CSP104 – SECURE SOFTWARE CODING

Course Learning Objectives

Explain the fundamentals of programming and different software development methodologies. Identify common software attacks and vulnerabilities. Describe defensive coding practices and controls. Implement programming safeguards using defensive coding principles. Explain the difference between static and dynamic code analysis. Describe how to build software with security mechanisms in place.

Description

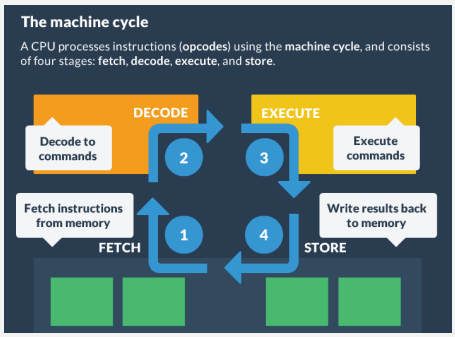
The Security Software Implementation/Coding domain will provide the learner with an understanding the importance of programming concepts that can effectively protect software from vulnerabilities. Learners will touch on topics such as software coding vulnerabilities, defensive coding techniques and processes, code analysis and protection, and environmental security considerations that should be factored into software.

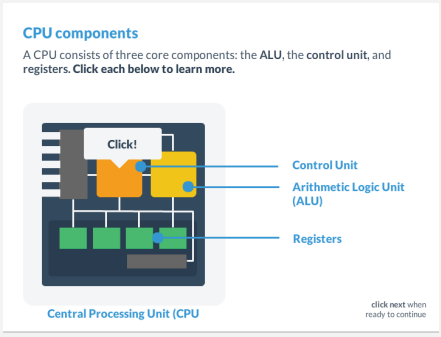
Audience

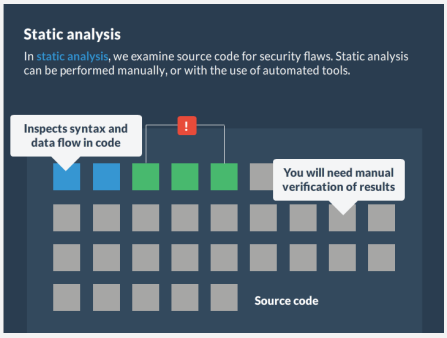
Certified Secure Software Lifecycle Professional (CSSLP)

Time Required

Tailored learning - 40 minutes total







CSP104 – SECURE SOFTWARE CODING

Course Outline

1. Programming Languages

- Computer architecture
- CPU components
- The machine cycle
- Internal memory
- Stack PUSH POP operations
- About programming languages
- Abstraction
- Compiled languages
- Interpreted and hybrid languages
- Managed vs. unmanaged languages
- Code access security
- .NET security transparency

2. Common Software Vulnerabilities

- Industry databases
- About
- Desktop software vulnerabilities
- Locality of reference
- About buffer overflow
- Attacker injects malicious code
- Attacker executes own code
- Dangling pointers
- Code obfuscation
- ASLR
- Data execution protection

3. Secure Software Processes

- Code analysis
- Static analysis
- Dynamic analysis
- Static vs. Dynamic analysis
- Code review
- Threat models and code reviews
- Securing build environments
- Newsflash
- Maintaining legacy code
- Securing build automation

Security Compass