R3 Abstract-Intro + LR + Meth+concl.docx

by Turnitin LLC

Submission date: 15-May-2024 02:16AM (UTC-0400)
Submission ID: 2361665229
File name: tmphuq_xv3x (58.03K)
Word count: 4786
Character count: 30066

INTEGRATING SECURITY IN APPLICATION DEVELOPMENT LIFECYCLE USING PROTECTION POKER TECHNIQUE

Abstract : This research work presents a methodology for effectively addressing critical software security risks (CSSRs) through effective risk management techniques, particularly focusing on the application of protection poker. Beginning with a meticulous identification and assessment of the specified risks, the methodology utilizes protection poker to collaboratively evaluate the likelihood and impact of each risk, fostering informed decision-making and prioritization. Quantifying risks using a numeric scale enables a comprehensive understanding of their severity, facilitating focused resource allocation and mitigation efforts. Through a comprehensive understanding of potential vulnerabilities and proactive mitigation efforts facilitated by protection poker, organizations can prioritize resources effectively and ensure the successful outcome of projects and initiatives in today's dynamic threat landscape.

1. **Introduction**. In the current era of 5G revolution where mobile phones are easily accessible to all masses, cyber security has gained paramount importance. Security attacks in the cyber domain have increased significantly and hence calls for increased protective measures.

Security in Application Development has gained substantial importance over the recent times owing to hacking and other attacks on computer systems. As a result, security has to be made intrinsic component of all the stages of mobile application development. If sensitive data is hacked or exposed, it may cause un-repairable loss to software companies repute with partners, customers and investors, therefore system managers and users are paying more and more attention to this important aspect of security. In order to have complete secure applications/ softwares, security aspects needs to be considered into every step of Software Development Life Cycle (SDLC) or Application Development Life Cycle (ADLC). To blend security into the software engineering model, it should be considered from the beginning of the SDLC/ ADLC [1]. Most organizations normally see security as a post-development process. Security concerns have a profound impact on the overall quality of software, as organizations grappling with insecure software are compelled to address and rectify unreliable applications, while also impeding the progress of other concurrent projects. Emerging cyber vulnerabilities, both internal and external to the organization, continue to surface persistently, posing formidable financial risks and jeopardizing the integrity of critical company data. The repercussions of such security lapses extend beyond monetary losses, encompassing substantial credibility deficits that can be detrimental to the reputation and standing of the organization [2].

This research aims to ascertain the effectiveness of Protection Poker technique as an effective risk estimation tool of CSSRs identified in security assurance model (SAM) [3]. This estimation would enable to identify areas which requires special attention in regards to security of the application under development. As a result, software houses can assess their level of security assurance and capability to produce more secure applications.

2. **LITERATURE REVIEW**.

a. **Definition and Importance of Integrating Security**. Integrating security in the Application Development Life Cycle (ADLC) refers to the systematic inclusion of security considerations and measures at every stage of software development. This approach is crucial for creating applications resilient to cyber threats, thereby protecting both the software providers and their users from potential breaches and losses. Security integration in ADLC is not just a technical necessity but also a strategic business decision, as it significantly reduces vulnerabilities and the costs associated with post-deployment security fixes [1]. Historically, security was often an afterthought in application development, addressed only after the software's design or even after deployment. This reactive approach led to increased vulnerabilities and exploitation risks. Over time, the evolution of security practices in ADLC has been marked by a shift towards proactive measures, with methodologies like DevSecOps gaining popularity for integrating security into all stages of the software development process.

(i) **Strategies and Practices.** Secure coding practices involve writing code with security in mind, aiming to prevent vulnerabilities at the source. This includes adhering to coding standards that avoid common security pitfalls and implementing code analysis tools to detect and rectify security flaws [4].

(ii) **Security Requirements Engineering.** This process involves identifying and documenting security requirements early in the development process. It ensures that the software is designed with these requirements in mind, thus embedding security into the very foundation of the application [5].

(iii) **Risk Assessment Methodologies.** Risk assessment in ADLC involves evaluating the potential risks associated with security threats and vulnerabilities. Techniques like the Failure Modes and Effects Analysis (FMEA) are used to systematically analyze potential failure points and their impacts on application security [6].

b. **Security in Different Stages of ADLC**

(i) **Initial Planning and Design.** In this phase, threat modeling and secure design principles are crucial. Threat modeling helps in identifying potential security threats and designing countermeasures, while secure design principles guide the architecture of the application for robust security [7].

(ii) **Implementation.** Secure coding guidelines and code reviews are essential in this stage. They ensure that the code is not only functionally correct but also secure against known vulnerabilities [7].

(iii) **Testing.** Security testing methods such as static and dynamic analysis are employed to uncover vulnerabilities that might have slipped through the earlier stages. This includes penetration testing and code reviews [8].

(iv) **Deployment.** Secure deployment practices and vulnerability management are critical at this stage to ensure that the application remains secure in its operating environment [8].

(v) **Maintenance.** Continuous monitoring and incident response are key to maintaining the security of the application post-deployment. This involves regular updates and patches to address new vulnerabilities [8].

c. **Types and Lifecycle of Cross-Platform Application Development Platforms.** Cross-platform development involves creating software applications that are compatible with multiple operating systems or platforms, using a single codebase. This approach offers several benefits like Cost-Effectiveness, Time Efficient, Broader [9].

(i) **Popular Cross-Platform Frameworks**

a) **React Native.** Developed by Facebook, React Native allows developers to build mobile apps using JavaScript and React. It inherits JavaScript's security issues, such as cross-site scripting (XSS). Additionally, reliance on third-party libraries can introduce vulnerabilities [10]

b) **Flutter.** Created by Google, Flutter is an open-source UI software development kit for building natively compiled applications for mobile, web, and desktop from a single codebase. Flutter apps are relatively secure but face risks related to insecure data storage and insecure communication. The framework's novelty also means fewer security resources and community support compared to more established frameworks [11].

c) **Xamarin.** Xamarin, a Microsoft-owned framework, uses C# for building Android and iOS apps. Xamarin apps can be susceptible to typical .NET security issues. Challenges include secure data storage, proper use of cryptography, and ensuring secure communication [12].

(ii) **Comparing Security Considerations with Native Platforms**

a) Code Security: While native apps require platform-specific security measures, cross-platform apps must address security across multiple environments [13].

b) Data Storage: Cross-platform apps often rely on shared security practices for data storage, which may not be as robust as platform-specific solutions [14].

c) API Security: Security of APIs in cross-platform apps is critical, as they often access the same back-end from different platforms.

d) Regular Updates: Cross-platform apps might face challenges in simultaneous updating across all platforms, affecting security patching consistency [15].

d. **Potential Research Gap**. The literature review exposes a potential gap in quantitatively estimating the severity of 46 x CSSRs identified by a SAM during the ADLC. While various security practices and risk assessment methodologies exist, a collaborative approach to estimate the impact of these risks on application security seems to be lacking. 46 x security risk identified by Security Assurance Model are given below :-

Ser	Issue
a.	Injection
b.	Broken authentication and session management
c.	Cross-site scripting (XSS)
d.	Insecure direct object references
e.	Security mis-configuration
f.	Sensitive data exposure

Ser	Issue
g.	Missed function-level access control
h.	Cross-site request forgery (CSRF)
i.	Using components with known vulnerabilities
j.	Un-validated redirects and forwards
k.	Insufficient logging and monitoring
l.	Broken access control
m.	Improper input user data
n.	Buffer overflow
o.	Improper error handling
p.	Race conditions
q.	Failure to restrict URL access
r.	Insufficient transport layer protection
s.	Server-side request forgery (SSRF)
t.	Insecure cryptographic storage
u.	Session fixation
v.	Poor password policy and management
w.	Improper authorization
x.	Click-jacking
y.	Excessive data exposure

Ser	Issue
z.	Broken anti-automation defenses
aa.	Improper file and resource protection
bb.	Not enough security configuration
cc.	Broken business logic
dd.	Improper certificate validation
ee.	Time and state-related attacks
ff.	Fail to restrict upload of dangerous file types
gg.	Insufficient session expiration
hh.	Cryptographic issues
ii.	Insecure communications
jj.	Inadequate encryption strength
kk.	Authentication bypass
ll.	Improper use of a security feature
mm.	XML external entities (XXE)
nn.	Insufficient security controls in a third-party service
oo.	Missing security headers
pp.	Broken cryptography usage
qq.	Improper asset management
rr.	Out-of-band channel exploitation

Ser	Issue
ss.	Security relevant mis-configuration
tt.	Elevation of privilege

e. **<u>Proposed Solution</u>**. Proposed solution involves interaction with software houses to evaluate effectiveness of protection poker technique in mitigating identified critical software security risks. Feed back of these software houses is also taken through a questionnaire. In Protection Poker, participants estimate the impact (likelihood and consequence) of security risks using playing cards. This game like approach fosters discussion and encourages a shared understanding of security risks within the development team. By incorporating Protection Poker into the ADLC, development teams can collaboratively estimate the severity of CSSRs identified by a SAM, enabling them to prioritize security efforts and resources on the most critical risks. Here's how Protection Poker can be integrated into the ADLC:

(i) Identify CSSRs: During the security assurance stage, a SAM would be employed to identify potential CSSRs within the application under development.

(ii) Prepare Protection Poker Session: A Protection Poker session would be organized involving developers, security professionals, and other relevant stakeholders.

(iii) Estimate Risk Impact: Each CSR would be discussed, and participants would use playing cards to estimate the likelihood and consequence of the risk. The assigned card values would then be combined to generate a risk score.

(iv) Prioritize Security Focus: Based on the risk scores assigned through Protection Poker, development teams can prioritize their security efforts on the most critical CSSRs.

Research Questionnaire:

1. **Identifying Risks:** How does your software house identify the 46 security risks outlined in the security assurance model during the application development lifecycle?
2. **Risk Management:** What strategies or methods do you use to manage and mitigate the identified risks?
3. **Risk Assessment:** How do you integrate risk assessment methods like the protection poker technique into your application development process?
4. **Risk Prioritization:** Based on the risk estimation provided, how do you prioritize risks and allocate resources for their mitigation?
5. **Protection Techniques:** Can you share your experiences with using protection poker or similar techniques to assess and address these risks during application development?
6. **Challenges and Solutions:** What specific challenges have you faced in managing the aforementioned risks, and what strategies have you found most effective in addressing them?

Proposed Methodology :

The methodology employed in this research involves the assessment and analysis of 46 security risks that have been identified through the security assurance model. To estimate the levels of risk associated with these vulnerabilities, the protection poker technique is utilized. This technique offers a structured and systematic approach to evaluate both the likelihood and impact of each risk, allowing for a comprehensive understanding of the potential threats and enabling the quantification of the risk level for each identified security risk.

The process includes the following steps:

1. **Risk Identification and Assessment:**

    a. The process commences by meticulously identifying the 46 risks specified within the security assurance model. These risks encompass a wide array of potential vulnerabilities and threats that could compromise the security and integrity of the application. By delving into the characteristics of each risk, we gain a deeper understanding of their nature, scope, and potential implications for the overall security posture of the application.

b. Moreover, it is essential to explore the root causes that underlie each risk, as this knowledge is instrumental in devising effective mitigation strategies. By discerning the factors that contribute to the emergence of these risks, we can proactively address vulnerabilities and bolster the resilience of the application against potential security breaches.

c. Equally significant is the assessment of each risk using the protection poker technique, a collaborative approach that leverages the expertise of team members versed in security and development domains. This technique entails a structured discussion wherein team members evaluate the likelihood and impact of each risk, drawing upon their collective insights and experiences to arrive at a consensus.

d. Through the protection poker technique, team members engage in a deliberative process that fosters critical thinking and informed decision-making. By assessing the likelihood of occurrence and the potential impact of each risk, team members can prioritize their mitigation efforts and allocate resources judiciously to address the most pressing security concerns

2. **Risk Estimation:**

a. In order to evaluate the risks associated with a particular situation or project, it is essential to quantify them using a numeric scale. This scale typically ranges from 1 to 10, with 1 indicating the lowest level of risk and 10 representing the highest. By assigning a numerical value to each risk, it becomes easier to assess both the likelihood of occurrence and the potential impact it could have.

b. The process of quantifying risks involves considering various factors that contribute to their overall severity. This includes analyzing the probability of a risk occurring and the extent to which it could affect the desired outcome. By breaking down risks into these two components - likelihood and impact - it becomes possible to create a more comprehensive understanding of their potential consequences.

c. Once the likelihood and impact of each risk have been assessed and assigned a numerical value, the next step is to combine these scores to produce an overall risk score. This combined score provides a clear indication of the level of risk associated with each potential issue, allowing decision-makers to prioritize their responses accordingly.

d. By following this structured approach to risk assessment, organizations can better understand the potential threats they face and take proactive measures to mitigate them. Ultimately, quantifying risks using a numeric scale enables more informed decision-making and helps to ensure the successful outcome of projects and initiative

3. **Data Analysis:**

a. When evaluating risk levels within an organization, it is crucial to compare them to industry standards and best practices. By doing so, companies can gain a better understanding of where they stand in relation to their peers and identify areas that may require further attention. This process allows for a more comprehensive assessment of potential risks and the development of effective mitigation strategies.

b. One key aspect of this evaluation is to identify risks with the highest estimated scores. These are the risks that pose the greatest threat to the organization and require immediate attention. By prioritizing these risks for further investigation and mitigation strategies, companies can allocate their resources more effectively and focus on addressing the most pressing issues.

4. **Data Collection from Software Houses:**

The questionnaire was distributed to three software houses (AppInSnap, Omnisoftex & Centangle Interactive) in order to gather their input on how the identified risks could be addressed during the application development lifecycle. The responses are being used to understand how the risks are managed in industry practices and to validate the risk estimations. By distributing the questionnaire and analyzing the responses, a comprehensive understanding of industry practices in managing risks is being obtained, and the accuracy of the risk estimations is being verified. This approach allows for a thorough examination of how different software houses address risks in their application development processes, contributing valuable insights to the research on risk management in software development

5. **Synthesis and Analysis:**

   The results from the risk estimation process and the feedback from the software houses are being analyzed. The data is being carefully examined to determine the potential risks and to understand the input from the software houses. This process involves thorough evaluation and consideration of the information gathered. The feedback from the software houses is being taken into account and is being used to inform decision-making. The risk estimation is being conducted to assess the potential hazards and to develop strategies to mitigate these risks. Overall, the findings from this analysis will provide valuable insights for the research.

By combining the feedback from the software houses with the risk estimations, the study aims to gain insight into industry practices and comprehend security requirements in a much better way to mitigate them in early phase of development.

RESULTS

Response from AppInSnap:

Identifying Risks:

AppInSnap employs a proactive approach to identify the 46 security risks outlined in the security assurance model during the application development lifecycle. They conduct thorough risk assessments at the beginning of each project, utilizing industry-standard frameworks and guidelines. The team collaborates closely with clients to understand their specific security requirements and potential vulnerabilities. Additionally, they leverage automated security testing tools and manual code reviews to identify any potential risks throughout the development process.

Risk Management:

AppInSnap's risk management strategy revolves around implementing robust security measures at every stage of the development lifecycle. They follow industry best practices and standards, such as OWASP guidelines, to mitigate identified risks effectively. The team continuously monitors and updates security protocols to address emerging threats and vulnerabilities promptly. Additionally, they conduct regular security audits and penetration testing to ensure the integrity of their applications.

Risk Assessment:

At AppInSnap, risk assessment methods like the protection poker technique are integrated into the application development process. This technique allows them to collaboratively evaluate the likelihood and impact of each identified risk, ensuring a comprehensive understanding of potential threats. By involving stakeholders from various domains, including security and development, they can prioritize risks and allocate resources more effectively.

Risk Prioritization:

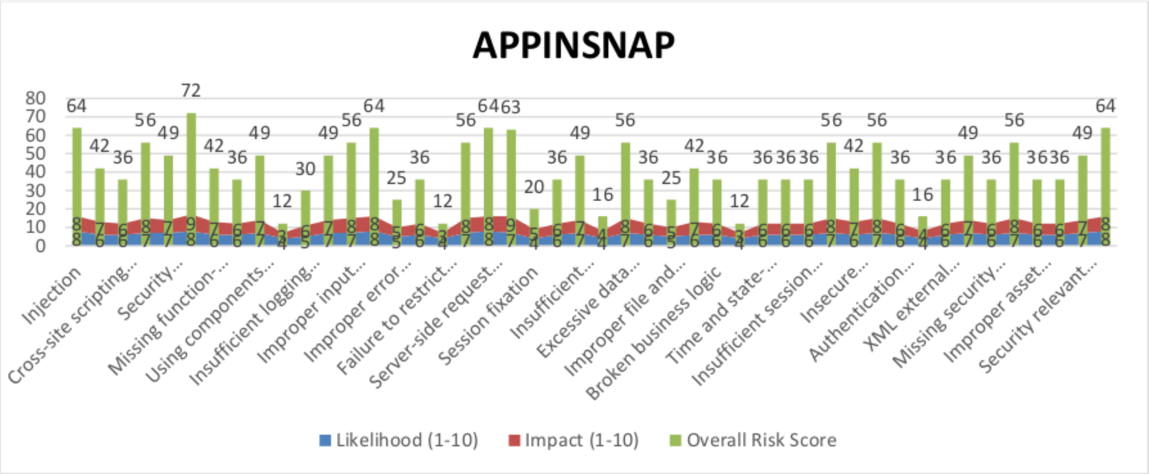
Based on the risk estimation provided by the protection poker technique, AppInSnap prioritizes risks according to their potential impact on the application and organization. High-risk areas are given immediate attention, with resources allocated accordingly for mitigation. The team regularly reviews and updates risk priorities as the project progresses to maintain a proactive approach to risk management.

Protection Techniques:

For the purpose of this questionnaire, AppInSnap specifically used the protection poker technique for identified 46 risks mentioned in the Security Assurance Model.. The structured approach of protection poker allows them to make informed decisions and prioritize mitigation efforts effectively.

Challenges and Solutions:

In managing security risks, AppInSnap has encountered challenges such as keeping pace with evolving threats and maintaining a balance between security and development timelines. To address these challenges, they invest in continuous training and education for their team members, stay updated with the latest security trends, and implement agile security practices. Additionally, fostering a culture of security awareness and collaboration within their organization has proven to be an effective strategy in mitigating risks.



Response from OmniSoftex:

Identifying Risks:

OmniSoftex has established processes and protocols to identify the 46 security risks outlined in the security assurance model during the application development lifecycle. They conduct comprehensive threat modeling exercises, code reviews, and security assessments to identify potential vulnerabilities. The team collaborates closely with clients to understand their security requirements and prioritize risks based on their potential impact.

Risk Management:

OmniSoftex's risk management approach focuses on implementing a multi-layered security framework to mitigate identified risks effectively. They follow industry best practices and standards, such as ISO 27001 and NIST, to ensure the security of their applications. Additionally, they utilize advanced security tools and technologies to monitor and detect security threats in real-time.

Risk Assessment:

At OmniSoftex, risk assessment methods like the protection poker technique are integrated into the application development process. This technique allows them to systematically evaluate and prioritize risks based on their likelihood and impact. By involving stakeholders from different departments, including security, development, and operations, they ensure a holistic approach to risk assessment.

Risk Prioritization:

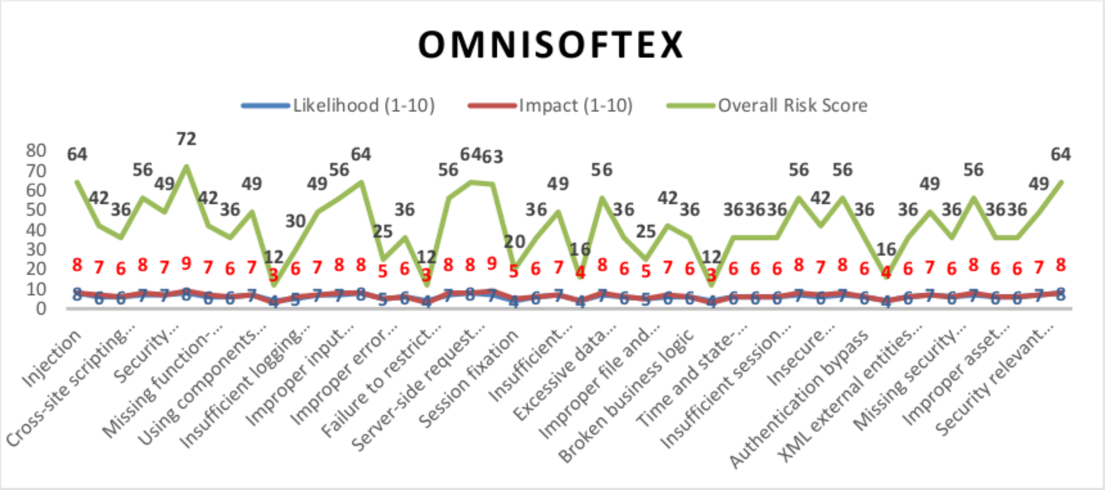
Based on the risk estimation provided by the protection poker technique, OmniSoftex prioritizes risks by considering their potential impact on the confidentiality, integrity, and availability of the application. High-risk areas are addressed with immediate attention, and resources are allocated accordingly for mitigation. The team continuously monitors and updates risk priorities to adapt to evolving threats.

Protection Techniques:

For the purpose of this questionnaire, OmniSoftex specifically utilized the protection poker technique for identifying and assessing the 46 security risks outlined in the Security Assurance Model. Their experience with this technique has been highly positive, as it fosters collaboration and consensus-building among team members. The structured approach of protection poker enables them to prioritize risks effectively and implement targeted mitigation strategies.

Challenges and Solutions:

In managing security risks, OmniSoftex faces challenges such as the complexity of integrating security into the development lifecycle and balancing security requirements with project timelines. To address these challenges, they have invested in automation tools, threat intelligence platforms, and continuous training for their team members. Additionally, fostering a culture of security awareness and accountability within their organization has been instrumental in mitigating risks effectively.



Response from Centangle Interactive:

Identifying Risks:

Centangle Interactive employs a proactive approach to identify the 46 security risks outlined in the security assurance model during the application development lifecycle. They conduct thorough risk assessments and security audits, leveraging their expertise in security best practices and industry standards. Additionally, they collaborate closely with clients to understand their security requirements and concerns.

Risk Management:

Centangle Interactive's risk management strategy focuses on implementing comprehensive security controls and protocols to mitigate identified risks effectively. They follow a defense-in-depth approach, combining technical controls, security policies, and user awareness training to safeguard their applications. Additionally, they conduct regular security testing and vulnerability assessments to identify and address potential vulnerabilities.

Risk Assessment:

At Centangle Interactive, risk assessment methods like the protection poker technique are integrated into the application development process. This technique allows them to systematically evaluate and prioritize risks based on their likelihood and impact. By involving stakeholders from different domains, including security, development, and business, they ensure a comprehensive understanding of potential threats.

Risk Prioritization:

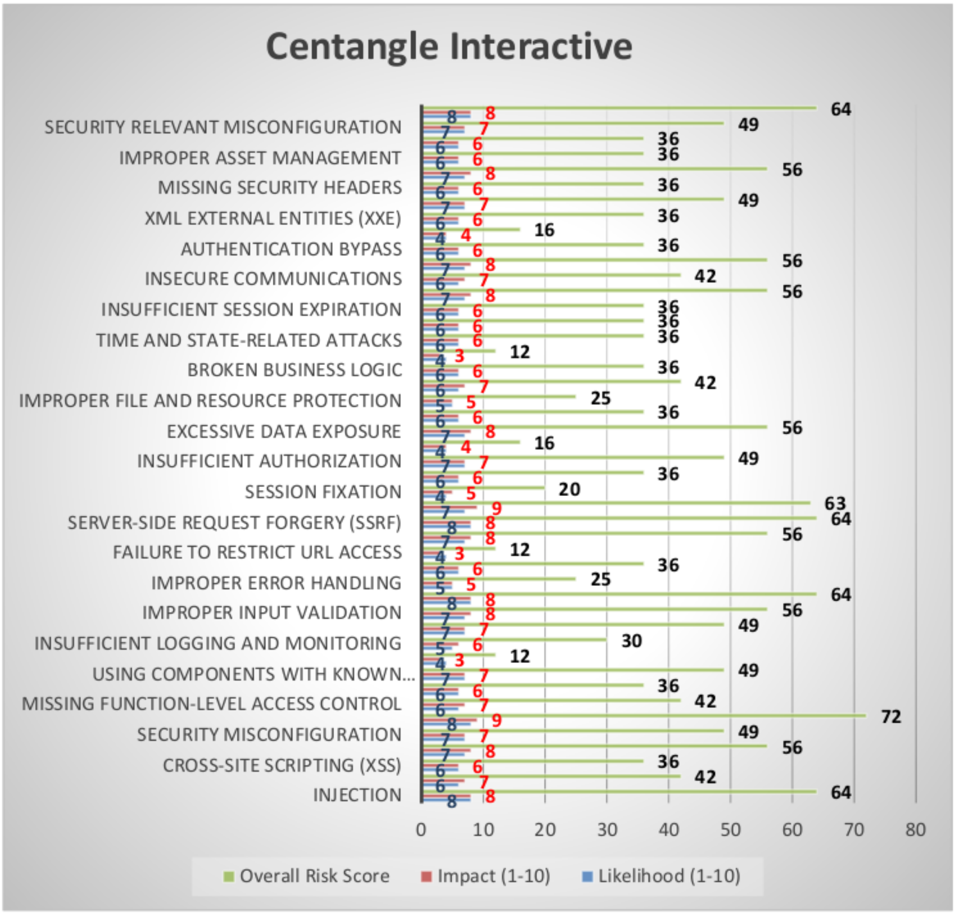
Based on the risk estimation provided by the protection poker technique, Centangle Interactive prioritizes risks by considering their potential impact on the confidentiality, integrity, and availability of the application. High-risk areas are addressed with immediate attention, and resources are allocated accordingly for mitigation. The team continuously reviews and updates risk priorities to adapt to changing threat landscapes.

Protection Techniques:

For the purpose of this questionnaire, Centangle Interactive specifically utilized the protection poker technique to assess and address the 46 security risks outlined in the Security Assurance Model. Their experience with this technique has been highly positive, as it fosters collaboration, consensus-building, and informed decision-making among team members. The structured approach of protection poker enables them to identify and prioritize risks effectively, leading to more robust security measures.

Challenges and Solutions:

In managing security risks, Centangle Interactive encounters challenges such as resource constraints, evolving threat landscapes, and compliance requirements. To address these challenges, they invest in automation tools, threat intelligence platforms, and continuous training for their team members. Additionally, fostering a culture of security awareness and accountability within their organization has been instrumental in mitigating risks effectively.



DISCUSSION AND ANALYSIS

Analysis of the response received from three different software houses (AppInSnap, OmniSoftex, and Centangle Interactive) show how they approach risk management, use protection poker, and prioritize risks based on likelihood and impact. Based on the provided risk assessment feedback, we can observe variations in the likelihood, impact, and overall risk scores assigned to each security risk. To convincingly declare the working of one of the software houses, let's analyze their risk assessment methodologies and outcomes:

AppInSnap:

- Total Risk Score: 2306
- Number of Risks: 46
- Average Risk Score: 50.13

OmniSoftex:

- Total Risk Score: 1917
- Number of Risks: 46
- Average Risk Score: 41.67

Centangle Interactive:

- Total Risk Score: 2332
- Number of Risks: 46
- Average Risk Score: 50.69

AppInSnap: AppInSnap's risk assessment methodology appears to focus on a balanced evaluation of likelihood and impact, resulting in moderate overall risk scores across most risk types. They prioritize risks such as Injection, Sensitive Data Exposure, and Buffer Overflow, which have high likelihoods and significant impacts, leading to higher overall risk scores. This approach suggests a methodical assessment process, where risks are carefully evaluated based on their potential to compromise application security and integrity.

OmniSoftex: OmniSoftex's risk assessment methodology seems to place more emphasis on the impact of risks rather than their likelihood. While they assign relatively high impact scores to risks like Sensitive Data Exposure and Injection, the likelihood scores for these risks are lower compared to other software houses. This approach indicates a focus on mitigating risks with significant potential impact, aligning with a strategy of prioritizing resources based on potential consequences rather than the probability of occurrence.

Centangle Interactive: Centangle Interactive's risk assessment methodology stands out for its comprehensive evaluation of both likelihood and impact, resulting in higher overall risk scores across several risk types. They prioritize risks such as Injection, Buffer Overflow, and Elevation of Privilege, which have high likelihoods and significant impacts, leading to elevated overall risk scores. This approach suggests a thorough risk assessment process, where risks are rigorously evaluated based on their potential to exploit vulnerabilities and compromise application security.

Conclusion:

Centangle Interactive's risk assessment methodology emerges as the most robust among the three software houses, evident from its higher average risk score. By meticulously evaluating both likelihood and impact, Centangle Interactive demonstrates a proactive approach to risk management, ensuring thorough scrutiny of potential threats to application security. Their methodology emphasizes the significance of identifying and prioritizing risks, facilitating the allocation of resources based on severity. Moving forward, future research could delve deeper into exploring the additional manpower or appointments required to effectively mitigate and resolve security risks throughout the development process would provide valuable insights into resource allocation strategies for enhancing application security.

References

[1]. H. Villamizar, A. Anderlin Neto, M. Kalinowski, A. Garcia and D. Méndez, "**An Approach for Reviewing Security-Related Aspects in Agile Requirements Specifications of Web Applications**," *IEEE 27th*

International Requirements Engineering Conference (RE), Jeju, Korea (South), pp. 86-97, 2019, doi: 10.1109/RE.2019.00020.

[2]. Martin Otieno, David Odera and Jairus Ekume Ounza **“Theory and practice in secure software development lifecycle: A comprehensive Survey”** , World Journal of Advanced Research and Reviews, 18(03), pp 53–78, 2023**.**

[3]. Wisdom Umeugo , Kimberly Lowrey and Shardul Y Pandya, “**Factors Affecting The Adoption Of Secure Software Practices In Small And Medium Enterprises That Build Software In-house**,” International Journal of Advanced Research in Computer Science, Volume 14(2), 2023..

[4]. R. A. Khan, S. U. Khan, M. Alzahrani and M. Ilyas, "**Security Assurance Model of Software Development for Global Software Development Vendors**," in IEEE Access, vol. 10, pp. 58458-58487, 2022, doi: 10.1109/ACCESS.2022.3178301.

[5] Mamdouh Alenezi and Sadiq Almuairfi “**Security Risks in the Software Development Lifecycle**” International Journal of Recent Technology and Engineering, Vol 8 (3), 2019, DOI:10.35940/ijrte.C5374.098319 .

[6]. W. Wang, Q. Zeng and A. P. Mathur, "**A Security Assurance Framework Combining Formal Verification and Security Functional Testing**," *12th International Conference on Quality Software*, Xi'an, China, pp. 136-139, doi: 10.1109/QSIC.2012.

[7] Samar Al-Saqqa, Samer Sawalha and Hiba AbdelNabi “**Agile Software Development: Methodologies and Trends**”, International Journal of Interactive Mobile Technologies, Vol. 14, No. 11, pp 246-270, 2020.

[8] K. Qian, R. M. Parizi and D. Lo, "**OWASP Risk Analysis Driven Security Requirements Specification for Secure Android Mobile Software Development**," *IEEE Conference on Dependable and Secure Computing (DSC)*, Kaohsiung, Taiwan, pp. 1-2, 2018 doi: 10.1109/DESEC.2018.8625114.

[9]. C. Onwubiko, "**Security operations centre: Situation awareness, threat intelligence and cybercrime**," *International Conference on Cyber Security And Protection Of Digital Services (Cyber Security)*, London, UK, pp. 1-6, 2017 doi: 10.1109/CyberSecPODS.2017.8074844.

[10] S. -J. Chen, Y. -C. Pan, Y. -W. Ma and C. -M. Chiang, "**The Impact of the Practical Security Test during the Software Development Lifecycle**," 2022 24th International Conference on Advanced Communication Technology (ICACT), PyeongChang Kwangwoon_Do, Korea, Republic of, 2022, pp. 313-316, doi: 10.23919/ICACT53585.2022.9728868.

[11] Y. Zeng, Y. Cheng, G. Xie and R. Wang, "**Design of Mobile Application Lifecycle Security Management Platform**," *2021 International Conference on Computer Network, Electronic and Automation (ICCNEA)*, Xi'an, China, 2021, pp. 26-30, doi: 10.1109/ICCNEA53019.2021.00017

[12] Al-Darwiash, A.I. and Choe, P. (2019), "**A framework of information security integrated with human factors", International Conference on Human-Computer Interaction (HCII)**, Springer, pp. 217-229. Arteaga, J.M., Gonzalez, R

[13] E. Khanna, R. Popli and N. Chauhan, "**Identification and Classification of Risk Factors in Distributed Agile Software Development**," in *Journal of Web Engineering*, vol. 21, no. 6, pp. 1831-1851, September 2022, doi: 10.13052/jwe1540-9589.2164.

[14] O. Kovalenko, O. Smirnov, A. Kovalenko and S. Kavun, "**Quantitative Risk Assessment Method Development in the Context of the SDLC-model**," *2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T)*, Kharkiv, Ukraine, 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143.

[15] Tøndel, I.A., Jaatun, M.G., Cruzes, D.S. and Williams, L. (2019), "**Collaborative security risk estimation in agile software**

development", *Information and Computer Security*, Vol. 27 No. 4, pp. 508-535. https://doi.org/10.1108/ICS-12-2018-0138.

R3 Abstract-Intro + LR + Meth+concl.docx

ORIGINALITY REPORT

18%	15%	8%	13%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	fastercapital.com Internet Source	3%
2	Submitted to University of Cincinnati Student Paper	2%
3	Submitted to University of Teesside Student Paper	1%
4	Submitted to MAHSA University Student Paper	1%
5	Submitted to Columbia Basin College Student Paper	1%
6	Jeffrey C. Carver, Birgit Penzenstadler, Leandro L. Minku, Ricardo Colomo-Palacios, Xabier Larrucea. "Conference Highlights: JIT Fault Prevention, Motivated Modeling, Security in Requirements, and Improving Team Performance", IEEE Software, 2020 Publication	1%
7	www.emerald.com Internet Source	1%

8	Submitted to University of Melbourne Student Paper	1 %
9	Submitted to ECPI College of Technology Student Paper	1 %
10	journals.riverpublishers.com Internet Source	1 %
11	Bilal Naqvi, Nathan Clarke, Jari Porras. "Incorporating the human facet of security in developing systems and services", Information & Computer Security, 2020 Publication	1 %
12	Submitted to Northcentral Student Paper	1 %
13	Submitted to University of Northumbria at Newcastle Student Paper	1 %
14	eprint.iacr.org Internet Source	1 %
15	ejurnal.seminar-id.com Internet Source	<1 %
16	aircconline.com Internet Source	<1 %
17	www.firstemployer.in Internet Source	<1 %
	Submitted to Purdue University	

18	Student Paper	<1 %
19	Submitted to Colorado State University, Global Campus Student Paper	<1 %
20	Submitted to Johns Hopkins Unversity Student Paper	<1 %
21	Submitted to Nelson and Colne College Student Paper	<1 %
22	Submitted to University of Johannsburg Student Paper	<1 %
23	Submitted to University of Sunderland Student Paper	<1 %
24	Submitted to University of Hertfordshire Student Paper	<1 %
25	do Nascimento, Gustavo Miguel Barroso Assis. "Anomaly Detection of Web-Based Attacks", Universidade de Lisboa (Portugal), 2024 Publication	<1 %
26	www.mdpi.com Internet Source	<1 %
27	9pdf.net Internet Source	<1 %
	open-innovation-projects.org	

28	Internet Source	<1 %
29	www.scilit.net Internet Source	<1 %
30	Yolanda Valdés-Rodríguez, Jorge Hochstetter-Diez, Jaime Díaz-Arancibia, Rodrigo Cadena-Martínez. "Towards the Integration of Security Practices in Agile Software Development: A Systematic Mapping Review", Applied Sciences, 2023 Publication	<1 %
31	Alexey S. Markov, Vitaliy V. Varenitca, Sas S. Arustamyan. "Issues in the Implementation of Secure Software Development Processes", 2023 Seminar on Information Systems Theory and Practice (ISTP), 2023 Publication	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		