OWASP #6 – Vulnerable and Outdated Components

Gavin Lillard (IST-103)

Vulnerable and outdated components consist of mainly legacy libraries, frameworks, or external software. Usually, these components are no longer maintained which allows potential attackers to have more time to figure out ways to infiltrate these components. A few examples of these components consist of open-source libraries such as Java, Python, PHP or JavaScript, middleware and server plugins, APIs and service connectors, and legacy dependencies baked into business logic.

Stated by Neubert (2025), a few real-world examples include using a vulnerable version of Log4j, running outdated jQuery libraries know to be affected by XSS flaws, still using legacy WordPress or Drupal plugins with insecure designs, and leaving APIs exposed.

Key risks of using vulnerable and outdated components are exploitability, automation, unauthorized access, compliance violations, cascading failures, and persistent threats. Being proactive usually helps solve these problems in most cases. You should maintain a full SBOM, use SCA tools for visibility, adopt a DAST-first approach, automate patching, integrate security into CI/CD, and apply virtual patching via WAF. Being consistent with your security details will help you maintain a secure web application. Neubert (2025) states that, “By maintaining a clear inventory, automating your patch management process, and focusing on dynamic, real-world exposure through DAST, you can protect against data breaches, unauthorized access, and other security risks without sacrificing functionality or velocity”.

Works Cited

Neubert, Jesse. “Vulnerable and Outdated Components: An OWASP Top 10 Risk.” *Acunetix*, 10 Apr. 2025, www.acunetix.com/blog/web-security-zone/vulnerable-and-outdated-components-owasp-top-10/. Accessed 23 Apr. 2025.