


Summary Post

◀ Summary Post

Initial Post ▶

Display replies in nested form

Settings ▼



Summary Post

by [Lauren Pechey](#) - Sunday, 18 August 2024, 7:55 AM

This post serves to summarise the various inputs given by fellow students in the collaborative discussion forum, regarding the top 10 key vulnerabilities identified in the OWASP (2021) in relation to secure security programming.

It has been highlighted that software security is crucial in the Software Development Lifecycle (SDLC), especially as personal data is increasingly shared across various devices (Gore, 2024). Pillai (2017) defines security as a system’s ability to prevent unauthorised access while providing services to authenticated users. The Open Web Application Security Project (OWASP), along with the posts of numerous students, identifies broken access control as the number one security risk for web applications (Pechey, 2024; Brayshaw, 2024). Mitigating this risk requires secure coding strategies, such as the principle of least privilege, where processes are run with minimal permissions (Jackson, 2024).

Injection attacks, ranked third by OWASP and emphasised in the discussion forum, also poses significant risks, including data theft and malicious software installation (Gore, 2024). Prevention involves filtering inputs, restricting access, and maintaining databases (Hird, 2024). Additionally, cryptographic failures are a major concern, as weak encryption systems or poor implementation can expose sensitive data (Wong, 2024). Continuous testing and monitoring are crucial to maintaining security.

Finally, it was emphasised that the traditional waterfall model focuses on design before coding, which can make it difficult to address emerging threats. Therefore, agile development offers more flexibility through iterative processes, but its speed can introduce risks, making secure Scrum practices essential (Marakgos, 2024).

Overall, effective software security involves secure coding, a strong architectural foundation, and ongoing threat assessment throughout the SDLC.

References:

Brayshaw, T. (2024) Initial Post. SSD July 2024. Forum post submitted to the University of Essex Online.

Gore, S. (2024) Initial Post. SSD July 2024. Forum post submitted to the University of Essex Online.

Hird, G. (2024) Initial Post. SSD July 2024. Forum post submitted to the University of Essex Online.

Jackson, K. (2024) Initial Post. SSD July 2024. Forum post submitted to the University of Essex Online.

Marakgos, I. (2024) Initial Post. SSD July 2024. Forum post submitted to the University of Essex Online.

OWASP (2021) OWASP Top Ten. Available from: https://owasp.org/Top10/A03_2021-Injection/ [Accessed 17 August 2024].

Pechey, L. (2024) Initial Post. SSD July 2024. Forum post submitted to the University of Essex Online.

Pillai, A.B. (2017) *Software architecture with Python: Design and architect highly scalable, robust, clean, and high performance applications in Python*. 1st ed. Birmingham: Packt Publishing.

Wong, C. (2024) Initial Post. SSD July 2024. Forum post submitted to the University of Essex Online.

Maximum rating: -

Permalink

Reply

◀ Summary Post

Initial Post ▶

You are logged in as Lauren Pechey (Log out)

Policies

Powered by Moodle

