Delaying the implementation of a comprehensive security policy increases the likelihood of breaches, which can lead to data loss, reputational damage, and financial penalties. Implementing security measures from the outset strengthens the overall security posture, reduces the risk of attacks, and ensures compliance with regulatory requirements.

One of the essential steps in maintaining secure code is adopting a secure coding standard. This involves implementing guidelines and best practices for writing secure code from the beginning of the development process, rather than addressing security as an afterthought. By following a standardized approach, developers can minimize vulnerabilities and prevent common security issues.

After code is developed, static analysis can be used to identify potential issues. This form of testing helps in assessing risks, which can vary in terms of likelihood and severity. Ideally, risks that are both highly likely and severe should be prioritized for immediate mitigation. For example, SQL injection vulnerabilities are both common and potentially catastrophic, so extra precautions should be taken to prevent them.

The concept of 'zero trust' is a security model that involves not assuming trust in any application, device, or user. Instead, all entities must be authenticated, authorized, and continuously verified before they can access resources. This approach helps minimize the risk of insider threats and unauthorized access.

Implementing security policies before development begins is crucial for addressing common coding issues proactively. These policies establish clear guidelines for developers to follow, ensuring that security is built into the code from the start. More detailed coding standards can be found in resources such as the SEI CERT C++ Coding Standard.