**Contingent Worker Security Skill Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **CW Role** | **Domain** | **Requirements** | **Example Training Resources** |
| **SW Developer**  **-and-**  **SW Validation Engineer** | **SDL** | Understanding of the SDL (Secure Development Lifecycle) process | <https://safecode.org/courses/security-development-lifecycle-101/> |
| **Secure Design Principles** | Familiarity with secure design principles  Examples:  Don't trust user input, minimize attack surface, Defense in depth, Run with the least privilege, Check for failure and fail secure | <https://safecode.org/wp-content/uploads/2018/03/SAFECode_Fundamental_Practices_for_Secure_Software_Development_March_2018.pdf>  <http://safecode.org/wp-content/uploads/2018/01/SAFECode_Dev_Practices0211.pdf> |
| **Secure Coding** | Technical understanding of common security vulnerabilities & risks of the required programming language, and countermeasures & compensating controls  Examples: input sanitation, buffer overflow prevention, handling integer correctness, canonicalization, SQL/script injection, TOCTOU issues, secrets handling, memory sanitation | <https://safecode.org/lessons/secure-memory-handling-in-c-101/>  <https://safecode.org/courses/cross-site-scripting-xss-101/>  <https://safecode.org/courses/injections-101-sql-and-beyond/>  <https://safecode.org/lessons/csrf-101-cross-site-request-forgery-for-everyone/> |
| **Security assurance methods and tools** | Understanding of usage of compiler defenses, static analysis tools usage, secure coding guidelines, fuzzing  Bonus: Experience with using fuzzing tools like Peach, AFL, libFuzzer  Bonus: Experience with binary analysis tools for security evaluation  Bonus: Familiarity with CVSS severity risk assessment method | <https://owasp.org/www-pdf-archive/OWASP_SCP_Quick_Reference_Guide_v2.pdf>  <https://safecode.org/lessons/secure-java-programming-101/>  <https://www.oracle.com/java/technologies/javase/seccodeguide.html>  <https://safecode.org/lessons/basic-practices-for-secure-development-of-cloud-applications-part-1/>  <https://safecode.org/courses/basic-practices-for-secure-development-of-cloud-applications-part-2/>  Can be learned on the job, if needed  <https://www.first.org/cvss/> |
| **Threat Modeling** | Bonus: Understand trust boundaries and enhance security focus on the trust boundaries | <https://safecode.org/courses/threat-modeling-101/>  <https://safecode.org/wp-content/uploads/2017/05/SAFECode_TM_Whitepaper.pdf> |
| **Cryptography** | Bonus: Applied crypto knowledge  Correct usage of crypto algorithms | <https://safecode.org/courses/introduction-to-cryptography/> |
| **SW QA Tester** | **SDL** | Understanding of the SDL (Secure Development Lifecycle) process | <https://safecode.org/courses/security-development-lifecycle-101/> |
| **Security Validation** | Importance of negative and out of spec testing.  Bonus: Familiarity with fuzzing approach | <https://www.youtube.com/watch?v=jm4WcE1QPJc> |