

Professional Skills
Agile Fundamentals
Jira
Git
Databases Introduction
Java Beginner
Maven
Testing (Foundation)
Java Intermediate
HTML
CSS
Javascript
Spring Boot
Selenium
Sonarqube
Advanced Testing (Theory)
Cucumber
MongoDB
Express
NodeJS
React
Express-Testing
Networking
Security
<div><div></div>Security Basics</div>
<div><div></div>Authentication and Authorization</div>
<div><div></div>Security Attacks</div>
<div><div></div>Passwords</div>
<div><div></div>Hashing vs Encrypting vs Encoding</div>
<div><div></div>Introduction to DevSecOps</div>
<div><div></div>Pen Testing, DAST &amp; SAST</div>

# Pen Testing, DAST & SAST

## Contents

- [Overview](#)
- [Penetration Testing](#)
- [Application Security Testing](#)
- [DAST](#)
  - [Tools](#)
- [SAST](#)
  - [Tools](#)
- [Key Benefits & Differences](#)
- [Tutorial](#)
- [Exercises](#)

## Overview

In this module, we will look at **Penetration Testing**, **Dynamic Application Security Testing (DAST)** and **Static Application Security Testing (SAST)**.

## Penetration Testing

Penetration Testing, also known as *Pen Testing* or *Ethical Hacking*, is the practice of testing a computer system, network or web application to find security vulnerabilities that an attacker could exploit.

This can be *automated*, with software applications, or performed *manually*.

The process involves:

- gathering information** about the target before the test
- identifying** possible entry points
- attempting to **break in** (either virtually or for real)
- reporting** back the findings.

## Application Security Testing

Past high-profile data breaches have forced organisations to be more concerned about the financial and business consequences of having their data stolen.

They know that they need to identify vulnerabilities in their applications and mitigate the risks as soon as possible.

To do this, they are adding **application security testing**, including *DAST* and *SAST*, to their software development workflows.

DAST and SAST are application security testing methodologies used to **find security vulnerabilities that can make an application susceptible to attack**.


## DAST

Dynamic **A**pplication **S**ecurity **T**esting (DAST) is a **black box** testing method that **examines an application as it's running** to find vulnerabilities that an attacker could exploit.

## Tools

Some of the more popular DAST tools are:

- [Acunetix Vulnerability Scanner](#)

 Introduction to OWASP
Cloud Fundamentals
AWS Foundations
AWS Intermediate
Linux
DevOps
Jenkins Introduction
Jenkins Pipeline
Markdown
IDE Cheatsheet

- [GitLab](#)
- [Appknox](#)
- [Netsparker](#)
- [CheckMarx](#)

Most of these tools also offer SAST!

## SAST

Static Application Security Testing (SAST) is a **white box** method of testing. It examines the code to find software flaws and weaknesses (such as SQL injection).

### Tools

Some of the more popular SAST tools (not mentioned above) are:

- [Coverity](#)
- [HCL AppScan](#)
- [Kiuwan](#)
- [AttackFlow](#)
- [CoreOS Clair](#)

## Key Benefits & Differences

DAST	SAST
<b>Black box security testing:</b> the tester has no knowledge of the technologies or frameworks that the application is built on. The application is tested from the outside in. This type of testing represents the <i>hacker</i> approach	<b>White box security testing:</b> The tester has access to the underlying framework, design, and implementation. The application is tested from the inside out. This type of testing represents the <i>developer</i> approach.
<b>Requires a running application:</b> DAST doesn't require source code. It analyses by executing the application.	<b>Requires source code:</b> SAST doesn't require a deployed application. It analyses the source code without executing the application.
<b>Finds vulnerabilities later:</b> Vulnerabilities discovered after the development cycle is complete and the application is running.	<b>Finds vulnerabilities earlier:</b> The scan can be executed as soon as code is deemed feature-complete.
<b>More expensive to fix vulnerabilities:</b> Since vulnerabilities are found towards the end of the software development lifecycle, remediation often gets pushed into the next cycle.	<b>Less expensive to fix vulnerabilities:</b> Since vulnerabilities are found earlier, it's easier and faster to fix them.
<b>Can discover run-time and environment-related issues,</b> as it analyses running applications.	<b>Can't discover run-time and environment-related issues,</b> as it analyses static code.

Please Note: DAST and SAST techniques complement each other. However, both need to be carried out for comprehensive testing.

## Tutorial

There is no tutorial for this module.

## Exercises

There are no exercises for this module.