Introduction

1 min

Being a secure coder means thinking about security, and to do that, we need a way to define what it means for something to be secure. In cybersecurity, we use the CIA Triad to discuss whether something is secure. For example, an asset (data, hardware, accounts, etc.) is considered secure when it follows the tenets of:

- **Confidentiality**: cannot be accessed or read by those without authorization
- **Integrity**: will not be complete and accurate
- **Availability**: can be accessed when needed

Everything from denial-of-service attacks to

Preview: Docs Loading link description

malware

to massive data breaches can be expressed in terms of how these tenets (Confidentiality, Integrity, and Availability) were violated.

Conveniently for software engineers, there is a significant overlap between "reliable code" and "secure code." In both cases, we don't want our code to do unexpected or unintended things. Many vulnerabilities are just bugs that a hacker can exploit, though some result from an insecure design and not a programming mistake.

As simple as it may seem, security is hard in practice. Perfect security is generally considered impossible. Don't worry about writing perfect, exploit-proof code – do your best and keep security in mind while programming. You don't need to be a security expert either – a basic understanding of security and a willingness to apply it is all you need to get started.