Volume 9, Issue 5, May – 2024 International Journal of Innovative Science and Research Technology
ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24MAY2349

Application Security: The Perspective of a Program Manager

Adetayo Adeyinka

Abstract:- Application security has become increasingly important as organizations digitally transform and rely more on software to operate. However, balancing security with competing development priorities like speed and new features presents ongoing challenges for program managers responsible for overseeing application projects. This study explored the perspectives of 10 cybersecurity program managers through interviews to understand their approaches to security governance and the common obstacles faced. Key challenges included pressuring developers focused on rapid delivery to also consider threats, limited security testing resources, and difficulty prioritizing among risks. However, establishing security requirements early in planning and integrating validation checks directly into workflows helped shift security left. Close collaboration between functions and leadership support for proper training and staffing also aided prioritization. While generalizability was limited, data saturation was reached on major themes. Establishing security guidelines upfront aligned with frameworks, yet deeper cultural changes may still be needed at firms resistant to oversight. Metrics and skills shortages also require attention. The research validated the pivotal role of program managers and provided insights into both barriers and effective practices, with implications for process improvements and leadership support to strengthen application defences.

Keywords:- *Application Security, Software Development, Program Management, Security Governance, Risk Management.*

I. INTRODUCTION

In today's digital landscape, applications have become the primary interface between organizations and customers, serving as a gateway for delivering services, transacting business, and sharing information (Kalakota & Robinson, 2000) (Bilgihan, Kandampully, & Zhang, 2016). Program managers play a pivotal oversight role in application development projects, responsible for planning, coordinating resources, monitoring progress, and addressing risks (Too & Weaver, 2014). Existing research has highlighted common issues around authentication, authorization, input validation, and encryption that can be introduced at various stages of security testing and reviews are not adequately enforced (Viega & Messier, 2003).

While developers and security engineers directly implement controls, the program manager acts as a lynchpin for coordinating cross-functional collaboration and integrating security practices into governance and project management frameworks (Radaelli, Spyridonidis, & Currie, 2024). This study aims to explore how program managers approach security oversight and what challenges they encounter in prioritizing it appropriately.

II. APPLICATION SECURITY TESTING METHODS

There are various methodologies used to test applications for security vulnerabilities, with the main categories being Dynamic Application Security Testing (DAST) and Static Application Security Testing (SAST) (Pan, 2019).

DAST involves running automated scanners against a live application to detect issues. Scanners emulate how attackers and users interact with the app by executing common exploits like injection attacks and credential brute-forcing. (Kennedy, O'gorman, Kearns, & Aharoni, 2011) This exposes any vulnerabilities that can be triggered dynamically during runtime. Some key DAST tools include Acunetix, Burp Suite, and Netsparker now(Invicti) (Häyrynen, 2020). They are useful for testing APIs, catching XSS flaws, identifying weak credentials, and more. However, DAST has limitations in that it cannot detect all possible logic and access control bugs.

SAST analyzes application source code and dependencies statically without executing the program. It looks for flaws like hardcoded passwords, missing input validation, insecure functions/libraries, and other code quality issues that could lead to exploits. (Forte, 2021) SAST tools parse code using techniques like data flow analysis and control flow graphing. Popular SAST platforms contain IDE plugins (e.g. SonarQube), CLI scanners (Bandit), and SaaS offerings (Checkmarx, Veracode) (Nilsson, 2019). They are effective for catching serious defects early but cannot identify all runtime vulnerabilities.

Most secure SDLC frameworks recommend performing both DAST and SAST at multiple stages. For example:

- SAST during development to find and fix bugs upfront
- DAST on pre-production builds to test deployments
- SAST on pull requests for code reviews
- DAST on production periodically to check for new vulnerabilities

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24MAY2349

- SAST after major code changes to prevent regressions

Using complementary DAST+SAST methodically provides comprehensive coverage to strengthen application defenses. Ongoing monitoring also helps shift security left in the development process.

III. LITERATURE REVIEW

Several studies have examined common security issues when development teams do not adequately consider application defenses throughout the software development lifecycle (Howard & Lipner, 2006). The OWASP Top 10 provides a comprehensive overview of the most critical risks, including injection flaws, broken authentication, sensitive data exposure, and more (Aljabri, et al., 2022).

These issues regularly occur due to a lack of security testing at various stages or failure to address earlier vulnerabilities before moving to production. The National Institute of Standards and Technology (NIST) also publishes guidelines for managing security throughout the SDLC to help organizations address such problems systematically (Grance, Hash, & Stevens, 2004).

Research on the causes of vulnerabilities has found that while malicious actors contribute, many defects are inadvertently introduced due to a lack of security expertise or oversight, not malice (Solove & Hartzog, 2022). Developers focused on functionality may overlook threats without the proper training, resources, and processes. Surveys show security expertise and staffing remain a challenge for many firms (Fischer, Fischer, Halibozek, Halibozek, & Walters, 2012) (Warkentin & Willison, 2009).

The Project Management Institute emphasizes the importance of the program manager in planning for security requirements, assessing risks, overseeing reviews, and issue tracking (Kerzner, 2017). Meanwhile, studies of DevOps practices highlight the need for collaboration between security, development, and operations teams, a key part of the program manager's remit (George, 2023).

IV. METHODOLOGY

This study employed a qualitative research design utilizing semi-structured interviews. This approach was chosen to allow for an in-depth examination of experiences and perspectives in the participants' own words (Moore, Dynes, & Chang, 2016). The target population for the research consisted of program managers currently working in the cybersecurity industry.

Participants were recruited through online professional networks like LinkedIn using a screening questionnaire to ensure they met the criteria. The final sample included 10 program managers from various cybersecurity software companies ranging in size from startups to large enterprises. Their levels of experience ranged from 3 to 15 years in program management roles. The 30-45-minute interviews followed a guided protocol but allowed flexibility to explore interesting topics that emerged (Mack & Woodsong, 2005). Sample questions included:

- How do you ensure security requirements are addressed throughout the development lifecycle?
- What barriers have you encountered when trying to enforce security processes and what have you done to overcome them?
- How do you work with development and security teams to integrate their feedback and priorities?

V. RESULTS

Thematic analysis of the interview transcripts revealed several shared challenges experienced by program managers in prioritizing application security (Lorona, 2023). All participants reported having to balance security against competing demands from developers focused on speed and new features. As one manager stated, *"Developers always want to rush ahead to the next thing, it can be a struggle to get them to slow down and consider threats."*

A common barrier cited was a lack of security expertise within development teams. As a program manager at a fintech startup explained, *"Many of our engineers came from non-security backgrounds, so basic things like input validation and authentication were foreign. We had to provide training."* Testing resources were also a limitation according to 40% of interviewees. As one said, *"It's difficult to perform all necessary security testing when you have tight deadlines and limited testers."*

Prioritizing security amid other risks proved challenging as well. As a manager at a cybersecurity software provider commented, *"There are always so many risks to juggle like bugs, outages, new regulations...it's hard to decide which get addressed first sometimes."* Metrics for security performance emerged as an area needing improvement, with 70% expressing difficulties in defining meaningful Key Performance Indicators for security teams.

However, several effective strategies also materialized. Establishing security requirements and governance processes early in the program's planning phases was emphasized, with 90% agreeing this helped secure buy-in and resources. As one explained, *"By the time development starts, security expectations are clear so we avoid rework."*

Integrating security testing routines into existing quality control workflows also helped normalize security. As a biotech company program manager noted, *"Rather than bolt on security later, we embedded checks into our definition of done so it's part of every step."* Close collaboration between

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24MAY2349

functions further supported prioritization, according to 80% of respondents. As one stated, *"When security, development, and operations work as partners, it's much easier to reach consensus."*

VI. DISCUSSION

The results of this study correlate and build upon several themes in existing literature regarding common application security issues and their root causes. For example, the challenges expressed by program managers around balancing developer priorities of speed and agility against security parallels findings that many vulnerabilities are inadvertently introduced when security is not considered early in the SDLC (Grance, Hash, & Stevens, 2004). When developers are solely focused on delivering new features quickly without security training or oversight, defects can easily slip in.

The limited security testing resources cited as an obstacle also mirrors the well-documented industry-wide shortage of cybersecurity skills and staffing reported in workforce surveys (Fischer, Fischer, Halibozek, Halibozek, & Walters, 2012). With security talent unable to meet demand, it is understandable that testing capacity could be constrained within development timelines and budgets. However, this research provides a valuable on-the-ground perspective not extensively captured before on the day-to-day barriers faced.

Some strategies employed by program managers interviewed, such as establishing security requirements upfront and integrating checks directly into workflows, demonstrate efforts to shift security left and align with best practices. Practices like threat modeling applications collaboratively with cross-functional teams also support proactively designing out vulnerabilities, as recommended in frameworks (Grance, Hash, & Stevens, 2004). However, the study also suggests deeper cultural changes may still be needed in organizations where certain development teams resist security reviews as slowing productivity. This underscores the importance of buy-in from leadership to influence mindsets and properly empower the program manager's role.

Additionally, while metrics and adequate testing resources were areas noted for improvement, defining meaningful security KPIs proves inherently difficult. Future research exploring examples of effective metrics could help address this challenge. A key limitation was the modest sample size of 10 program managers, restricting the ability to generalize findings. However, data saturation was reached indicating the major themes were covered. The cybersecurity industry focus also limits broader applicability versus other sectors with potentially different risk postures and maturity levels.

Overall, this research lends empirical support to the significant influence program managers can wield over application defences through governance while highlighting the multifaceted and systemic obstacles confronted. With implications for enhanced collaboration, skills development, and leadership support, it provides insights that could strengthen practices across organizations.

VII. CONCLUSION

This study aimed to understand how program managers approach application security oversight and the challenges they face through interviews with cybersecurity professionals. The research helps validate the important advocacy role of the program manager in governing defences. Yet it also highlights the systemic and cultural changes still needed at some firms to fully empower their responsibilities, such as addressing skills shortages and ensuring security metrics. Leadership support for properly training teams and allocating testing resources appears critical. While generalizability was constrained by the sample size, data saturation was reached on major obstacles. The study adds an applied perspective not extensively captured before to supplement existing literature. It also identifies potential areas for future research, such as defining effective security KPIs.

REFERENCES

[1]. Aljabri, M., Aldossary, M., Al-Homeed, N., Alhetelah, B., Althubiany, M., Alotaibi, O., & Alsaqer, S. (2022). Testing and Exploiting Tools to Improve OWASP Top Ten Security Vulnerabilities Detection. 2022 14th International Conference on Computational Intelligence and Communication Networks (CICN), 797-803.
[2]. Bilgihan, A., Kandampully, J., & Zhang, T. (2016). Towards a unified customer experience in online shopping environments: Antecedents and outcomes. International Journal of Quality and Service Sciences, 8(1), 102-119.
[3]. Fischer, R. J., Fischer, R., Halibozek, E., Halibozek, E. P., & Walters, D. (2012). Introduction to security. Butterworth-Heinemann.
[4]. Forte, V. (2021). Automatic Binary Analysis and Instrumentation of Embedded Firmware for a Control-Flow Integrity Solution.
[5]. George, A. S. (2023). Evolving with the Times: Renaming the IT Department to Attract Top Talent. Partners Universal International Innovation Journal, 1(5), 21-46.
[6]. Grance, T., Hash, J., & Stevens, M. (2004). Security considerations in the information system development life cycle. US Department of Commerce, Technology Administration, National Institute of Standards and Technology.
[7]. Häyrynen, E. (2020). Evaluation of state-of-the-art web application vulnerability scanners.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24MAY2349

[8]. Howard, M., & Lipner, S. (2006). The security development lifecycle (Vol. 8). Redmond: Microsoft Press.
[9]. Kalakota, R., & Robinson, M. (2000). e-Business. Roadmap for Success.
[10]. Kennedy, D., O'gorman, J., Kearns, D., & Aharoni, M. (2011). Metasploit: the penetration tester's guide. No Starch Press.
[11]. Kerzner, H. (2017). Project management: a systems approach to planning, scheduling, and controlling. John Wiley & Sons.
[12]. Lorona, N. (2023). Strategies Employed by Project Managers when Adopting Agile DevSecOps to Manage Software Development in the DoD. Doctoral dissertation, Colorado Technical University.
[13]. Mack, N., & Woodsong, C. (2005). Qualitative research methods.
[14]. Moore, T., Dynes, S., & Chang, F. R. (2016). Identifying how firms manage cybersecurity investment. Workshop on the Economics of Information Security (WEIS), 1-27.
[15]. Nilsson, M. (2019). A Comparative Case Study on Tools for Internal Software Quality Measures.
[16]. Pan, Y. (2019). Interactive application security testing. In 2019 International Conference on Smart Grid and Electrical Automation (ICSGEA), 558-561.
[17]. Radaelli, G., Spyridonidis, D., & Currie, G. (2024). Platform evolution in large inter-organizational collaborative research programs. Journal of Operations Management, 70(1), 22-49.
[18]. Solove, D. J., & Hartzog, W. (2022). Breached!: Why data security law fails and how to improve it. Oxford University Press.
[19]. Too, E. G., & Weaver, P. (2014). The management of project management: A conceptual framework for project governance. International journal of project management, 32(8), 1382-1394.
[20]. Viega, J., & Messier, M. (2003). Secure programming cookbook for C and C++: recipes for cryptography, authentication, input validation & more. O'Reilly Media, Inc.
[21]. Warkentin, M., & Willison, R. (2009). Behavioral and policy issues in information systems security: the insider threat. European Journal of Information Systems, 18(2), 101-105.