OWASP #10 – Server-Side Request Forgery

Gavin Lillard (IST-103)

Gorbe (2024) summarizes the essence of this vulnerability by stating, in my own words, SSRF flaws ensue due to web applications prematurely fetching a remote resource without the validation of the user-supplied URL. This means the potential attacker can tell the web application to send a manipulated request to any destination despite a firewall, VPN or any other ACL. Unfortunately, SSRF incidents are increasing due to the implementation of Quality-of-Life features, such as the fetching of URLs. “The severity of SSRF is becoming higher due to cloud services and the complexity of architectures” (Gorbe, 2024). Sadly, as modern technology caters to the average-joe, these vulnerabilities will become more and more frequent and require constant patching at a mass scale.

Examples provided by Gorbe (2024) include port scan internal servers, which is just unsegmented architectures, allows potential attackers to track the availability of ports at an internal level. Another example would be sensitive data exposure allowing potential attackers to access local/internal information.

The mitigation of this vulnerability is split up into both the network and application layers. Starting with the network layer, Gorbe (2024) states you should segment remote resource access in separate networks and enforce “deny by default” policies. At the application layer, Gorbe (2024) suggests that you validate user input data, secure the integrity of the URL schema, port and destination with a positive allow list, never send raw responses to users, disable HTTP redirections, and monitor consistency of URLs. Mitigation can be achieved by staying attentive to security-to-functionality processes.

Works Cited

Gorbe, Daniel. “A10:2021 – Server-Side Request Forgery (SSRF).” *G⌬RBE*, 10 Oct. 2024, gorbe.io/posts/owasp/top-10/server-side-request-forgery/. Accessed 23 Apr. 2025.