OWASP #4 – Insecure Design

Gavin Lillard (IST-103)

Insecure design is one of the modern leading causes of web applications’ integrity declines. According to QAwerk (2024), potential attackers can seek weaknesses in the web applications’ architecture which said weaknesses would allow for “the bypass of authentication mechanisms, modification of certain URL parameters through unauthorized channels, access to systems to mine for sensitive information, assume legitimate user accounts and gain unauthorized access to protected resources, obtain application wide access and further extend the attack to other environments, overload servers in the attempt to crash them, extract information with directed queries, user-account takeover, and other attacks such as XSS, injections, SSRF, and path transversal.

An example would be a website that allows for mass reservation without deposit or credit card information. This would lead to a loss of income due to “fake” reservations. Another example would be NVIDIA’s e-commerce site neglects bots that are used to scalp high-end products. They will lose customers due to lack of product which is assured by lack of security against bot distinguishment.

Insecure designs cannot depend on perfect implementation, instead they rely on security controls. QAwerk (2024) states mitigation can be achieved by moving from DevOps to DevSecOps, critical authentication, secure development procedures, use a standard library that contains secure designs, integrate security language into user stories, configure bot signatures to distinguish against good and bad bots, filter all output thoroughly, and limit user resource consumption.

Works Cited

“Insecure Design Vulnerability: Explanation and Examples.” *QAwerk*, qawerk.com/blog/insecure-design/.