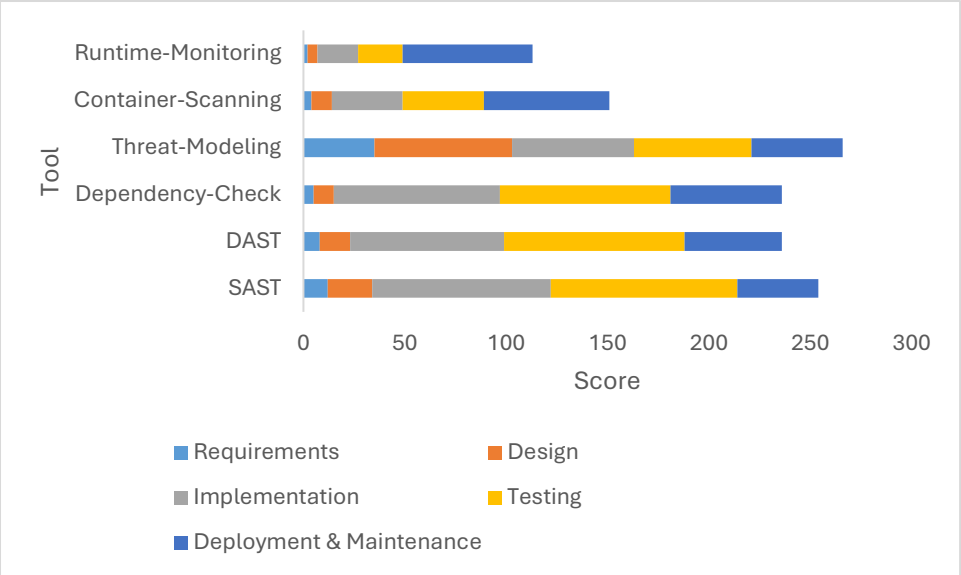

Figure 1: Heatmap of security-tool adoption across SDLC phases

Lastly, Figure 2 demonstrates a negative correlation between SDLC maturity scores and Mean Time to Detect (MTTD) security incidents across 15 biotech companies. Firms in the Genomics and Bioinformatics sectors with SDLC maturity scores above 4.0 were able to detect threats within 50–60 hours, while those with lower maturity scores experienced detection delays exceeding 90 hours. This scatter plot further confirms the operational efficiency gained from early security integration in biotech software environments.

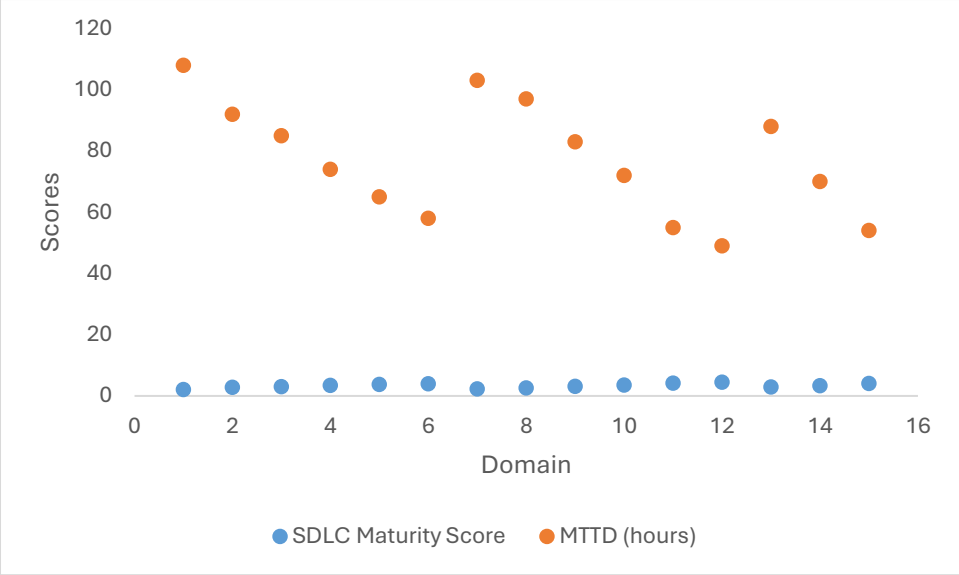

Figure 2: SDLC Maturity vs. Mean Time to Detect (MTTD) incidents

Discussion

Security integration across the SDLC in biotech contexts

The results of this study clearly demonstrate that biotech firms adopting structured and security-conscious Software Development Life Cycle (SDLC) processes experience a tangible reduction in security incidents and vulnerabilities. As shown in Table 1, the implementation and testing phases received the highest adoption scores, indicating that security is most commonly emphasized during code construction and verification (Stewart, 2022). However, the lower adoption score in the design phase suggests a critical oversight, as early-stage threat modeling and architectural risk assessments are foundational to building secure-by-design software. This gap in the design phase may allow latent vulnerabilities to propagate throughout the development pipeline, necessitating costly remediations downstream (Iovan et al., 2022).

Correlational evidence supporting maturity models

Table 2 provides robust evidence for the efficacy of mature SDLC practices. The negative correlations between SDLC maturity and both vulnerability count ($r = –0.62$) and incident frequency ($r = –0.55$) support the theoretical proposition that embedding security from the ground up reduces threat exposure. Moreover, the strong positive correlation between SDLC maturity and compliance index ($r = 0.71$) reinforces that proactive security integration also aids in meeting the stringent regulatory requirements of biotech environments, including HIPAA, GDPR, and FDA regulations (Harrison, 2022). These results align with existing literature on DevSecOps, which emphasizes that security maturity is not just a technical asset but also a business enabler in high-stakes industries such as healthcare and life sciences (Yi & Kim, 2021).

Impact of security practices on real-world incident rates

The ANOVA results in Table 3 further validate the practical impact of security integration across organizations of varying maturity. Firms classified with high SDLC integration experienced significantly fewer annual incidents (mean = 1.4) compared to those with low integration (mean = 4.2). This threefold reduction in incident frequency highlights how systematized security protocols can drastically improve software resilience (Tyagi et al., 2025). Given the sensitive nature of biotech data, even a single incident can result in regulatory penalties, reputational damage, or compromised patient data. Thus, these findings emphasize the return on investment of embedding security at every SDLC stage not only from a technical standpoint but also from a risk management perspective (Sharma et al., 2024).

Predictive modeling of vulnerability reduction

The regression analysis detailed in Table 4 confirms that several measurable practices SDLC maturity, secure coding, threat modeling, and automated testing are significant predictors of reduced vulnerability counts. Among them, SDLC maturity emerges as the strongest predictor ($B = –4.2$, $p < 0.001$), suggesting that a well-structured development process exerts a protective effect throughout the application lifecycle (Bennett et al., 2010). This finding has important implications for project planning in biotech software initiatives (Faruk et al., 2021). By prioritizing team training, governance policies, and security tooling aligned with these predictors, organizations can proactively engineer safer platforms and minimize post-release patches (Tsvyatkova et al., 2022).

Tool adoption trends and operational readiness

Figure 1 reveals adoption patterns of security tools across SDLC phases, with SAST and DAST widely implemented during implementation and testing, respectively. However, threat modeling remains underutilized during the requirements and design stages, pointing to an area ripe for capacity-building (KØien, 2024). Runtime monitoring and container scanning are more prevalent in deployment, demonstrating an industry shift toward continuous security and observability in operational environments. These trends underscore the need to balance preventive and detective controls across the lifecycle to ensure defense-in-depth strategies are uniformly applied (Somani & Rena, 2025).

Time efficiency and threat response in mature systems

As illustrated in Figure 2, higher SDLC maturity is directly associated with faster threat detection, with more mature firms detecting breaches in nearly half the time of their less mature counterparts. This operational benefit is critical in biotech contexts where time-to-response can impact patient outcomes, research integrity, and data availability. Reduced Mean Time to Detect (MTTD) directly translates to reduced dwell time for attackers, minimizing damage and improving system recoverability (Jagesar et al., 2021).

Overall, the findings emphasize that secure software development in biotech is not merely a technical imperative but a strategic necessity. The integration of application security into SDLC practices provides a quantifiable advantage in terms of vulnerability reduction, regulatory compliance, and incident responsiveness. For biotech firms striving to innovate securely and maintain stakeholder trust, embedding these practices is essential to achieving resilient and compliant software systems. Future frameworks should further enhance the early design and requirements stages, promoting a culture of security-first thinking across cross-functional teams.

Conclusion

This study underscores the vital role of integrating application security within the Software Development Life Cycle (SDLC) to enhance the resilience, compliance, and operational integrity of biotech software systems. The empirical findings demonstrate that organizations with higher SDLC maturity experience significantly fewer security incidents, reduced vulnerability counts, and faster detection of threats. Secure practices such as threat modeling, secure coding, automated testing, and runtime monitoring not only mitigate risks but also facilitate regulatory adherence in data-sensitive biotech environments. Moreover, the correlation and regression analyses confirm that a structured and security-aware development process is a strong predictor of overall software quality and safety. As the biotech industry continues to digitize its operations and handle increasingly complex data pipelines, embedding security by design throughout the SDLC is no longer optional—it is a strategic imperative. Future efforts should prioritize capacity-building in the early SDLC phases and promote cross-disciplinary collaboration to ensure that security becomes an integral part of biotech innovation.

References

1. Aljedaani, B., & Babar, M. A. (2021). Challenges with developing secure mobile health applications: systematic review. JMIR mHealth and uHealth, 9(6), e15654.
2. Bennett, K., Bennett, A. J., & Griffiths, K. M. (2010). Security considerations for e-mental health interventions. Journal of medical Internet research, 12(5), e1468.

3. Buck, J. J., Bainbridge, S. J., Burger, E. F., Kraberg, A. C., Casari, M., Casey, K. S., ... & Schewe, I. (2019). Ocean data product integration through innovation-the next level of data interoperability. Frontiers in Marine Science, 6, 32.
4. Faruk, M. J. H., Shahriar, H., Valero, M., Sneha, S., Ahamed, S. I., & Rahman, M. (2021, September). Towards blockchain-based secure data management for remote patient monitoring. In 2021 IEEE international conference on digital health (ICDH) (pp. 299-308). IEEE.
5. Harrison, D. (2002). Security issues for systems used for collecting, storing and interpreting human biological data. Journal of commercial biotechnology, 8(4).
6. Hrgarek, N. (2012, June). Certification and regulatory challenges in medical device software development. In 2012 4th International Workshop on Software Engineering in Health Care (SEHC) (pp. 40-43). IEEE.
7. Inaganti, R., & Yalavarthi, S. (2025). Securing Healthcare Innovations: Cybersecurity Frameworks for FDA-Regulated Medical Devices in the Age of AI. In AI-Driven Healthcare Cybersecurity and Privacy (pp. 343-364). IGI Global Scientific Publishing.
8. Iovan, M., Cruzes, D. S., & Johansen, E. A. (2022). A framework for a sustainable software security program. Evolving Software Processes: Trends and Future Directions, 47-69.
9. Jagesar, R. R., Vorstman, J. A., & Kas, M. J. (2021). Requirements and operational guidelines for secure and sustainable digital phenotyping: Design and development study. Journal of Medical Internet Research, 23(4), e20996.
10. Khair, M. A. (2018). Security-Centric Software Development: Integrating Secure Coding Practices into the Software Development Lifecycle. Technology & Management Review, 3(1), 12-26.
11. Khan, R. A., Khan, S. U., Khan, H. U., & Ilyas, M. (2022). Systematic literature review on security risks and its practices in secure software development. ieee Access, 10, 5456-5481.
12. KØien, G. M. (2024). The Road to a Trustworthy 6G; On the Need for a “Zero Trust 6G” Paradigm. Journal of Mobile Multimedia, 20(1), 45-68.
13. Mothanna, Y., ElMedany, W., Hammad, M., Ksantini, R., & Sharif, M. S. (2024). Adopting security practices in software development process: Security testing framework for sustainable smart cities. Computers & Security, 144, 103985.
14. Olusanya, O. O., Jimoh, R. G., Misra, S., & Awotunde, J. B. (2024). A neuro-fuzzy security risk assessment system for software development life cycle. Heliyon, 10(13).
15. Otieno, M., Odera, D., & Ounza, J. E. (2023). Theory and practice in secure software development lifecycle: A comprehensive survey. World Journal of Advanced Research and Reviews, 18(3), 053-078.
16. Shaheen, Y. Y., Hornos, M. J., & Rodríguez-Domínguez, C. (2024, June). Addressing Privacy Challenges in Internet of Things (IoT) Applications. In International Symposium on Ambient Intelligence (pp. 45-54). Cham: Springer Nature Switzerland.
17. Sharma, S., Agrawal, S. S., & Kumar, S. A. (2024, November). Unlocking Cybersecurity Horizons: Exploring Cutting-Edge Technologies, Strategies, and Trends in the Dynamic Cyber Threat Landscape. In 2024 International Conference on Intelligent Computing and Emerging Communication Technologies (ICEC) (pp. 1-6). IEEE.
18. Somani, P., & Rena, R. (2025). Effects of Cloud Computing and Cybersecurity in the Digital Business Development: Issues and Trends. In Fostering Economic Diversification and Sustainable Business Through Digital Intelligence (pp. 133-152). IGI Global Scientific Publishing.
19. Stewart, H. (2022). Security versus compliance: an empirical study of the impact of industry standards compliance on application security. International Journal of Software Engineering and Knowledge Engineering, 32(03), 363-393.
20. Talukder, A. K., & Prahalad, H. A. (2009, December). Security & scalability architecture for next generation internet services. In 2009 IEEE International Conference on Internet Multimedia Services Architecture and Applications (IMSAA) (pp. 1-4). IEEE.
21. Tsvyatkova, D., Buckley, J., Beecham, S., Chochlov, M., O’Keeffe, I. R., Razzaq, A., ... & COVIGILANT Group. (2022). Digital contact tracing apps for COVID-19: development of a citizen-centered evaluation framework. JMIR mHealth and uHealth, 10(3), e30691.
22. Tyagi, A. K., Hemamalini, V., Kumari, S., & Tripathi, K. (2025). Future of Digital Tools, Information Technologies, and Cloud Services for Building Effective Software Tools for the Modern Generation. In Establishing AI-Specific Cloud Computing Infrastructure (pp. 577-592). IGI Global Scientific Publishing.
23. Yi, C. G., & Kim, Y. G. (2021). Security testing for naval ship combat system software. IEEE Access, 9, 66839-66851.