Unit Test Development using Generative AI

Security Test Pyramid

1 min

Security considerations are sometimes seen as obstacles to software development, but this isn't necessarily true. The techniques that are used to create reliable software are also used to create secure software. Security and reliability are often parallel goals, and improving one will positively impact the other.

Software cannot be secure without reliability; reliability does not guarantee security, but it is an essential prerequisite. Software exploits and software bugs are alike because they both involve a program entering an invalid state (i.e., the program starts behaving in unintended ways).

The goal of software testing is to prevent programs from being able to enter these invalid states. The test pyramid is a concept used in software development (and cybersecurity, with some modifications) consisting of multiple layers of tests, increasing in complexity as you ascend the pyramid. Unit testing forms the foundation of this pyramid.

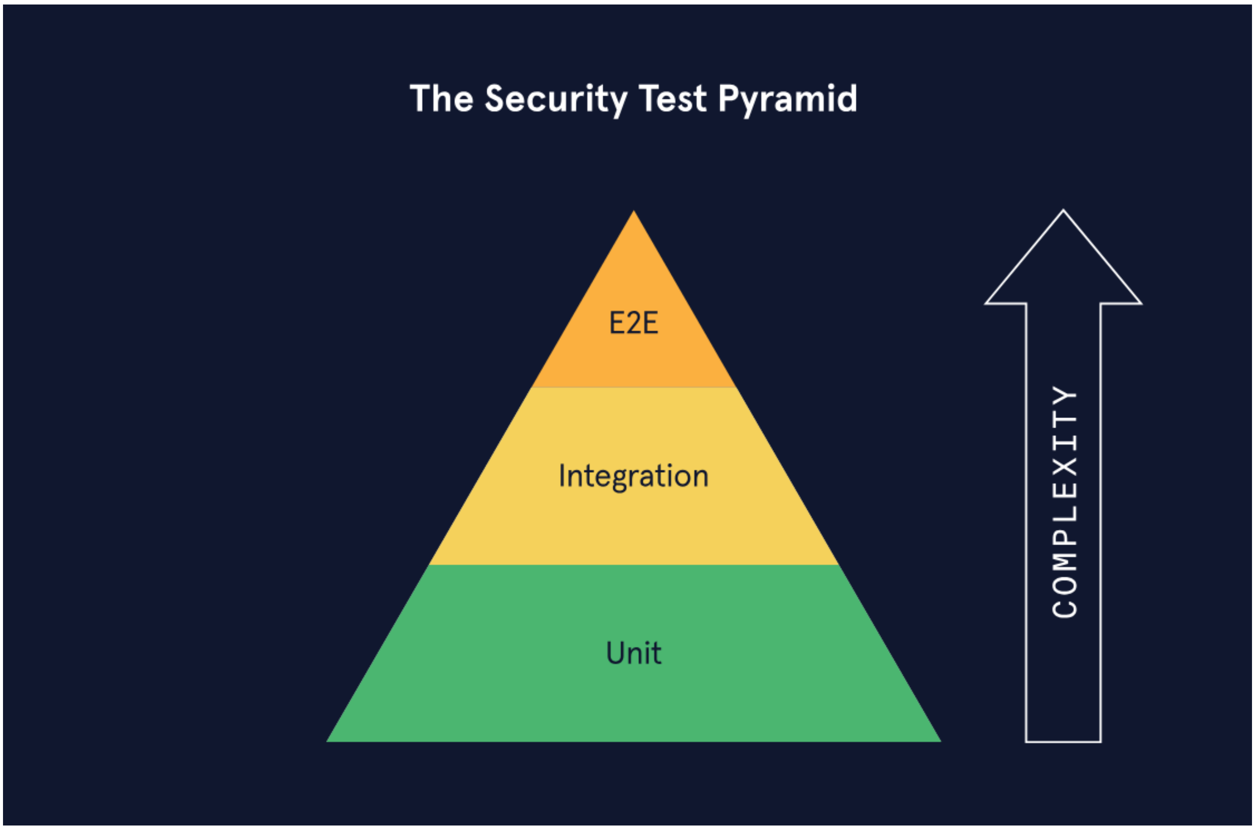
A unit in unit testing refers to a piece of a program that can operate in isolation and cannot meaningfully be broken down further. Units are often classes, objects, or even individual functions. Unit testing involves running tests on these individual units, ensuring they operate correctly in isolation.

Programs tend to have a lot of units, and often, there will be more than one test per unit, so creating these tests can take a lot of time and repetitive work. We'd like to have the AI handle the boring stuff so we, as programmers, can spend more time developing good tests.

Instructions

Take a look at the diagram of the Security Test Pyramid. Each type of test increases in complexity as it moves up the pyramid. In this lesson, we will focus on using AI to help create unit tests.

When ready, click **Next** to go to the next exercise.


The Security Test Pyramid
E2E
Integration
Unit
COMPLEXITY