Drivers for a Secure Mobile App Development Framework

Christoff Jacobs
Academy of Computer Science and Software Engineering
University of Johannesburg
Johannesburg, South Africa
toffie.cj@gmail.com

Marijke Coetzee
School of Computer Science and Information Systems
North-West University
Potchefstroom, South Africa
marijke.coetzee@nwu.ac.za

***Abstract*—To stay competitive in a fast-evolving landscape, lifestyle, e-commerce, finance, and health sectors use mobile apps to enhance their traditional capabilities with new innovative features to satisfy customer demands. Developing sophisticated and secure mobile apps requires a skilled understanding of software development techniques, protection mechanisms, and mobile security practices to safeguard customers against cyberattacks. Software development teams often apply frameworks and best practices according to their unique experience and knowledge, resulting in vulnerable mobile apps. Numerous software development challenges, a lack of guidelines, and a standardised agile approach for creating secure mobile apps need to be addressed. The primary objective of this research is to initiate the process of delineating mobile application security drivers. This initial step lays the foundation for the subsequent development of a comprehensive, secure software development framework for mobile applications, with an overarching emphasis on security across all stages of development. This framework is designed to be adaptable and customizable to meet the specific security needs of diverse industries. The security drivers can form the foundation for a novel framework to guide the creation of secure mobile apps.**

Keywords-mobile application; secure software development frameworks, cybersecurity.

I. INTRODUCTION

The Covid-19 pandemic's uncertainty caused a sharp rise in the usage of mobile apps, particularly in the banking sector, where branches closed worldwide [1] [2]. Mobile app usage has become commonplace across banking, finance, e-commerce and mHealth industries. The Digital.ai Threat Report for 2023 [3] underscores a notable trend wherein over 50% of mobile applications are subjected to at least one cyberattack. Remarkably, these cyberattacks transcend the boundaries of application popularity, affecting both widely recognized and less popular applications. This prevailing threat landscape further intensifies the urgency surrounding the accelerated delivery of mobile applications. Consequently, companies must deploy more advanced mobile protection mechanisms to mitigate the increased risk posed by a larger attack surface. For example, mechanisms, such as the One-Time-Pin (OTP), biometric trait verification, and liveness assessment to authenticate customers digitally are increasingly becoming standard practice across various industries. Unfortunately, individual software development teams continuously reinvent the wheel as no standard approach exists when defining a new sophisticated security mechanism [4] [5]. In addition to the challenges software developers face in keeping up with evolving security threats, the 2021 Global DevSecOps Survey [6] indicates constant friction between security professionals and software developers. More than three-quarters of security teams believe that software developers find errors and bugs too late in the development process. The challenges for mobile app software developers are that they need specialist security knowledge of mobile apps, application frameworks and operating systems and how they can be compromised.

Software developers do not have adequate guidelines when developing secure mobile apps [6]. Historically development teams apply software development best practices in conjunction with their team's expertise and ad-hoc processes developed over many years. Frameworks such as Open Web Application Security Project (OWASP) [7], National Institute of Standards and Technology (NIST) [8] and MITRE ATTACK [9] and approaches, such as DevSecOps [10] are excellent foundations for supporting security within a mobile app. However, a security gap becomes evident when the inner workings of these frameworks are analysed, as the available techniques, tools, and testing requirements do not fully support the need to develop a secure mobile app [11]. Software development frameworks and best practices support the implementation of technical security aspects to safeguard mobile apps against cyber criminals. However, factors, such as resolving issues and new enhancements, sharing security knowledge between various internal teams and collaborating teams between different companies are not addressed [12], [13]. In principle, the secure development of a mobile app should be supported by a secure software development framework tailored to the security requirements of mobile apps. Current mobile app development processes are ad-hoc in nature and do not provide sufficient guidance for companies and development teams [14]–[16]. Research focuses on developing secure mobile apps using various processes and techniques [14] [17] [18] but does not focus on creating a secure software development framework for mobile apps. This research contributes by identifying mobile app security drivers for a secure software development framework as the first step in this direction.

The following section identifies various actors and their roles in mobile app security to understand the complexities

 ISBN: 978-1-68558-092-6

of the landscape. Section three briefly reviews relevant software development frameworks to understand methods and approaches commonly used to secure a mobile app. Next, nine security drivers are described by analysing secure mobile app development challenges. Section five evaluates the security drivers against common software development frameworks to identify a research gap. Finally, the paper is concluded in section six.

II. MOBILE APP ECOSYSTEM

A mobile app ecosystem is a massive integrated network connecting hardware and various systems that communicate securely. Unfortunately, mobile apps open the door to heightened risk and fraud. For example, 693 banking apps across over 80 countries unveiled 2,157 vulnerabilities [19]. Customers, mobile app stores, mobile apps, mobile network operators, security vendors and technologies, such as firewalls, web servers, core systems and various development and security teams exist within the ecosystem [20]. Actors indicate integration points where security controls are required to guarantee a customer transaction's privacy and integrity. Over and above the user interface, software development and security teams, the risk management, anti-money laundering, digital forensics and financial reporting teams aim to establish a trusting relationship between customers, the company and the mobile app [21]. When a mobile app or a new requirement is developed, software teams do not focus on security requirements and mechanisms as they prefer to leave it to experts [22]. Instead, development teams use security vendors to provide mobile security services, such as security controls and app security testing. Once mobile apps are complete and ready for testing, penetration tests are performed by third parties to validate the behaviour of the integrated security mechanisms and mobile security controls [23]. Then the mobile app is submitted to app stores for review. App store reviews include security tests to check for unsolicited application code injection, stealing personally identifiable information, and keeping customers' mobile data safe from cybercriminals. App store reviewers require credentials to access the mobile app, but they are limited and only authorised to access features associated with the intended review. Customers download and install the mobile app onto their mobile device. A firewall filters the mobile app requests to prevent unsolicited access between these entities. The network operator and internet service providers provide the network channel for the mobile app and company to communicate.

Having provided a comprehensive insight into the intricacies of the mobile app ecosystem, it is essential to recognize that organizations may contemplate alternative implementation strategies to develop a secure mobile application. Nevertheless, it is worth noting that these alternative approaches bring forth their unique risks and challenges, which, while relevant, fall beyond the scope of this research inquiry [24].

The mobile app ecosystem has unique security and software development requirements that a unique secure software development framework should address. Therefore, an analysis of current security frameworks used by mobile app developers and the security requirements of mobile apps is needed to determine a research gap, described next.

III. SECURE SOFTWARE DEVELOPMENT FOR MOBILE APPS

There is a lack of Software Development Life Cycle (SDLC) models and frameworks for mobile application development [25] [26]. Most works focus on the development of the technical components of a mobile app and not the life cycle. Traditional software development methodologies, such as waterfall or agile, have been implemented over the years but do not directly address security. Generally, software development methodologies address the following activities from a high-level point of view: identification of requirements, architecture and design, coding, testing, production and maintenance of the application. When security is added to traditional SDLC phases, it can result in low-quality insecure apps, as software developers do not conform to SDLC phases and lack training and experience in application and security technologies. Software developers decide how and when various guidelines, standards and practices are applied to the different stages of the development life cycle leading to various ad-hoc approaches.

Searching papers on - a secure software development framework for mobile apps - reveals that no such framework exists. In contrast, generalised software development frameworks are an active field of current research with many approaches addressing security [27]. Recent studies recommend that a secure software development lifecycle should address security considerations in each development phase. Industry standards, such as OWASP, ISO/IEC, MITRE ATT&CK knowledgebase and NIST are recommended to be used, but no guidance is provided on how to include security and when to do it [28]. Other frameworks and standards identified by [29], are Common Criteria (CC), Software Assurance Forum for Excellence in Code (SAFECode), Open Group Architecture Framework, TOGAF, ISO/IEC 41062, Payment Card Industry (PCI), National Information Assurance Partnership (NIAP), ioXt alliance and CREST [29]. Exploring NIST and OWASP frameworks has surfaced myriad distinct security and software development requisites. Notable among these include but are not limited to robust Authentication and Authorization mechanisms, Encryption protocols, diligent Threat Modeling, adherence to Secure Coding Practices, the implementation of Secure Communication protocols, routine Security Testing, and the fortification of Third-party Libraries and Application Programming Interfaces (APIs).

A review of secure software development standards concludes that many do not cover all the security requirements for secure software development when used individually. Instead, a framework to guide the application of relevant standards is required [30].

Next, commonly used industry practice approaches for mobile app security are briefly discussed. The National Institute of Standards and Technology (NIST) and Open Web Application Security Project (OWASP) are described.

 ISBN: 978-1-68558-092-6

The MITRE ATT&CK knowledgebase is also included in the discussion, as MITRE provides many techniques and tools to mitigate mobile app threats. Finally, DevSecOps is discussed as it is a trending practise that brings security into the early stages of software development.

A. NIST

The National Institute of Standards and Technology (NIST) continually updates and publishes many software development and security regulations, guidelines, and rules. For example, the NIST 800-163, Vetting the Security of Mobile Applications security framework combines various security-focused stages [8]. NIST 800-163 focuses on mobile security and is a generic framework for mobile applications. Mobile app teams use different stages within 800-163 and apply the knowledge gained to improve the security of a mobile app. For example, the App Security Requirements stage identifies general security requirements, organisation-specific requirements and risk tolerance. In addition, each company's development teams use their expertise and skills to identify threats and risks of their mobile app. Development teams execute mobile app security testing cycles using the NIST App Testing and Vulnerability Classifiers. Other security stages are App Vetting System, App Vetting Considerations and App Vetting Process. In addition, NIST 800-218 Secure Software Development Framework (SSDF) [31] provide comprehensive guidelines to mitigate risks.

B. OWASP

The Open Web Application Security Project is a security standard that contributes to mobile app security by introducing Mobile Security Testing Guide (MSTG) and Mobile Application Security Vetting System (MASVS) security categories [32] [33]. The MSTG and MASVS categories comprise security focus areas for mobile app development. Mobile app teams use various stages within MSTG and MASVS and apply the knowledge gained to improve mobile app development security. For example, the development and testing teams invoke Tampering and Reverse Engineering Code. Teams analyse the application code and web server requests sent and received between the mobile app server to give feedback to the broader team regarding mobile vulnerabilities. Other security focus areas within the MSTG are Data Storage, Authentication mechanisms, Code Quality, and Anti-Reversing techniques. The MASVS stipules eight vulnerability areas that concentrate on mobile app security. Additionally, mobile app teams use the areas as guidelines for vulnerabilities companies should address. Since August 2020, the CREST alliance have introduced the OWASP MASVS as an officially certified standard for mobile app security [34] [35].

C. MITRE ATT&CK

The MITRE's ATT&CK is a widely known and utilised knowledgebase to understand cyber-attacks or threat actors. The MITRE ATT&CK knowledge base consists of multiple platform-specific security topics to address enterprise and mobile attacks [36]. Mobile app teams use various attacks within MITRE ATT&CK Mobile and apply the knowledge gained to improve developers' mobile app development security expertise. For example, developers can invoke onboot or login initialisation scripts, analyses application code and invoke onboot scripts to bypass mobile device manufacturer security checks. In addition, login initialisation scripts are used to analyse the login web server calls and ultimately manipulate the values to masquerade as a different customer.

D. DEVSECOPS

DevSecOps [37] is a software development approach that combines development, security, and operations to enable the creation of secure, reliable, and high-quality software products. It is built over DevOps, therefore promoting the integration of security principles and practices through collaboration, communication, and team integration. DevSecOps emphasises secure coding practices to prevent vulnerabilities in the code, requires continuous testing and integration to ensure that the application is secure and reliable using automated testing tools, ensures that new features and updates are released quickly and securely, involves continuous security monitoring and incident response to detect and respond to security threats in real-time. Even though DevSecOps promises much, aspects, such as security or privacy by design, architectural risk analysis, threat modelling, and risk management are complex to implement as substantial human input is required to execute these processes. Silo-based teams are a barrier to secure DevOps that prevent collaboration [38].

Next, nine security drivers are described from an analysis of the mobile app ecosystem and literature to uniquely focus the processes and activities required for secure mobile app development. Both management and technical drivers are identified to provide a more comprehensive approach

IV. SECURITY DRIVERS FOR AN SECURE MOBILE SOFTWARE DEVELOPMENT FRAMEWORK

The NIST SecureSsoftware Development Framework (SSDF) [39] is a formal approach that embeds secure development activities, such as security requirements elicitation and threat modelling into the software development cycle to address mobile app security requirements, risks, vulnerabilities, development, and a vetting process.

Key security areas or drivers are presented as a first step towards establishing such a framework. Next, the security drivers are identified and described.

A. Management of software developers for security

Software development teams should be well-managed to ensure that security is a priority. By defining principles that should be followed and ensuring that cross-team collaboration becomes a way of work, a security culture can grow, resulting in motivated teams. Teams should be provided with tools to increase productivity and cut expenditures to increase the return on investment for the company. Development teams require security guidance, job satisfaction and adequate security controls to secure remote

 ISBN: 978-1-68558-092-6

work environments [40] [41]. A level of autonomy given to developers can strengthen the symbiosis between management and developers [42]. Agile development processes allow management to break the silos between various development teams [43].

B. A structured security approval strategy for security vendors

Large organisations outsource security functions to security vendors. Security vendors must be managed with care [44] as they may be required to access intricate details and propriety knowledge to reproduce any security compromise and find the cause quickly. While NIST provides comprehensive guidance across multifarious facets of mobile application development, it is prudent to acknowledge a noticeable void in their approach, specifically in the seamless integration of security vendors [39]. This deficiency underscores the challenges of facilitating extensive security approvals for vendors, granting them immediate access to sensitive information. This task presents considerable complexities within the existing secure framework landscape. Customers' confidentiality and privacy are of concern, and trust needs to be ensured. Multiple processes and authorisations are required in order to onboard security providers. Companies should ensure that there is a good motivation for appointing a vendor based on a security gap, and a return on investment (ROI). A legally binding contract needs to identify whether the vendor allows a proof-of-concept trial period, addresses a security gap for the mobile app and whether they will be a long-term partner as these factors influence the company's security posture.

C. Integrate security education into secure software development

In agile software development, there is no phase for security training, but it is assumed that developers have the required knowledge. Developers must be convinced that security is part of their job and trained to add security to their code competently. Unfortunately, adequate mobile app security training is not a focus for development teams [45]. To foster a security-first mobile app, best practices and design techniques must be encouraged [46]. Security education can include specialised security training and certifications of individuals. For those in technical roles, security training should be mandatory. [47] offers a commendable comparative analysis of security certifications accessible to development teams. These certifications serve as valuable resources for mobile developers to refine and authenticate their proficiency in mobile security and other security-related domains. Furthermore, knowledge of security vendors' products and services is another potential gap in the education of developers and security specialists. Integrating a security vendor product with the mobile app and systems requires expertise, extensive documentation and practical experience. In addition, where there is a high staff turnover, a knowledge gap can pose a risk to the company as it may become vague about the vendor's role due to a lack of experience. In such cases, the vendor is responsible for driving the process, which may be to the company's detriment.

D. Standardised secure software development practices and coding principles

A standardised approach to mobile app development is required in the fast-changing mobile app environment. Various secure software development practices and coding principles exist, with development teams having a range of experience and skills gained over the years to create their mobile app. Unfortunately, over time, teams apply several approaches in a non-standard manner, thereby creating the potential for gaps in their security posture. Using the DevOps approach, companies implement security using a layered approach [19] as DevSecOps focuses on implementing software security practices and tools at every stage of the lifecycle. DevSecOps requires an increased focus on collaborations between the development, security, and operations teams to be effective. Unfortunately, different groups may experience intergroup conflicts and may not trust each other.

A mobile app contains various features, performance enhancements, user interface and security, to name a few [48]. Therefore, companies should comply with a standardised level of best practices and coding principles for their mobile apps to keep up with the ever-changing advancements in technology and practices. For example, the OWASP secure coding practices guide can be extended to specifically include mobile app secure coding requirements [7]. In addition, security design patterns help mitigate issues and ensure that trust is established within the mobile ecosystem [7].

E. A baseline set of standardised security mechanisms for mobile apps

Trust within a mobile ecosystem entails app confidentiality, communication integrity between the app and company, app availability, customer authentication and authorisation, and non-repudiation of a financial transaction [49]. Introducing new security mechanisms and altering an existing mechanism require intricate security knowledge of the mobile app, code practices and all mobile ecosystem actors. Developers must implement new security mechanisms without guidance, as no approved library exists. Security mechanisms, such as OTP, Device Registration, Image verification, Passphrase, Digital certificates and Biometrics, and more could fall victim to attackers due to vulnerabilities caused by inexperienced developers. Therefore, companies would benefit from standardised security mechanisms that follow best practices [50]. In conjunction with standardised software development techniques, standardised security mechanisms would support the seamless updating of security mechanisms.

F. Standardised threat modelling approach

Threat modelling is a critical element in integrating security into a mobile app. There are various threat modelling approaches and methodologies that can be employed. Unfortunately, threat modelling for agile

 ISBN: 978-1-68558-092-6

development is immature, and few sources are available to consult [51]. Moreover, a specific threat model often limits threats to the mobile app's trust boundaries and communication channels, thereby neglecting potential threats. Threat modelling is a challenging task and is potentially best addressed using a hybrid approach, to combine the best features of different approaches. Currently, no such approach exists for developing a secure mobile app. A potential approach should be informed by the specifics of the environment and software architecture and threats and vulnerabilities specific to the environment.

G. Standardise testing schedule

Software testing is a complex phase that takes time and is very costly. Furthermore, software developers may resist the full scope of required testing, lowering overall software quality. Mobile-specific security testing is often performed only at the end of the development life cycle, just before a release date. As a result, releases may be shipped with a risk. An extensive range of open-source tools and techniques are available for companies to automatically validate security mechanisms and mobile security controls [31]. Penetration tests find vulnerabilities within software systems. While it is recognized that no system can be entirely devoid of flaws, it is imperative to emphasize the significance of adhering to a standardized penetration testing regimen. Such a regimen is obligatory for proactively addressing and mitigating 'low-hanging fruit' vulnerabilities, often exploited by cybercriminals as prime targets [52]. To address all vulnerabilities, security tests need to be conducted by multiple teams within a company. Extensive security testing can only be performed when a formalised threat modelling approach is present and teams are well-trained and knowledgeable.

H. Standardised mobile app vetting system for an industry

App vetting determines whether an app conforms to an organisation's security requirements. Each app store has its unique in-house requirements and vetting processes. A mobile app should be vetted to assure customers that the required security mechanisms are implemented [53]. The mobile app vetting system informs customers of security mechanisms and security controls embodied within the mobile app and the precautions the company took to ensure trust between the mobile app, customer and company. Unfortunately, app stores do not thoroughly examine all security vulnerabilities due to cost constraints. Therefore, each industry requires a more stringent mobile app vetting process to contribute to secure mobile apps.

I. Regulated security reporting and collaboration

A software development framework for secure mobile apps requires interaction between companies, industry custodians, and regulators with authority. Furthermore, reporting incidents and knowledge sharing should be compulsory to ensure a more robust community. For example, a successful attack against a new sophisticated mobile security mechanism must be communicated to safeguard the community. The landscape of security frameworks is abundant, with numerous frameworks catering to diverse regulations and industry sectors [54]. Nonetheless, a conspicuous need arises for a dedicated secure mobile application development framework focusing on security and agile methodologies. Furthermore, while many security controls exist for creating secure mobile applications [55], it becomes evident that various industries must select and extract controls that align with their specific requisites and prioritize their significance. Governance and regulation are essential in safeguarding mobile apps and preventing cybercrime.

Next, the security drivers identified by this research are compared to the software development guidelines, standards, and knowledgebase. The comparison aims to determine to what extent the current frameworks address the identified driver

V. EVALUATION

This research aims to define a secure software development framework that a software developer or team, who may have limited security knowledge, can use as a guide to secure a mobile app throughout all development life cycle phases. Such a guide can be tailored to an industry, such as banking to reflect its security concerns and best practise.

A research gap is now identified by comparing the various development approaches and guidelines and the identified security drivers. The research gap forms a foundation for developing a secure mobile app development framework. Table 1 compares the security drivers and frameworks, followed by an evaluation. OWASP meets several security drivers. However, managing development teams to develop secure mobile apps and enabling software developers to work remotely from the comfort of their choice is cumbersome and is lacking from all the frameworks. Furthermore, integrating a secure development framework into the various sectors with the structured industry-and-governance process is also missing. NIST guides the validation of security controls within a mobile app by invoking a penetration testing schedule and educating development teams on various security-related topics. Although NIST directs threat modelling, the assistance only applies within the Recommended Standard for Vendor or Developer Verification Code [56]. The MITRE ATT&CK framework educates development teams on various security-related topics from an attacker's perspective, providing in-depth knowledge and relevant tools to attack a mobile app. However, MITRE and NIST lack fundamental security mechanisms for companies to authenticate and authorise customers and ensure integrity, confidentiality and non-repudiation in financial transactions.

DevSecOps provides support for managing teams and collaborations without focusing on security aspects. A focus is on the integration of, e.g., education and testing into all phases. No guidance is provided for a baseline set of security mechanisms and guidelines to follow.

 ISBN: 978-1-68558-092-6

TABLE I. COMPARISON OF SECURE DEVELOPMENT FRAMEWORKS AND SECURITY DRIVERS

Security drivers	NIST	OWASP	MITRE	DEVSECOPS
Management of software developers for security				X
A structured security approval strategy	X			
Integrate security education for secure software development	X	X	X	X
Standardised secure software development practices and coding principles		X		
A baseline set of standardised security mechanisms for mobile apps				
Standardised threat modelling approach		X		
Standardise testing schedule	X			X
Standardised mobile app vetting system for an industry		X		
Regulated security reporting and collaboration				

OWASP supports more security drivers than NIST and MITRE as OWASP assists with testing tools using the Mobile Security Testing Guide and vetting mobile apps using the Mobile Application Security Vetting System. OWASP is an open-source and freely available repository of security controls for applying mobile security controls.

The comparison in Table 1 indicates a need for a secure development framework explicitly tailored to the mobile app space. As this environment is under attack and will be more so in the future, it is worth developing a tailored framework to ensure better security.

VI. CONCLUSION AND FUTURE WORK

This paper identified the complexities of a mobile ecosystem and the lack of guidance for software developers. Software development teams implement and test new and often sophisticated security mechanisms, as no standard approach currently exists. Additionally, the lack of adequate guidelines and frameworks for secure mobile app development creates challenges for software developers who require specialist security knowledge of mobile apps, application frameworks, and operating systems. While frameworks, such as OWASP, NIST and MITRE ATTACK, and approaches such as DevSecOps are excellent foundations for supporting security within a mobile app, they do not fully support the need to develop a secure mobile app. The research suggests the need for a secure software development framework tailored to the security requirements of mobile apps. The study contributes to this area by identifying nine mobile app security drivers for a secure software development framework as the first step in this direction.

Future work includes an in-depth analysis of the identified security drivers and current software development approaches and frameworks to identify activities and other deliverables that can be used to create a secure software development framework for mobile apps.

REFERENCES

[1] S. Majumdar and V. Pujari, "Exploring usage of mobile banking apps in the UAE: a categorical regression analysis," *J Financ Serv Mark*, Aug. 2021, doi: 10.1057/s41264-021-00112-1.

[2] Y. W. Prihatiningtias and N. Wipraganang, "The Impact of Mobile Payment on Non-Financial Performance of SMEs During the COVID-19 Pandemic," presented at the Brawijaya International Conference on Economics, Business and Finance 2021 (BICEBF 2021), Atlantis Press, Jan. 2022, pp. 252–258. doi: 10.2991/aebmr.k.220128.033.

[3] Digital.ai, "Digital.ai." 2023. Accessed: Feb. 19, 2023. [Online]. Available: https://digital.ai/

[4] M. Divya and C. Hebbar, "A case study on 'mobile banking is a boon to banking customers during the covid-19 pandemic situation'-with special reference to the sbi customers of mangalore city," *epra*, vol. 8, no. 4, Apr. 2021, [Online]. Available: https://eprajournals.com/jpanel/upload/1243am_2.EPRA%20JOURNALS-6865.pdf

[5] M. Hensher *et al.*, "Scoping review: Development and assessment of evaluation frameworks of mobile health apps for recommendations to consumers," *Journal of the American Medical Informatics Association*, vol. 28, no. 6, pp. 1318–1329, Jun. 2021, doi: 10.1093/jamia/ocab041.

[6] GitLabs, "GitLabDevSecOps," 2022. https://about.gitlab.com/developer-survey (accessed Nov. 01, 2022).

[7] OWASP, "OWASP Secure Coding Practices-Quick Reference Guide | OWASP Foundation," 2022. https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/migrated_content.html (accessed Nov. 01, 2022).

[8] M. Ogata, J. Franklin, J. Voas, V. Sritapan, and S. Quirolgico, "Vetting the security of mobile applications," National Institute of Standards and Technology, Gaithersburg, MD, NIST SP 800-163r1, Apr. 2019. doi: 10.6028/NIST.SP.800-163r1.

[9] W. Xiong, E. Legrand, O. Åberg, and R. Lagerström, "Cyber security threat modeling based on the MITRE Enterprise ATT&CK Matrix," *Softw Syst Model*, Jun. 2021, doi: 10.1007/s10270-021-00898-7.

[10] M. A. Aljohani and S. S. Alqahtani, "A Unified Framework for Automating Software Security Analysis in DevSecOps," in *2023 International Conference on Smart Computing and Application (ICSCA)*, Feb. 2023, pp. 1–6. doi: 10.1109/ICSCA57840.2023.10087568.

[11] Y. Valdés-Rodríguez, J. Hochstetter-Diez, J. Díaz-Arancibia, and R. Cadena-Martínez, "Towards the Integration of Security Practices in Agile Software Development: A Systematic Mapping Review," *Applied Sciences*, vol. 13, no. 7, Art. no. 7, Jan. 2023, doi: 10.3390/app13074578.

[12] B. Reed, "Rethinking the Status Quo of Mobile App Security," *Security Boulevard*, Apr. 19, 2023. https://securityboulevard.com/2023/04/rethinking-the-status-quo-of-mobile-app-security/ (accessed Apr. 28, 2023).

[13] H. Taherdoost, "Understanding Cybersecurity Frameworks and Information Security Standards—A Review and Comprehensive Overview," *Electronics*, vol. 11, no. 14, Art. no. 14, Jan. 2022, doi: 10.3390/electronics11142181.

[14] D. Ghelani, T. K. Hua, and S. K. R. Koduru, "Cyber Security Threats, Vulnerabilities, and Security Solutions Models in

Banking," Preprints, preprint, Sep. 2022. doi: 10.22541/au.166385206.63311335/v1.

[15] J. R. Kala Kamdjoug, S.-L. Wamba-Taguimdje, S. F. Wamba, and I. B. Kake, "Determining factors and impacts of the intention to adopt mobile banking app in Cameroon: Case of SARA by afriland First Bank," *Journal of Retailing and Consumer Services*, vol. 61, p. 102509, Jul. 2021, doi: 10.1016/j.jretconser.2021.102509.

[16] Y. Liu, Z. Liang, C. Li, J. Guo, and G. Zhao, "An Investigation into the Adoption Behavior of mHealth Users: From the Perspective of the Push-Pull-Mooring Framework," *Sustainability*, vol. 14, no. 21, Art. no. 21, Jan. 2022, doi: 10.3390/su142114372.

[17] Y. Huang and C. Chen, "Smart App Attack: Hacking Deep Learning Models in Android Apps." arXiv, Apr. 23, 2022. Accessed: Apr. 29, 2023. [Online]. Available: http://arxiv.org/abs/2204.11075

[18] S. A. Butt, T. Jamal, M. A. Azad, A. Ali, and N. S. Safa, "A multivariant secure framework for smart mobile health application," *Trans Emerging Tel Tech*, vol. 33, no. 8, Aug. 2022, doi: 10.1002/ett.3684.

[19] R. T. Yarlagadda and M. Sabri, "An exploratory study of devops & it's future in usa," *SSRN Electronic Journal*, vol. 8, pp. 82–92, Jan. 2021.

[20] N. T. Msweli and T. Mawela, "Financial Inclusion of the Elderly: Exploring the Role of Mobile Banking Adoption," *AIP*, vol. 10, no. 1, pp. 1–21, Jun. 2021, doi: 10.18267/j.aip.143.

[21] A. Mahalle, J. Yong, and X. Tao, "Challenges and Mitigation for Application Deployment over SaaS Platform in Banking and Financial Services Industry," in *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, Dalian, China: IEEE, May 2021, pp. 288–296. doi: 10.1109/CSCWD49262.2021.9437798.

[22] M. Tahaei, K. Vaniea, K. (Kosta) Beznosov, and M. K. Wolters, "Security Notifications in Static Analysis Tools: Developers' Attitudes, Comprehension, and Ability to Act on Them," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, Yokohama Japan: ACM, May 2021, pp. 1–17. doi: 10.1145/3411764.3445616.

[23] B. Yankson, J. V. K, P. C. K. Hung, F. Iqbal, and L. Ali, "Security Assessment for Zenbo Robot Using Drozer and mobSF Frameworks," in *2021 11th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, Apr. 2021, pp. 1–7. doi: 10.1109/NTMS49979.2021.9432666.

[24] A. S. George and S. Sagayarajan, "Securing Cloud Application Infrastructure: Understanding the Penetration Testing Challenges of IaaS, PaaS, and SaaS Environments," *Partners Universal International Research Journal*, vol. 2, no. 1, pp. 24–34, 2023.

[25] R. Nataraj and S. Jagatheesan, "Lack of SDLC Models and Frameworks in Mobile Application Development - A Systematic Literature Review and Study," *Journal of Xi'an University of Architecture & Technology*, vol. 13, no. 8, pp. 250–258, Oct. 2021.

[26] C. Catal, A. Ozcan, E. Donmez, and A. Kasif, "Analysis of cyber security knowledge gaps based on cyber security body of knowledge," *Education and Information Technologies*, vol. 28, pp. 1809–1831, 2022.

[27] A. Kudriavtseva and O. Gadyatskaya, "Secure Software Development Methodologies: A Multivocal Literature Review." arXiv, Nov. 29, 2022. Accessed: May 08, 2023. [Online]. Available: http://arxiv.org/abs/2211.16987

[28] R. Fujdiak *et al.*, "Managing the Secure Software Development," in *2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, CANARY ISLANDS, Spain: IEEE, Jun. 2019, pp. 1–4. doi: 10.1109/NTMS.2019.8763845.

[29] M. Naeem, W. Ozuem, and P. Ward, "Understanding the accessibility of retail mobile banking during the COVID-19 pandemic," *IJRDM*, vol. 50, no. 7, pp. 860–879, Jun. 2022, doi: 10.1108/IJRDM-02-2021-0064.

[30] A. Ramirez, A. Aiello, and S. J. Lincke, "A Survey and Comparison of Secure Software Development Standards," in *2020 13th CMI Conference on Cybersecurity and Privacy (CMI) - Digital Transformation - Potentials and Challenges(51275)*, Nov. 2020, pp. 1–6. doi: 10.1109/CMI51275.2020.9322704.

[31] M. Hassan, Z. Shukur, and M. Mohd, "A Penetration Testing on Malaysia Popular e-Wallets and m-Banking Apps," *International Journal of Advanced Computer Science and Applications*, vol. 13, Jun. 2022, doi: 10.14569/IJACSA.2022.0130580.

[32] E. B. Blancaflor, G. A. J. Anson, A. M. V. Encinas, M. A. V. Marin, and S. L. G. Zamora, "A Vulnerability Assessment on the Parental Control Mobile Applications' Security: Status based on the OWASP Security Requirements," p. 10, 2021.

[33] A. Pradeep *et al.*, "A comparative analysis of certificate pinning in Android & iOS," in *Proceedings of the 22nd ACM Internet Measurement Conference*, Nice France: ACM, Oct. 2022, pp. 605–618. doi: 10.1145/3517745.3561439.

[34] CREST, "CREST OVS Web Application Programme," *CREST*, 2022. https://www.crest-approved.org/membership/crest-ovs-programme/ (accessed May 08, 2023).

[35] H. Al-Shaikh *et al.*, "SHarPen: SoC Security Verification by Hardware Penetration Test," in *Proceedings of the 28th Asia and South Pacific Design Automation Conference*, Tokyo Japan: ACM, Jan. 2023, pp. 579–584. doi: 10.1145/3566097.3567918.

[36] R. Al-Shaer, J. M. Spring, and E. Christou, "Learning the Associations of MITRE ATT&CK Adversarial Techniques," *arXiv:2005.01654 [cs]*, May 2020, Accessed: Dec. 08, 2021. [Online]. Available: http://arxiv.org/abs/2005.01654

[37] H. Myrbakken and R. Colomo-Palacios, *DevSecOps: A Multivocal Literature Review*. 2017, p. 29. doi: 10.1007/978-3-319-67383-7_2.

[38] R. N. Rajapakse, M. Zahedi, M. A. Babar, and H. Shen, "Challenges and solutions when adopting DevSecOps: A systematic review." arXiv, Jul. 29, 2021. Accessed: Dec. 12, 2022. [Online]. Available: http://arxiv.org/abs/2103.08266

[39] M. Souppaya, K. Scarfone, and D. Dodson, "Secure Software Development Framework (SSDF) Version 1.1: (Draft): Recommendations for Mitigating the Risk of Software Vulnerabilities," National Institute of Standards and Technology, Gaithersburg, MD, NIST 800-218, Feb. 2022. doi: 10.6028/NIST.SP.800-218.

[40] D. Russo, P. P. H. Hanel, S. Altnickel, and N. van Berkel, "The Daily Life of Software Engineers during the COVID19 Pandemic." arXiv, Jan. 12, 2021. Accessed: Aug. 19, 2022. [Online]. Available: http://arxiv.org/abs/2101.04363

[41] S. Mandal and D. Khan, *A Study of Security Threats in Cloud: Passive Impact of COVID-19 Pandemic*. 2020. doi: 10.1109/ICOSEC49089.2020.9215374.

[42] S. Fraser and D. Mancl, "Engineering for Chaos: Lessons Learned from COVID-19," vol. 46, no. 2, p. 3, 2021.

[43] C. NicCanna, M. A. Razzak, J. Noll, and S. Beecham, "Globally Distributed Development during COVID19." arXiv, Mar. 31, 2021. Accessed: Aug. 19, 2022. [Online]. Available: http://arxiv.org/abs/2103.17181

[44] S. Viveka, "Lessons learnt from COVID-19 for business continuity management in banking sector," in *Building Resilient Organizations: Predicaments & Prospects*, 2022, pp. 241–251. Accessed: Aug. 25, 2023. [Online]. Available: https://books.google.co.za/books?hl=en&lr=&id=XrOGEAAAQBAJ&oi=fnd&pg=PA241&dq=Covid+vendors+challenge+banking+&ots=ezXw5xnOpC&sig=NZGqRg_5QFae5I_B2LLNIlpY5Nc&redir_esc=y#v=onepage&q&f=false

[45] E.-L. Nawa, M. Chitauro, and F. B. Shava, "Assessing Patterns of Cybercrimes Associated with Online Transactions in Namibia Banking Institutions' Cyberspace," in *2021 3rd International Multidisciplinary Information Technology and Engineering*

Conference (IMITEC), Nov. 2021, pp. 1–6. doi: 10.1109/IMITEC52926.2021.9714697.

[46] A. “bunnie” Huang, “Betrusted: Improving Security Through Physical Partitioning,” *IEEE Pervasive Computing*, vol. 19, no. 2, pp. 13–20, Apr. 2020, doi: 10.1109/MPRV.2020.2966190.

[47] E.-C. Davri *et al.*, “Cyber Security Certification Programmes,” in *2021 IEEE International Conference on Cyber Security and Resilience (CSR)*, Rhodes, Greece: IEEE, Jul. 2021, pp. 428–435. doi: 10.1109/CSR51186.2021.9527974.

[48] A. Patidar and U. Suman, “Towards Analyzing Mobile App Characteristics for Mobile Software Development,” in *2021 8th International Conference on Computing for Sustainable Global Development (INDIACom)*, Mar. 2021, pp. 786–790.

[49] Q. Hammouri, T. Majali, D. Almajali, A. Aloqool, and J. A. AlGasawneh, “Explore the Relationship between Security Mechanisms and Trust in E-Banking,” vol. 26, no. 6, pp. 17083–17093, 2021.

[50] F. Alt and S. Schneegass, “Beyond Passwords—Challenges and Opportunities of Future Authentication,” *IEEE Secur. Privacy*, vol. 20, no. 1, pp. 82–86, Jan. 2022, doi: 10.1109/MSEC.2021.3127459.

[51] K. Bernsmed, D. S. Cruzes, M. G. Jaatun, and M. Iovan, “Adopting threat modelling in agile software development projects,” *Journal of Systems and Software*, vol. 183, p. 111090, Jan. 2022, doi: 10.1016/j.jss.2021.111090.

[52] N. Kshetri, “Cybercrime and Cybersecurity in Africa,” *Journal of Global Information Technology Management*, vol. 22, no. 2, pp. 77–81, Apr. 2019, doi: 10.1080/1097198X.2019.1603527.

[53] T. L. Andarzian, “SANT: Static Analysis of Native Threads for Security Vetting of Android Applications,” vol. 14, no. 1, p. 13, 2022.

[54] CISecurity, “3 Ways We’ve Made the CIS Controls More Automation-Friendly,” *CIS*, 2023. https://www.cisecurity.org/insights/blog/3-ways-weve-made-the-cis-controls-more-automation-friendly (accessed Aug. 25, 2023).

[55] D. Miessler, “The Consumer Authentication Strength Maturity Model (CASMM) V6,” 2023. https://danielmiessler.com/p/casmm-consumer-authentication-security-maturity-model/ (accessed Aug. 25, 2023).

[56] A. Shostack, “NIST Brings Threat Modeling into the Spotlight,” *Dark Reading*, Sep. 23, 2021. https://www.darkreading.com/threat-intelligence/nist-brings-threat-modeling-into-the-spotlight (accessed Oct. 25, 2022).

 ISBN: 978-1-68558-092-6