**Journal 8-2: Portfolio Reflection**

A security policy helps an organization define core security principles guiding coding, authorization, authentication, auditing, and encryption standards. Adopting secure coding standards helps the organization’s stakeholders to agree on the implementation of code in a particular programming language in ways that harmonize with the organization’s security policy. Helping the organization to assess risks attached to each standard and how automation can be used to detect and mitigate non-compliant code solutions.

The adoption of a security policy containing secure coding standards lends itself easily to the practice of not leaving security to the end. Every stage of the project from preproduction to production must implement security measures as layers to build on in subsequent stages. Not leaving security to the end also involves taking proactive steps on the nature and type of data to collect, how long to keep it, and who can access it, to mitigate the possibility of future security threats and data breaches.

Being able to evaluate and assess the risk using a threat matrix comparing severity and priority helps an organization to carefully consider the remediation costs and how likely the risk is. Rules guiding codes with high severity and priority levels must be thoroughly monitored and enforced to avoid implications in cost and operations.

Zero trust involves the verification of all users, data, devices, and networks accessing a system irrespective of whether they are within the organization's security perimeter or not. This further extends the principle of least privilege protecting critical system assets and data from unauthorized use. Developers in the organization must trained to appreciate the value of zero trust despite its supposed distrust stigma and sometimes intrusive authentication methods.

An effective security policy uses defense in depth based on the Triple-A Framework (Authentication, Authorization, and Accounting) and is enforced. The elements of the policy must be regularly presented to train the adopting organization’s developers. It can also be used to guide the performance of policy compliance audits and to determine how to add security automation to an organization's existing DevOps pipeline.