# Gray Box Testing

Gray box testing (a.k.a grey box testing) is a method you can use to debug software and evaluate vulnerabilities. In this method, the tester has limited knowledge of the workings of the component being tested. This is in contrast to [**black box testing**](https://www.imperva.com/learn/application-security/black-box-testing/)**,** where the tester has no internal knowledge, and [**white box testing**](https://www.imperva.com/learn/application-security/white-box-testing/)**,** where the tester has full internal knowledge.

Gray box testing is most effective for evaluating web applications, integration testing, distributed environments, business domain testing, and performing security assessments. When performing this testing, you should create clear distinctions between testers and developers to ensure test results aren’t biased by internal knowledge.

**The Gray Box Testing Process**

In gray box testing, the tester is not required to design test cases. Instead, test cases are created based on algorithms that evaluate internal states, program behavior, and application architecture knowledge.

When performing gray box testing, take the following steps:

* Identify and select Inputs from white and black box testing methods.
* Identify probable outputs from these inputs.
* Identify key paths for the testing phase.
* Identify sub-functions for deep-level testing.
* Identify inputs for sub-functions.
* Identify probable outputs from sub-functions.
* Execute sub-function test cases.
* Assess and verify outcomes.
* Repeat steps 4-8.
* Repeat steps 7 and 8.

## Why Gray Box Testing

* It provides combined benefits of both black box testing and white box testing both
* It combines the input of developers as well as testers and improves overall product quality
* It reduces the overhead of long process of testing functional and non-functional types
* It gives enough free time for a developer to fix defects
* Testing is done from the user point of view rather than a designer point of view

## Gray Box Testing Techniques

Gray box testing techniques are designed to enable you to perform penetration testing on your applications. These techniques enable you to test for [insider threats](https://www.imperva.com/learn/application-security/insider-threats/), such as employees attempting to manipulate applications, and external users, such as attackers attempting to exploit [vulnerabilities](https://www.imperva.com/learn/application-security/cve-cvss-vulnerability/).

* **Matrix Testing:**This testing technique involves defining all the variables that exist in their programs.
* **Regression Testing**: To check whether the change in the previous version has regressed other aspects of the program in the new version. It will be done by testing strategies like retest all, retest risky use cases, retest within a firewall.
* **Orthogonal Array Testing or OAT**: It provides maximum code coverage with minimum test cases.
* **Pattern Testing:** This testing is performed on the historical data of the previous system defects. Unlike black box testing, gray box testing digs within the code and determines why the failure happened.

## Gray Box Testing Challenges

* When a component under test encounter a failure of some kind may lead to abortion of the ongoing operation
* When test executes in full but the content of the result is incorrect.