21-L-7512 Abdullah Dar BSCS-6A

Cyber Security

Assignment no: 2

Software Security

Section no 1: Introduction to Software Security

Software security, in the context of cybersecurity, refers to the practice of implementing measures to protect software systems, applications, and data from various security threats and vulnerabilities. It is a crucial aspect of cybersecurity, as software vulnerabilities can be exploited by attackers to gain unauthorized access, manipulate data, disrupt operations, or carry out other malicious activities. The goal of software security is to ensure the confidentiality, integrity, and availability of software and its associated data.

Important components of secure software comprise:

- **Safe Coding Procedures:** Ensuring that software developers follow secure coding practices is fundamental to software security. This involves writing code in a way that minimizes vulnerabilities and reduces the risk of exploitation.
- **Vulnerability Evaluation:** Regularly scanning and assessing software applications for vulnerabilities is essential. This process involves identifying and addressing potential weaknesses that could be exploited by attackers.
- **Testing for Penetration:** Penetration testing involves simulating cyberattacks to identify and exploit vulnerabilities in a controlled environment. This helps organizations understand potential risks and weaknesses in their software.
- **Authentication and Authorization:** Implementing strong authentication mechanisms and proper authorization controls ensures that only authorized users have access to the software and its data.
- **Encryption:** Employing encryption techniques to protect sensitive data both in transit and at rest helps prevent unauthorized access even if the data is intercepted.
- **Secure Development Lifecycle (SDLC):** Incorporating security into the software development lifecycle from the initial design phase through testing and deployment helps identify and mitigate security issues early in the development process.
- **Incident Response Planning:** Developing and implementing incident response plans ensures that organizations can effectively respond to and recover from security incidents, minimizing the impact of potential breaches.
- **Security Training and Awareness:** Educating software developers, administrators, and end-users about security best practices helps create a security-aware culture and reduces the likelihood of security breaches.

Conclusion:

Overall, software security is a comprehensive approach that involves a combination of technical, procedural, and human-centric measures to protect software systems and data from cyber threats. It plays a critical role in maintaining the overall cybersecurity posture of an organization.

Section no 2: Selected Research Papers

Reference No.	Publication Year	Journal/Conference	Citation Count	Paper Link
1	2020	Computers & Security	175	Link
2	2023	Migration Letters	33	Link
3	2020	Journal of Systems and Software	43	Link
4	2021	9th International Conference in Software Engineering Research and Innovation (CONISOFT)	8	Link
5	2021	International Electronics Symposium (IES)	13	Link
6	2020	Arabian Journal for Science and Engineering	258	Link
7	2020	IEEE Access Vol. 8	141	Link
8	2021	Journal of Network and Computer Applications	111	Link
9	2021	ACM Journals	140	Link
10	2020	Empirical Software Engineering	74	Link

Section no 3: Overview of One Article

Title: Web API Security Vulnerabilities and Mitigation Mechanisms

Summary: -

This research paper, published in the Journal of Systems and Software in 2020, addresses the evolving landscape of web-based software systems due to the growth of the web in recent decades. The proliferation of such systems has led to the adoption of web APIs for communication between distributed software entities. However, the integration of web APIs has introduced new challenges, particularly the imperative to secure APIs at a design level against potential malicious attacks.

The paper emphasizes the non-trivial nature of securing APIs by design and underscores the importance of understanding common vulnerabilities and the corresponding defense mechanisms. To systematically gather existing scientific knowledge on security threats faced by web APIs, as well as design-level mechanisms for detecting, resisting, reacting, and recovering from attacks, the researchers conducted a mapping study.

The study identified a total of 66 threats documented in the literature, with a notable focus on Spoofing and Tampering threats, primarily associated with network traffic interactions. Conversely, threats related to repudiation were found to be the least reported. The researchers further discovered 21 techniques, 11 patterns, and 34 methods that can be employed at a design level to address these threats effectively.

In summary, the research contributes valuable insights into the security challenges inherent in web APIs and provides a comprehensive overview of the threats they face, along with design-level strategies to mitigate these challenges. This information is crucial for developers, security professionals, and organizations aiming to enhance the security posture of their web-based software systems.

Section no 4: References (APA Style)

- ✓ Hart, S., Margheri, A., Paci, F., & Sassone, V. (2020). RisKiO: a serious game for cyber security awareness and education. *Computers & Security*, *95*, 101827. https://doi.org/10.1016/j.cose.2020.101827
- ✓ Shwedeh, F. (2023). The Moderation Effect of Artificial Intelligent Hackers on the Relationship between Cyber Security Conducts and the Sustainability of Software Protection: A Comprehensive Review. migrationletters.com. https://doi.org/10.59670/ml.v20iS9.4947
- ✓ Gkortzis, A., Feitosa, D., & Spinellis, D. (2021). Software reuse cuts both ways: An empirical analysis of its relationship with security vulnerabilities. Journal of Systems and Software, 172, 110653. https://doi.org/10.1016/j.jss.2020.110653
- ✓ Web API Security Vulnerabilities and Mitigation Mechanisms: A Systematic Mapping study. (n.d.). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/9653437
- ✓ Development of vulnerable web application based on OWASP API security risks. (n.d.). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/9593934
- ✓ Humayun, M., Niazi, M., Jhanjhi, N. Z., Alshayeb, M., & Mahmood, S. (2020). Cyber Security Threats and Vulnerabilities: A Systematic Mapping study. Arabian Journal for Science and Engineering, 45(4), 3171–3189. https://doi.org/10.1007/s13369-019-04319-2
- ✓ KeySplitWatermark: Zero Watermarking Algorithm for software protection against Cyber-Attacks. (n.d.). IEEE Journals & Magazine | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/9068217
- ✓ Ahuja, N., Singal, G., Mukhopadhyay, D., & Kumar, N. (2021). Automated DDOS attack detection in software defined networking. Journal of Network and Computer Applications, 187, 103108. https://doi.org/10.1016/j.jnca.2021.103108
- ✓ Cheng, X., Wang, H., Hua, J., Xu, G., & Sui, Y. (2021). DeepWukong. ACM Transactions on Software Engineering and Methodology, 30(3), 1–33. https://doi.org/10.1145/3436877
- ✓ Ponta, S. E., Plate, H., & Sabetta, A. (2020). Detection, assessment and mitigation of vulnerabilities in open source dependencies. Empirical Software Engineering, 25(5), 3175–3215. https://doi.org/10.1007/s10664-020-09830-x