

**Web Application Security in Python**

**Course Number:** SEC-154TP
**Duration:** 3 days

**Overview**

This Web App Security in Python course guides developers through secure design principles and the OWASP Top Ten vulnerabilities and teaches them how to defend against common attacks like injection and cross-site scripting. Participants also learn the crucial cryptography, authentication, and server configuration aspects. By the end of this course, developers can build and maintain secure, resilient web applications using Python.

**Prerequisites**

All Python Security training students must have general Python development experience.

**Materials**

All attendees receive comprehensive courseware.

**Software Needed on Each Student PC**

Attendees will not need to install any software on their computers for this class. The class will be conducted in a remote environment that Accelebrate will provide; students will only need a local computer with a web browser and a stable Internet connection. Any recent version of Microsoft Edge, Mozilla Firefox, or Google Chrome will work well.

**Objectives**

* Explain core security concepts and the OWASP Top Ten vulnerabilities
* Prevent broken access control by implementing proper authorization and mitigating file upload risks
* Apply cryptography to ensure data confidentiality and integrity
* Defend against injection attacks by validating input and utilizing parameterized queries
* Design secure applications using the STRIDE model and Saltzer and Schroeder's principles
* Correctly configure servers, cookies, and XML to prevent security misconfigurations
* Manage vulnerable components and implement authentication best practices

**Outline**

* Introduction to Cyber Security
	+ What is security?
	+ Threat and risk
	+ Cyber security threat types – the CIA triad
	+ Consequences of insecure software
* The OWASP Top Ten
	+ Broken Access Control (A01)
		- Access control basics
		- Failure to restrict URL access
		- Confused deputy
		- File upload
	+ Cryptographic Failures (A02)
		- Cryptography for developers
	+ Injection (A03)
		- Injection principles
		- Injection attacks
		- SQL injection
		- Code injection
		- HTML injection - Cross-site scripting (XSS)
	+ Insecure Design (A04)
		- The STRIDE model of threats
		- Secure design principles of Saltzer and Schroeder
		- Client-side security
	+ Security Misconfiguration (A05)
		- Configuration principles
		- Server misconfiguration
		- Python configuration best practices
		- Cookie security
		- XML entities
	+ Vulnerable and Outdated Components (A06)
		- Using vulnerable components
		- Assessing the environment
		- Hardening
		- Untrusted functionality import
		- Malicious packages in Python
		- Vulnerability management
	+ Identification and Authentication Failures (A07)
		- Authentication
		- Password management
	+ Software and Data Integrity Failures (A08)
		- Integrity protection
		- Subresource integrity
	+ Server-Side Request Forgery (A10)
		- Server-side Request Forgery (SSRF)
* Wrap Up
	+ Secure coding principles
	+ Software security sources and further reading
	+ Python resources
* Conclusion