УДК 004.056.5

CLOW-CODE PLATFORMS: DEMOCRATIZING DEVELOPMENT OR CREATING A DELAYED DANGER

Chaimae Michich, Kadatska Olha
e-mail: olha.kadatska@nure.ua
Kharkiv National University of Radio Electronics,
V.V. Popovskyy dep. ICE,
Kharkiv, Ukraine

Low-code platforms have revolutionized software development by enabling non-technical users to build applications. However, their ease of use often bypasses traditional security safeguards, creating systemic vulnerabilities. This article analyzes security risks in low-code ecosystems, including shadow IT, insecure integrations, and compliance gaps. Using real-world case studies, we propose mitigation strategies such as centralized governance and DevSecOps integration to balance agility with security.

Low-code platforms like Microsoft Power Apps and OutSystems empower "citizen developers" to create enterprise-grade applications without coding expertise. By 2025, 70% of new apps will use low-code tools [1]. However, 60% of these apps fail basic OWASP security checks due to misconfigurations and poor access controls [2]. This article examines the dual role of low-code platforms as innovation accelerators and security liabilities, focusing on theoretical models and practical mitigation strategies.

Low-code platforms abstract backend complexity through visual interfaces and prebuilt components. Security implementations vary across two models:

1) Native security: built-in authentication (e.g., Power Apps' Azure AD integration), limited encryption (data-at-rest often unencrypted by default) and basic RBAC tied to platform user roles.

2) External security tools: API gateways (e.g., Apigee) for threat detection and third-party secrets managers (e.g., Hashi Corp Vault).

Shadow IT proliferation. Shadow IT proliferation is a major challenge in low-code development, as illustrated by a scenario where a hospital nurse built a patient scheduling app using Power Apps. The app stored unencrypted protected health information (PHI) in a public SharePoint folder, leading to the leakage of 12000 records before IT discovered the issue. This incident underscores a broader trend - 83% of low-code apps are deployed without security reviews, leaving organizations vulnerable to data breaches [3].

Insecure third-party integrations also pose significant risks. For example, a retail company's inventory app integrated an unvetted API, which exposed AWS credentials [4]. This highlights a widespread problem, as 67% of low-code apps rely on APIs with outdated OAuth scopes, increasing the likelihood of exploitation. Data leakage via embedded logic is another critical

issue. In one case, a financial analyst developed a loan approval app that inadvertently exposed sensitive credit scores due to misconfigured UI components. This vulnerability is exacerbated by the fact that many low-code platforms lack row-level security for databases like Snowflake, making it easier for sensitive data to be accessed unintentionally.

Compliance gaps further compound these challenges. This reflects a systemic issue, as only 29% of low-code platforms automatically enforce data residency rules, leaving organizations at risk of regulatory penalties.

Governance gaps are a significant risk in low-code development, as decentralized development often bypasses IT oversight. To address this, organizations can deploy tools like Microsoft Cloud App Security to catalog and monitor shadow IT apps, ensuring visibility and control over unauthorized applications.

Another critical issue is inadequate access controls, where default "edit-all" permissions for app creators can lead to overprivileged users. Implementing the least privilege access models through identity management solutions like Okta or SailPoint can mitigate this risk by restricting permissions to only what is necessary for each role.

Static secrets in workflows, such as hardcoded API keys in low-code automation tools like Zapier, also pose a serious threat. Integrating secrets managers like AWS Secrets Manager can help securely store and manage sensitive credentials, reducing the risk of exposure, we highlight the importance of governance and education in preventing similar breaches.

In conclusion, low-code platforms democratize development but introduce systemic risks when security is treated as an afterthought. Organizations must adopt a dual strategy to address these challenges. First, centralized governance tools like Mendix Control Center can enforce policies and ensure compliance. Second, embedded security practices, such as shift-left testing with tools like Snyk or Checkmarx, can identify and resolve vulnerabilities early in the development process. Looking ahead, future research should explore AI-powered anomaly detection tailored to low-code environments, offering proactive measures to identify and mitigate risks in real time. By combining governance, education, and advanced tools, organizations can harness the benefits of low-code development while minimizing its inherent risks.

References:

1. Gartner. (2023). Market Guide for Enterprise Low-Code Application Platforms. URL: https://www.gartner.com (accessed 01/12/2025).

2. OWASP. (2023). Low-Code Security Top 10. URL: https://owasp.org (accessed 01/12/2025).

3. HIPAA Journal. (2023). Microsoft Power Apps Data Breach.URL: https://www.hipaajournal.com (accessed 01/12/2025).

4. Darktrace. (2024). Low-Code Cryptojacking Incident Report .URL: