Certified Secure Software Lifecycle Professional

Study Guide & Practice Questions

Table of Contents

Certified Secure Software Lifecycle Professional Certification & Exam

CSSLP Certification Requirements

The candidate must have a minimum of four (4) years cumulative paid full-time SDLC experience in one (1) or more of the eight (8) domains of the CSSLP. Alternatively, if the candidate has a four (4) year college degree, $(ISC)^2$ will waive one (1) year of the required experience (only one (1) year experience exemption is allowed for education).

If the candidate does not meet the minimum experience requirements but passes the certification exam they will be eligible to become an ***Associate*** of $(ISC)^2$. Once they have passed the exam, they will have a period of five (5) years in which they can earn the four (4) years required experience. Once they have earned the necessary experience they will become a fully certified CSSLP

Register for the CSSLP Exam

You need to perform several steps to book your CSSLP exam at ***Pearson VUE website***. The Pearson VUE conducts innovative computer-based testing solutions through a secure and electronic test delivery.

- Review exam availability by credential
- Visiting the Pearson VUE website
- Create a Pearson VUE account and then review the Pearson VUE NDA
- Select an appropriate testing center
- Select a convenient time
- Pay for the exam
- Check the confirmation through E-mail that the Pearson VUE will send to you. This E-mail includes appointment details, testing location, and all other relevant instructions.

CSSLP Exam Cost

The CSSLP exam price varies from country to country, or continents. Please refer to the following table for price details of different countries.

CCSP Country	Price for the exam
The United States	$599
The Asia Pacific	$599
Europe	EUR 555
The United Kingdom	GBP 479
The Middle East	$599
Africa	$599

CSSLP Exam Format

Number of Questions	175
Length of the examination	4 hours
Type of Questions	Multiple Choice
Passing grade	700 points out of 1000 points
Language	English
Testing Center	Pearson VUE

The CSSLP CBK and Exam Weights

The CSSLP Common Body of Knowledge (CBK) includes the wide spectrum of topics/material. The topics and their weights are given in table below.

Domain Name	Percentage of the Exam
1—Secure Software Concepts	13%
2—Secure Software Requirements	14%
3—Secure Software Design	16%
4—Secure Software Implementation/Programming	16%
5—Secure Software Testing	14%
6—Secure Lifecycle Management	10%
7—Software Deployment, Operations, and Maintenance	9%
8—Supply Chain and Software Acquisition	8%
Total	100%

Core Concepts of the CSSLP Exam

The core concepts of CSSLP exam include:

- secure software concepts
    - Including confidentiality, integrity, availability, authentication, authorization, and non-repudiation
- software requirements
- design
- implementation/programming
- testing
- lifecycle management
- deployment
- operations
- maintenance

Secure Software Concepts

Secure software concepts further include core concepts and security design principles.

Core Concepts—

- Confidentiality prevents unauthorized disclosure of information and data to the attackers. It can be achieved through covert, overt, and encryption.
- Integrity is the assurance that the data is not changed, altered, or manipulated. Software Integrity can be achieved through authentication, reliability, alteration, code signing, digital signatures, and hashing.
- Availability ensures that the software is available to the right person at the right time. Availability can be achieved by means of resiliency, scalability, replication, clustering, and failover techniques.
- Authentication is an access mechanism that is used to ensure that only authorized persons have access to the software applications. Authentication can be applied through multifactor authentication, single sign-on, and identify and access management.

Security design principles— includes the core principles of least privilege (e.g., run-time privileges, need-to-know, and access control), separation of duties (e.g., secret sharing and splitting, and multi-party control), defense in depth (e.g., security zones, input validation, and layered controls), fail safe (e.g., deny by default , exception handling, non-verbose errors, and deny by default), economy of

mechanism, complete mediation (e.g., session management, caching of credentials, and cookie management), open design, least common mechanism, psychological acceptability (e.g., screen layouts , complexity, and password), leveraging existing components (e.g., libraries, controls, common), and eliminating the single point of failure.

- The principle of least privilege is the act of providing only the minimum level of access that is necessary to perform a particular task regarding the software design.
- Separation of duties is the process of assigning a task to a group of people so that the chances of potential theft can be thwarted.
- Defense in depth requires the implementation of multi-layer security to prevent threats to the software application. Moreover,
- Fail safe principle ensures that the software reliably functions if the attack occurs. It also makes sure that the software functions are rapidly recoverable into its normal and original form.

Secure Software Requirements

It is imperative to understand that explicitly articulating the software requirements is essential before developing the source code of a program. Without software requirements, your application may fail to perform or even causes serious implications for your organization. Other issues include poor product quality, end-user dissatisfaction, unexpected errors, and increased cost to fix the errors, scope creep, and extensive timelines.

Core Security Requirements—

The fundamental issues that must be addressed include

- confidentiality
- integrity
- availability
- policy decomposition (e.g., external and internal requirements)
- legal
- regulatory
- industry requirements

In addition to these requirements, authentication requirement is also essential. Authentication validates the legitimate access to the company's resources and services. Authentication has further

types that include Anonymous Authentication, Basic Authentication, Digest Authentication, Integrated Authentication, Client Certificate-Based Authentication, Forms Authentication, Token-based Authentication, Smart Cards-based Authentication, and Biometric Authentication

Data Classification Requirements—Data is a valuable digital asset in any organization, therefore, its protection from being attacked, changed, altered, or manipulated is essential. Data can be comprised of two types that include structured and unstructured data. The NIST published special document 800-18 provides a framework for classifying data assets based on impact to the three core security objects (i.e., confidentiality, integrity, and availability).

Other essential elements of data classification include Labeling (e.g., impact, sensitivity), Data Ownership and Roles (data custodian, data owner), Data Lifecycle (e.g., disposal, retention, generation).

Identify Privacy Requirements—it includes three core concepts that include Data anonymization, User consent, and Disposition.

Some other fundamental concepts of Secure Software Requirements include:

- Develop Misuse and Abuse Cases
- Develop Security Requirements Traceability Matrix
- Include Security in Software Requirement Specification

Secure Software Design

The design phase is the most important in the Software Development Lifecycle (SDLC). In this phase, the software specifications interpreted into architectural blueprints that, afterwards, can be coded during the implementation phase. The important concepts of software design are described below.

Perform Threat Modeling—it involves understanding common threats that include common malware, insider threat, APT, and third party/supplier. The understanding of the "Attack surface evaluation" is also essential.

Define the Security Architecture— security architecture includes "control identification and prioritization." Various other components of security architecture include Distributed computing (e.g., message queuing, peer-to-peer, and client server), Service-oriented architecture (e.g., web services, service bus, and enterprise), Rich internet applications (e.g., constant connectivity, remote code

execution, client side exploits), Pervasive/ubiquitous computing (e.g., sensor networks, near field communication, RFID, location-based, wireless, and IoT), Embedded (e.g., firmware, control systems), Cloud architectures (e.g., infrastructure as a service, platform as a service, software as a service), Hardware platform concerns, Mobile applications.

Performing Secure Interface Design—the candidate must know the security management interfaces, log interfaces, and out-of-band management. He/she should also know the downstream/upstream dependencies (e.g., data and key sharing between the apps), protocol design choices (APIs, model, state, weakness).

Select and Evaluate Reusable Secure Design—the candidates must understand the selection and evaluation of the reusable secure design. This includes credential management (e.g., SSO, X.509), flow control (e.g., queuing, protocols, firewalls, and proxies), data loss prevention, virtualization (e.g., hypervisor, and software defined network), trusted computing (e.g., TCB, TPM), database security (privilege management, triggers, view, and encryption), programming language environment (JVM, CLR), and operating system controls and services.

Design Secure Assembly Architecture for Component-based System—this section discusses the two important concepts that include "network attached storage" and "client-side data storage."

Other important sections under Secure Software Design contain:

- Use Secure Design Principles and Patterns
- Use Security Enhancing Architecture and Design Tools
- Perform Design Security Review
- Model and Classify Data
- Modeling (Non-Functional) Security Properties and Constraints
- Performing Architectural Risk Assessment

Secure Software Implementation/Programming

Writing a secure code is a critical and important component to ensure the resiliency of software security controls. Since most of the attackers possess in-depth knowledge of programming, software developers must take every vulnerability into consideration when writing source code of a program.

Follow Secure Coding Practices—secure code involves some best practices that include:

- declarative versus imperative security
- concurrency
- output sanitization (such as encoding)
- error and exception handling
- input validation
- logging & auditing
- session management
- safe APIs
- type safety
- memory management (e.g., garbage collection, locality)
- tokenizing
- sandboxing
- cryptography (e.g., algorithm selection, encryption, agility, storage)

Analyze Code for Security Vulnerabilities—the candidate must know how to analyze the code for security vulnerabilities. Doing so requires the complete understanding of code reuse, vulnerability lists/databases (CWE, OWASP Top 10), static analysis, dynamic analysis, manual code review, and peer review.

Securely Integrate Components—the CSSLP professionals must have a great understanding of securely integrate components that security testing and analysis.

Apply Security during the Build Process—the candidates should have the knowledge of how to apply security during the building process. This involves code signing, obfuscation, and compiler switches.

Other important concepts are listed below:

- Implement Security Control
- Securely Reuse Third-Party Code or Libraries
- Look for Malicious Code/malware
- Fix Security Vulnerabilities
- Debug Security Error

Secure Software Testing

Although designing and implementing phases are important, the complete security is not achieved with their completion. Instead, the security and functionality of the software application must be verified and validated. This can be accomplished by quality assurance testing which needs to include security testing and security functionality. The CSSLP must understand what to test, who is to perform the test, and how to test for software security issues. Doing so requires the great understanding of the following core concepts.

Develop Security Test Cases—candidates can develop security test cases by understanding some important techniques that include:

- attack surface validation
- penetration
- fuzzing (e.g., mutated, generated)
- scanning (e.g., privacy, content, vulnerability)
- simulation (e.g., data and environment)
- failure (e.g., break testing, stress testing, fault injection)
- cryptographic validation (e.g., PRNG)
- regression
- continuous (e.g., synthetic transaction)
- unit testing

Develop Security Testing Strategy and Plan—the security testing strategy and plan can include:

- functional security testing (e.g., logic)
- nonfunctional security testing (e.g., scalability, performance, reliability)
- testing technique (e.g., black box and white box)
- environment (e.g., testing harness, interoperability)
- standards (e.g., SEI, OSSTMM, ISO)

Classify and Track Security Errors—this includes bug tracking (e.g., errors, defects, and vulnerabilities), and risk scoring (e.g., CVSS).

Secure Test Data—the candidates must understand the privacy and referential integrity of secure test data technique.

Other important concepts under Secure Software Testing include:

- Perform Verification and Validation Testing

- Develop Security Test Data
- Interpret Security Implications of Test Results
- Identify Undocumented Functionality

Secure Lifecycle Management

Core concepts:

- Report Security Status
- Develop Security Metrics
- Create Security Documentation
- Identify Security Standards and Frameworks
- Choose a Secure Software Methodology
- Establish Security Milestones
- Secure Configuration
- Version Control

Decommission Software—for decommission software, the CSSLP professionals must understand some key concepts that include the end of life policies, license cancellation, removal, configuration, credential removal, and data destruction.

Support Governance, Risk, and Compliance (GRC)— GRC includes regulations and compliance, legal (e.g., breach notification, intellectual property), standard and guidelines (e.g., BSIMM, Open SAMM, SAFECODE, OWASP, NIST, PCI, ISO), risk management, terminology (e.g., impact, probability, controls, residual risk, threats vulnerabilities), technical risk vs. business risk, strategies (e.g., avoid, transfer, accept, mitigate).

Software Deployment, Operations, and Maintenance

If the software is acceptable to a client or customer, then it must be installed and deployed before the final submission. After deployment, the software architects ensure that it is working in a resilient, recoverable, and reliable manner. The important security topics for software deployment, operations, and maintenance are described below.

Securely Store and Manage Security Data—the data must be stored and managed in a secure manner. Doing so requires the understanding of credentials, secrets, keys/certificates, configurations.

Ensure Secure Installation—the secure installation can be ensured through bootstrapping (management, access, and key generation), least privilege, environment, secure activation (e.g., licensing, network configuration, device configuration, whitelisting, credential, etc).

Obtain Security Approval to Operate—this includes the understanding of risk acceptance (e.g., sign-off, exception).

Support Incident Response—the CSSLP professionals must understand "Root cause analysis" technique to support incident Response.

Support Continuity of Operation—the candidates must understand the concepts that support continuity of operations. These include backup, retention, archiving, and disaster recovery.

Supply Chain and Software Acquisition

The candidates attain the knowledge some important concepts to understand supply chain and software acquisition. They will be discussed in the subsequent sections.

Verify Pedigree and Provenance—to verify pedigree and provenance, the CSSLP professional must understand secure transfer, system sharing/interconnection, code repository, build environments security, cryptographically-hashed, and digitally signed components.

Provide Security Support to the Acquisition Process—the students must understand how to provide security to the acquisition process. Doing so requires the in-depth understanding of some key concepts that include:

- audit of security policy compliance
- incident response, and reporting
- Service Level Agreements (SLAs)
- Support and maintenance structure (e.g., commercial versus community)
- Assessment of software engineering approaches
- information system security policy compliance
- security track record
- product sustainment and deployment controls (e.g., GPL requirements, operational readiness, code extension, secure configuration, upgrades, and customs)

Exam Practice Questions

1. Being software developer in the company, you are asked to monitor the functionality of the working software when it was down and unable to provide the expected results to the business. In accordance with this statement which functions are you achieving?
    a. Authentication
    b. Integrity
    c. Confidentiality
    d. Availability

Correct Answer is D – Availability guarantees the protection against destruction of information and denial of services. Remaining options are incorrect. For example, Authentication is a technique used to validate credentials of an object. Integrity deals with assuring protection against unauthorized alterations and Confidentiality provides protection against unauthorized disclosure of information while it's in transit.

2. Ten most critical web application security risks (Top 10) are published by which organization?
    a. Forums for Incident Response and Security Teams (FIRST)
    b. Computer Emergency Response Team (CERT)
    c. Open Web Application Security Project (OWASP)
    d. Web Application Security Consortium (WASC)

Correct Answer is C – OWASP, or Open Web Application Security Project provides a powerful awareness document for the security of web applications. It addresses most critical security flaws related to web applications.

3. As a CSSLP-certified in the company, your task is to implement internet protocol security (IPsec) to ensure the confidentiality of the data while it is being transmitted. This is an example of which services?
    a. Acceptance
    b. Mitigation
    c. Avoidance
    d. Transference

Correct Answer is B – Implementing IPsec at network layer helps mitigate threats to ensure the confidentiality of data being transmitted. We can also say that mitigation decreases the severity of any action.

4. Identify the correct statement that must be addressed by software security requirements.
    a. External auditor requirements
    b. Technology used in building application
    c. Software quality requirements
    d. Goals and objectives of the organization

Correct Answer is D – While defining software quality requirements, it is compulsory to address the goals and objectives of the company. They must be incorporated in the security policy of the company. Whereas, software quality requirement, external auditor requirements, and technology used in building applications are the factors that need compliance with company's policy.

5. Which of the followings information is not included in confidentiality requirements?
    a. User's cardholder data
    b. Software architecture and network diagram
    c. Directory information
    d. Personally identifiable information (PII)

Correct Answer is C – Directory information is not included in confidentiality requirements because this information can be seen publically. It also provides free access to view all sorts of public data. It can be found in a public directory like tax directory or phone book.

6. _____is the major reason due to which an application can be susceptible to a Man-in-the-Middle Attack.
    a. Lack of encryption
    b. Improper archiving
    c. Lack of auditing

d. Improper session management

Correct Answer is D – Man-in-the-Middle Attack also known as Janus attack is a situation in which the hacker secretly changes and relays the communication channel between two parties who are unaware of being attacked. Anyone can compromise the system if sessions are not managed properly. Session identifiers should not be easily guessable.

7. Threat modeling is started at which stage of SDLC (software development life cycle)?
    a. Deployment
    b. Implementation
    c. Requirements analysis
    d. Design

Correct Answer is D – Threat modeling is initiated at the design phase of software development lifecycle. Basically, it is a technique for optimizing application security by recognizing vulnerabilities and objectives. It also defines procedures to prevent the potential threats that can badly compromise the software application.

8. Internal structure and working of a database application can be protected from disclosure by using which method?
    a. Encryption
    b. Views
    c. Normalization
    d. Triggers

Correct Answer is B – By defining multiple views, the database can be protected from disclosure because they provide a variety of benefits for the protection of database. Other options are incorrect in this scenario. For example, normalization is a process of making databases more reliable and simple.

9. Which layer of the open systems interconnection model deals with security controls to mitigate side channel attacks effectively?
    a. Physical layer
    b. Data link layer
    c. Network layer
    d. Transport layer

Correct Answer is A – Side channel attacks mostly need physical access to the system/device, that's why you need a physical layer to mitigate side-channel attacks in a proper manner. On the other hand, transport, network, and data link layers provide other services related to the network.

10. Why a software developer develops program/software? Identify the major purpose from the followings.
    a. To mitigate hacker threats
    b. To solve business problems
    c. To capture market share
    d. To create new products

Correct Answer is B – The major objective of developing programs/software is to solve different kinds of problems faced by businesses. It is also used to automate an existing system like departmental stores. The other options are incorrect in this scenario.

11. Which of the following services are not provided by code signing? Choose the nearest option.
    a. Authentication of users
    b. Authenticity of code origin
    c. Anti-tampering protection
    d. Runtime permissions for code

Correct Answer is A – All of the following services are provided by code signing like authenticity of code origin, anti-tampering protection and runtime permissions for the code. It does not support the services for user's authentication.

12. Name the process that is used to combine libraries, variables, dependency files, and functions necessary for the machine to run a program.
    a. Instantiation
    b. Interpretation
    c. Compilation
    d. Linking

Correct Answer is D – Linking is the process that combines all the libraries, variables, dependency files, and functions. All these files are vital for the program to run successfully. Remaining options are incorrect in this regard.

13. IF-THEN rule is a feature of which type of software testing?
    a. Unit testing
    b. Integration
    c. Logic
    d. Scalability

Correct Answer is C – IF-THEN rule is used to construct a logic that is required for software testing. This method is known as logic testing. Remaining choices are incorrect in this particular scenario.

14. What type of method is used to support software's white box testing against insider threats? Choose the nearest option.
    a. Scanners
    b. Source code analyzers
    c. Banner grabbing programs
    d. Fuzzers

Correct Answer is B – Source code analyzer is a type of structural analysis or white box testing against insider threats. Prior knowledge of the code and configuration must be known in this type of testing. Different attacks caused by viruses like Trojan horse and malware can be detected by using this facility. Other options are not true in this case. For example, banner

grabbing is a method used to collect details about any remote computer on a network and the services running on its open ports.

15. Vulnerability scans are used to_____.
    a. Measure the skills and technical know-how of security tester
    b. Detect weaknesses and loopholes in the software/program
    c. Measure the resiliency of software by attempting to exploit weaknesses
    d. Detect the effectiveness of security controls implemented in the software

Correct Answer is B – Any vulnerability is a flaw in the software and they are detected by using vulnerability scans. It scans the whole software to detect loopholes in the software.

16. During the accreditation process, residual risk of software estimated for deployment should be accepted formally by which of the followings?
    a. Security organization
    b. Information technology management
    c. Business owner
    d. Executive management and board members

Correct Answer is C – Risk is a factor which is directly associated with the business owner. He is the only person who must accept the risk of software deployment.

17. The concluding activity in software acceptance method can be determined by using _____ testing.
    a. User acceptance testing
    b. Unit testing
    c. Regression control
    d. Integration testing

Correct Answer is A – End users of any business have the final verdict on the software deployment whether to go/no-go decision. Users acceptance testing determines the readiness of any software to be deployed in an environment. The remaining options are incorrect. For example, during unit testing, software is tested unit wise separately.

18. Which of the following principles provides an insight into the verification of activities used to determine whether the deployed software is working properly or not?
    a. Recoverability
    b. Resiliency
    c. Redundancy
    d. Reliability

Correct Answer is D – Reliability makes sure whether the software is properly working or not. Remaining options are not true in this case. For example, redundancy is an attribute associated mostly with databases.

19. Check-ins and check-outs, backups and versioning are all main components of _____.
    a. Incident management
    b. Problem management
    c. Patch management
    d. Release management

Correct Answer is D – All the activities mentioned above are mainly associated with release management. Remaining choices are not true in this regard. For example, problem management is a method to prevent complications and resulting events from happening. It is also used to reduce the effect of events that cannot be prevented.

20. From the given choices, which of the following is NOT a feature of good security measure? Select the most suitable option.
    a. Collected manually
    b. Contextually relevant
    c. Quantitatively expressed
    d. Objectively expressed

Correct Answer is A – An up to the mark security measure should be relevant to its context and can be expressed quantitatively regardless of how many times it is collected. The outcomes do not have variations most of the times.

21. The process of removing unnecessary documentation, maintenance hooks, flags, and debugging code prior to deployment are the examples of software _____.
    a. Obfuscation
    b. Reversing
    c. Hardening
    d. Patching

Correct Answer is C – Providing various methods of protection is typically known as software hardening that includes changing passwords, minimizing accessible means of attack, and disabling unnecessary services. Other options are not true in this regard.

22. An increased requirement for security in the software supply chain is mainly accredited to _____.
    a. Decreasing the trust of consumers on software
    b. Occurrences of malicious code and logic found in acquired software
    c. Increased foreign trade agreements
    d. Cessation of development activities in the organization

Correct Answer is B – Malicious codes and other destructive logics impact the software development domain a lot. That's the reason the need for security in the software supply chain is imperative. On the contrary, outsourcing is an act used by enterprises to shift tasks, job, or operations to an external third party in order to save time and budget. Foreign trade agreements are also increasing but they are not the main focus in security for software supply chain.

23. Identify the phase of acquisition life cycle that issues advertisements to source and evaluate suppliers.
    a. Delivery
    b. Contracting
    c. Development
    d. Planning

Correct Answer is B – Contracting is the phase of acquisition life cycle which deals with the sourcing of suppliers, evaluating their responses, and issuing contract award to winning supplier. The other options are not related to this statement as planning is the initial phase in which we plan to make a software. Development phase includes the developmental activities related to the software.

24. Being a website manager in the organization, the top management asks you to protect the data residing on the website, such as images and database models, from being copied or duplicated. In this case, what type of legal instrument should you use?
    a. Trade secret
    b. Patents
    c. Copyright
    d. Trademark

Correct Answer is C – Copyright is a legal instrument used to grant the developer of work exclusive rights for his/her distribution and use. © Symbol can be used in copyrighted information. Trademarks are recognizable symbols/signs which identifies a product or a service from a particular source. Likewise, patents are used for inventions. Using patents, no one can claim your invention.

25. Which of the followings cannot be detected using the code review process?
    a. Trojan horse
    b. Backdoors
    c. Logic bombs
    d. Logic Flaws

Correct Answer is D – Logic flaws are not related to syntax but are directly related to semantic issues. They are pertaining to design and can be detected by threat modeling method, not by code review process. Contrarily, Trojan horse, backdoors, and logic bombs can be found by using code review process because these issues are implanted in the code.

*****