

Desktop Application Security in Python

Download Whitepaper: Accelerate Your Modernization Efforts with a Cloud-Native Strategy
Get Your Free Copy Now

Course Number: CYD-APPSECPYTH

Duration: 3 days

Overview

Course Description

This Desktop Application Security in Python course teaches developers basic and advanced skills for building secure desktop applications using Python. Participants learn how to identify and mitigate threats, implement secure coding practices, and leverage cryptography to safeguard sensitive data. By the end of the course, learners can create robust and secure desktop applications that protect against cyberattacks.

Skills Gained

- Understand fundamental cyber security concepts and threats
- Apply input validation techniques to mitigate injection attacks
- Identify and prevent integer handling vulnerabilities
- Securely handle files and streams to avoid path traversal attacks
- Implement robust authentication and password management strategies
- Utilize cryptography to protect data confidentiality and integrity

- Recognize and address common software security weaknesses related to time, state, and errors

Prerequisites

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

Audience

Course Details

Introduction to Cyber Security

- What is security?
- Threat and risk
- Cyber security threat types – the CIA triad
- Cyber security threat types – the STRIDE model
- Consequences of insecure software

Input Validation

- Input validation principles
- Denylists and allowlists
- What to validate – the attack surface
- Where to validate – defense in depth
- When to validate – validation vs transformations
- Validation with regex
- Regular expression denial of service (ReDoS)
- Dealing with ReDoS

Injection

- Injection principles
- Injection attacks
- SQL injection
- Code injection

Integer Handling Problems

- Representing signed numbers
- Integer visualization
- Integers in Python
- Integer overflow

- Integer overflows in ctypes and numpy

Files and Streams

- Path traversal
- Additional challenges in Windows
- Path traversal best practices

Security Features

- Authentication
- Password management
- Information exposure

Platform Security

- Python platform security

Using Vulnerable Components

- Assessing the environment
- Hardening
- Malicious packages in Python
- Vulnerability management

Cryptography for Developers

- Cryptography basics
- Cryptography in Python
- Elementary algorithms
- Confidentiality protection
- Integrity protection
- Public Key Infrastructure (PKI)

Time and State

- Race conditions

Errors

- Error and exception handling principles
- Exception handling

Wrap Up

- Secure coding principles

- And now what?

Conclusion