CSSLP Secure Software Prep

Free Practice Test — 30 Real Exam-Style Questions

with full answer key & explanations

Unlock the full bank of 1010 questions
+ unlimited timed mock exams + mistake book

Practice on the web: https://certs.theorypractice.app/csslp
$2.99 / week · $6.99 / month · cancel anytime

What you unlock: all 1010 questions • unlimited timed mock exams • mistake book • instant explanations

Also on iOS & Android — and watch the full Q&A walkthrough on YouTube @CertsQuizPrep

Practice Questions

Try all 30 first, then check the answer key at the back.

Want the other 980+ questions & full timed mock exams? Unlock at https://certs.theorypractice.app/csslp

1. During software deployment, which category includes sensitive information like database connection strings and API tokens that need to be secured through encryption and strict access controls?

A. Credentials
B. Keys/Certificates
C. Configurations
D. Secrets

2. The terms Transactions, Constrained Transactions, and Well-formed Transactions relate to which of the following security models?

A. Biba
B. Brewer-Nash
C. Clark-Wilson
D. Bell-LaPadula

3. Which of the following security principles is LEAST related to the concept of confidentiality?

A. Data Hiding
B. Availability
C. Encryption
D. Access Controls

Also on iOS & Android — and watch the full Q&A walkthrough on YouTube @CertsQuizPrep

4. Which of the following is MOST critical when ensuring the proper actions are taken after discovering a software vulnerability?

A. Authentication
B. Authorization
C. Accessibility
D. Responsibility

Unlock all 1010 questions + timed mock exams
→ **https://certs.theorypractice.app/csslp**
$2.99/week or $6.99/month · cancel anytime · scan to start

Unofficial study material · not affiliated with any certifying body

5. In a hardware security module (HSM), a random number generator and key generator are part of which of the following?

A. Cryptographic Processor
B. Secure Memory
C. Root of Trust
D. Network Interface

6. Which of the following types of certificates would be most useful in ensuring the integrity and authenticity of a distributed mobile application?

A. Server Certificate
B. Extended Validation Certificate
C. Software Publisher Certificate
D. Personal Certificate

Want the other 980+ questions & full timed mock exams? Unlock at https://certs.theorypractice.app/csslp

7. Which of the following is used to manage the dependencies of a software project?

A. Compiler
B. Static linker
C. Dynamic linker
D. Package manager

8. Which of the following database architecture models provides the HIGHEST level of data centralization?

A. Federated Database Architecture
B. Client-Server Database Architecture
C. Centralized Database Architecture
D. Distributed Database Architecture

9. Which of the following is NOT one of the software security requirements recommended in OWASP's Secure Coding Practices?

A. Input validation
B. Output encoding
C. Session management
D. Monitor network traffic

Unlock all 1010 questions + timed mock exams
→ **https://certs.theorypractice.app/csslp**
$2.99/week or $6.99/month · cancel anytime · scan to start

Unofficial study material · not affiliated with any certifying body

Also on iOS & Android — and watch the full Q&A walkthrough on YouTube @CertsQuizPrep

10. Which of the following roles is responsible for ensuring the secure configuration and regular patching of servers and network devices?

A. Security manager
B. System administrator
C. Network engineer
D. Security auditor

11. Which of the following is the HIGHEST level of security assurance that can be obtained from following the NIST Cybersecurity Framework?

A. Adaptive
B. Intermediate
C. Basic
D. Foundational

12. Which of the following methods for ensuring the authenticity and integrity of a message is the LEAST effective?

A. Checksum
B. Digital Signature
C. Hashing
D. Message Authentication Code (MAC)

Want the other 980+ questions & full timed mock exams? Unlock at https://certs.theorypractice.app/csslp

13. Which type of protection is used to safeguard the secretive formulation of a popular computer software algorithm that gives a competitive advantage?

A. Patent
B. Copyright
C. Trademark
D. Trade secret

Unlock all 1010 questions + timed mock exams
→ **https://certs.theorypractice.app/csslp**
$2.99/week or $6.99/month · cancel anytime · scan to start
Unofficial study material · not affiliated with any certifying body

14. Which of the following software testing approaches involves a public challenge to find security flaws by awarding prizes or incentives?

A. Vulnerability assessments
B. Code reviews
C. Bug bounties
D. Penetration testing

15. Which of the following is a classification of secure software testing activities?

A. Logical
B. Administrative
C. Procedural
D. Functional

Also on iOS & Android — and watch the full Q&A walkthrough on YouTube @CertsQuizPrep

16. Which of the following vulnerabilities exploits the way JSON is processed in web applications?

A. Cross-site request forgery (CSRF)
B. Command injection
C. SQL injection
D. JSON Injection

17. Which of the following secure coding practices should be used as a last resort?

A. Use of blacklisting
B. Input validation
C. Use of whitelisting
D. Error handling

18. During a secure code review process, what is the term used for code practices that degrade performance and clarity?

A. Privilege Levels
B. Inefficient Code
C. Known Vulnerabilities
D. Logging

Want the other 980+ questions & full timed mock exams? Unlock at https://certs.theorypractice.app/csslp

19. During which phase of the secure software development lifecycle should a comprehensive threat model be established?

A. Implementation
B. Testing
C. Design
D. Planning

20. Which type of software license allows modification and distribution of the code but ensures that any modified versions are also freely available under the same license?

A. Permissive
B. Open
C. Copyleft
D. Copyright

21. At which stage of the software development lifecycle should security requirements be defined and formalized to mitigate supply chain risks?

A. Testing
B. Requirements Gathering
C. Design
D. Implementation

Also on iOS & Android — and watch the full Q&A walkthrough on YouTube @CertsQuizPrep

22. Which software maintenance model relies on publicly available resources and forums for support in the management of vulnerabilities and patches?

A. Open-source
B. Proprietary
C. Collaborative
D. Unmanaged

23. Which of the following practices is most effective in ensuring the integrity and authenticity of software updates from third-party vendors?

A. Secure Transfer
B. System Sharing/Interconnections
C. Code Repository Security
D. Cryptographically Hashed, Digitally Signed Components

Unlock all 1010 questions + timed mock exams
→ **https://certs.theorypractice.app/csslp**
$2.99/week or $6.99/month · cancel anytime · scan to start

Unofficial study material · not affiliated with any certifying body

24. Which of the following controls are used to maintain integrity and traceability of code and are established through code repositories in a secure software supply chain?

A. Vendor traceability controls
B. Vendor authentication controls
C. Code integrity controls
D. Vendor management controls

Want the other 980+ questions & full timed mock exams? Unlock at https://certs.theorypractice.app/csslp

25. During the Secure Software Development Lifecycle (SDLC), what is the primary purpose of conducting a security audit on third-party software components?

A. Assess
B. Identify
C. Respond
D. Monitor

26. Which of the following licensing models allows for modifications of a purchased software's source code by the end user?

A. Freeware
B. Open Source
C. Proprietary Software
D. COTS

27. Which of the following approaches reduces the likelihood of a security breach in the software supply chain?

A. Implement comprehensive code review practices
B. Utilize open-source libraries without verification
C. Ignore vendor security audits
D. Outsource critical components without risk assessment

Also on iOS & Android — and watch the full Q&A walkthrough on YouTube @CertsQuizPrep

28. Which of the following terms is commonly associated with ensuring that software development processes align with organizational standards and policies?

A. Compromise
B. Conformance
C. Compliance
D. Collaboration

29. Who is responsible for managing changes in software requirements during the development lifecycle?

A. Project managers
B. Requirements management team
C. Configuration managers
D. Quality assurance team

30. In software development, which of the following is NOT a common phase in the Secure Software Development Lifecycle (SDLC)?

A. Design
B. Implementation
C. Marketing Strategies
D. Requirement Analysis

Unlock all 1010 questions + timed mock exams
→ **https://certs.theorypractice.app/csslp**
$2.99/week or $6.99/month · cancel anytime · scan to start

Unofficial study material · not affiliated with any certifying body

Answer Key & Explanations

You just practised 30 of 1010. Unlock every question + timed mocks at https://certs.theorypractice.app/csslp

1. D — Secrets
Answer: Secrets During the deployment and maintenance phases of software development, it is crucial to securely store and manage various types of sensitive information. Examples include: Credentials: These control access to different environments and tools. Each environment and account should have dedicated credentials following the principle of least privilege. Secrets: This category includes data such as database connection strings, API tokens, and other sensitive information which should be protected with encryption and access controls. Keys/Certificates: Proper management of encryption keys and digital certificates is necessary. They should not be embedded directly into the code and should be validated before usage. Configurations: The security configuration of an application must be preserved through access control measures and ensuring the integrity and authenticity of the configuration data.

2. C — Clark-Wilson
Answer: Clark-Wilson The Clark-Wilson security model emphasizes the concept of data integrity by defining Constrained Data Items (CDIs), and Unconstrained Data Items (UDIs). Transactions refer to processes that change the state of CDIs while ensuring compliance with integrity policies. Well-formed Transactions are sequences of operations that transition CDIs from one consistent state to another. Bell-LaPadula is a confidentiality protection model that combines attributes of Mandatory Access Control (MAC) and Discretionary Access Control (DAC). Its Simple Security Rule prevents reading data at a higher level of classification, while its * property prevents writing data to a system with a lower classification level. Biba is an integrity model designed to protect higher-level, more trustworthy data from being corrupted by lower-level data. Its no-write-up rule blocks systems from writing data to a system with a higher classification level. Its second rule states that a system reading/processing data from a lower-level system will have its integrity level lowered as a result. Brewer-Nash or the Chinese Wall is a confidentiality model for enterprises. It addresses the case where one group within an organization may have information that cannot be shared with another.

3. B — Availability
Answer: Availability Availability is a principle that ensures systems and data are accessible when needed, but it is not directly related to confidentiality. Confidentiality is more concerned with preventing unauthorized access to information. Here are the definitions of the other principles: Encryption: A technique to secure information by transforming it into a coded format that is unreadable without a decryption key. Access Controls: Mechanisms that restrict access to information and resources to only authorized users. Data Hiding: Techniques used to conceal information within other data to prevent unauthorized access or detection.

4. D — Responsibility
Answer: Responsibility Responsibility: Assigning ownership for ensuring the software vulnerability is assessed, documented, and resolved. This includes determining who is accountable for each action step in the remediation process. Authentication: Verifying the identity of users and administrators who access the systems. Common factors include passwords, tokens, and biometrics. Authorization: Confirming that authenticated users have appropriate permissions to perform certain actions. Typically managed through access controls. Accessibility: Ensuring that the software and data are available to users when needed. This

Unlock all 1010 questions + timed mock exams
→ **https://certs.theorypractice.app/csslp**
$2.99/week or $6.99/month · cancel anytime · scan to start

Unofficial study material · not affiliated with any certifying body

might involve implementing redundant systems, backups, and ensuring minimal downtime.

5. A — Cryptographic Processor
Answer: Cryptographic Processor A hardware security module (HSM) provides hardware protection for cryptographic keys on a system. Key elements include: Cryptographic Processor: The cryptographic processor includes a random number generator, key generators, encryption algorithms, and digital signature algorithms. Secure Memory: Secure memory includes storage for sensitive keys and certificates. Root of Trust: The root of trust ensures that the identity and integrity of the module can be verified. Network Interface: The network interface handles communication between the HSM and other systems but is not involved in cryptographic processing.

6. C — Software Publisher Certificate
Answer: Software Publisher Certificate Digital certificates have several uses, including: Type of Certificate Purpose Usage Scenario Personal Certificate Uniquely identify a person Secure email or similar functions Server Certificate Prove the identity of a server Enable encryption for SSL/TLS traffic (such as HTTPS) Extended Validation Certificate Provide extra validation of domain ownership by a company Largely deprecated due to vulnerability to attackers registering companies with the same name in different jurisdictions Software Publisher Certificate Digitally sign software Prove the authenticity and integrity of the code

7. D — Package manager
Answer: Package manager Package managers automate the process of installing, upgrading, configuring, and removing software dependencies. Compilers convert source code into machine code. Static linking copies required dependencies into an executable during compilation, creating a faster, easily-distributed, and bloated file. Dynamic linking stores the names and locations of dependencies to be resolved at runtime. It creates smaller files at risk of hijacked dependencies.

8. C — Centralized Database Architecture
Answer: Centralized Database Architecture Some common database architecture models include: Architecture Description Centralized Database Architecture All data is stored and managed in a single central location, providing the highest level of centralization. Distributed Database Architecture Data is distributed across multiple physical locations, enhancing availability and redundancy. Federated Database Architecture Multiple autonomous databases are interconnected, allowing access to data across different systems. Client-Server Database Architecture Clients make requests to a central server, which processes and manages data. It provides a balance between centralization and distribution.

9. D — Monitor network traffic
Answer: Monitor network traffic The OWASP Secure Coding Practices guide lists several important software security requirements, including: Requirement Input validation Output encoding Authentication and password management Session management Access control Although monitoring network traffic is important for overall security, it is not explicitly listed as a software security coding practice by OWASP.

10. B — System administrator
Answer: System administrator A system administrator is responsible for the upkeep, configuration, and reliable operation of computer systems, especially multi-user computers, such as servers. They ensure that the servers and network devices are securely configured and patched regularly to prevent security vulnerabilities. A network engineer focuses on the design and implementation of network infrastructures, while

Unlock all 1010 questions + timed mock exams
→ **https://certs.theorypractice.app/csslp**
$2.99/week or $6.99/month · cancel anytime · scan to start
Unofficial study material · not affiliated with any certifying body

a security auditor reviews and assesses the effectiveness of security policies and controls. A security manager oversees the security of an organization's information systems but is not typically responsible for the hands-on management of server configurations and patches.

11. A — Adaptive
Answer: Adaptive The NIST Cybersecurity Framework provides a policy framework for computer security guidance to private sector organizations. The framework is divided into five functions: Identify, Protect, Detect, Respond, and Recover. These functions help organizations manage cybersecurity risk by organizing information, enabling risk management decisions, addressing threats, and improving by learning from past actions. The levels of implementation tiers described in the framework are: Tier Description Tier 1: Partial Cybersecurity risk management is not formalized and may be ad hoc. Tier 2: Risk Informed Risk management practices are approved by management but lack consistency and alignment with organizational risk strategy. Tier 3: Repeatable Formal policy and processes are consistently followed and organizational-wide. Tier 4: Adaptive The organization adapts its cybersecurity practices based on lessons learned and predictive indicators derived from previous cybersecurity activities.

12. A — Checksum
Answer: Checksum. Several methods can be used to ensure the authenticity and integrity of a message, including: Digital Signature: A digital signature is a cryptographic value that is calculated from the data and a secret key known only by the sender. It is highly effective for both authenticity and integrity. Hashing: Hashing involves using a hash function to convert the data into a fixed-size string of characters. Even a small change in the input will produce a significantly different output, making it effective for ensuring integrity. Message Authentication Code (MAC): A MAC is created using a secret key and can be used to verify the integrity and authenticity of a message. It is more secure than a simple checksum. Checksum: A checksum is a simple error-detection scheme that can detect accidental errors in data. It is less effective against intentional tampering as it does not use cryptographic methods.

13. D — Trade secret
Answer: Trade secret Intellectual property (IP) can be protected in various ways, including: 1. Patent: A patent provides exclusive rights to an invention for a specified period of time. Patents can prevent others from using the invention even if they claimed to have invented it independently. 2. Copyright: Copyright protects written works and artistic expression from being used or copied without the creator's consent and proper attribution. They limit adaptations, performances, and who can profit from the work. 3. Trademark: Trademark protects brand association and can be either registered or common-law. Images and company names are commonly trademarked items. 4. Trade Secret: A trade secret is intellectual property that is protected only as long as it remains secret. The formula for a computer software algorithm that provides a competitive edge is often protected as a trade secret.

14. C — Bug bounties
Bug bounties use a public challenge system to incentivize security researchers to discover and report vulnerabilities in software applications. This approach often includes cash rewards or other incentives based on the severity of the flaws found.

15. D — Functional
Answer: Functional. Secure software testing is classified as functional or non-functional. Functional testing verifies that the software performs its intended functions, while non-functional testing ensures aspects like performance, usability, and reliability.

Unlock all 1010 questions + timed mock exams
→ **https://certs.theorypractice.app/csslp**
$2.99/week or $6.99/month · cancel anytime · scan to start
Unofficial study material · not affiliated with any certifying body

16. D — JSON Injection
Answer: JSON Injection. JSON Injection vulnerabilities occur when user input is not properly sanitized and is included within server-sent JSON data. Attackers can craft malicious inputs that alter the structure of JSON data, potentially leading to data exposure, data modification, or execution of arbitrary code. Injection vulnerabilities are a significant category of security flaws and include many types, such as: Type Description SQL Injection SQL injection attacks involve providing malicious input that is included in a database request. SQL injection can read, write, or delete data contained within a database accessible to a vulnerable application. Command Injection Command injection vulnerabilities allow an attacker to run commands in the system terminal. For example, an application may run a command in the shell using user-provided input, which may be crafted to change the intent of the command or run additional commands. Integer Overflow Integers have a fixed size in memory and are only able to store a certain range of values. If a value to be stored in a variable exceeds this range, it wraps around and is interpreted as a smaller value. Path Traversal In a filepath, ../ indicates that the system should look in the next directory up in the file system. Path traversal vulnerabilities allow an attacker who can specify the name of a file to be read/written by an application to read/write files outside of the intended directory. JSON Injection User-provided input should be properly sanitized and encoded before being included in JSON data. Exploits can occur when unsanitized input is included within server-sent JSON data. Cross-Site Request Forgery (CSRF) Cross-site request forgery (CSRF) attacks involve tricking the browser of an authenticated user into performing an HTTP request without their knowledge/consent. For example, a user logged into social media could have their password changed if a malicious webpage tricked their browser into performing a password change request and the social media site lacked CSRF protections.

17. A — Use of blacklisting
Answer: Use of blacklisting In secure software development, blacklisting should be considered a last resort. Here's why: Practice Description Priority Input validation Ensuring data is accurate and safe before use. High Use of whitelisting Only allowing known good inputs, blocking all others. High Error handling Managing errors gracefully without exposing sensitive information. High Use of blacklisting Blocking known bad inputs; however, new threats may bypass it. Low Blacklisting is considered less effective as it relies on blocking known bad inputs, which may not cover all potential threats. Whitelisting and input validation are preferred as they provide a broader security coverage.

18. B — Inefficient Code
Answer: Inefficient Code During code reviews, developers assess code for various issues including inefficiencies. Complex or obfuscated code is categorized as Inefficient Code because it hinders code readability, maintainability, and performance. Common checks in a code review include: Inefficient Code: Simplifying complex or obfuscated code to enhance analysis or execution. Known Vulnerabilities: Checking code against vulnerabilities listed in resources like OWASP Top 10. Errors and Exception Handling: Ensuring code tests for all possible error cases and handles exceptions effectively. Injection Flaws: Implementing input validation to protect against injection attacks. Cryptographic Strength: Using trusted algorithms and libraries for cryptographic operations. Unsafe and Deprecated Function Calls: Using only approved functions and removing deprecated or unnecessary ones. Privilege Levels: Following the principle of least privilege when implementing code. Logging: Properly logging errors without revealing sensitive information. Secure Key Information: Protecting cryptographic keys, passwords, and similar authentication data.

19. C — Design
Answer: Design A comprehensive threat model should be established during the design phase of the secure

Unlock all 1010 questions + timed mock exams
→ **https://certs.theorypractice.app/csslp**
$2.99/week or $6.99/month · cancel anytime · scan to start
Unofficial study material · not affiliated with any certifying body

software development lifecycle. This phase involves identifying potential threats, vulnerabilities, and potential impacts on the system. Establishing a threat model early allows the team to integrate security measures effectively from the beginning. The main phases of a secure software development lifecycle are: Planning: Define security requirements and objectives. Design: Develop a comprehensive threat model and security architecture. Implementation: Write secure code and conduct code reviews. Testing: Perform security testing, such as penetration testing. Deployment: Ensure secure deployment and configuration. Maintenance: Monitor and maintain security practices.

20. C — Copyleft
Answer: Copyleft Third-party software may operate under various licensing models, including: Copyright: A copyright protects the intellectual property of the author, restricting how it can be used Permissive: Permissive licenses (MIT, BSD, etc.) impose minimal requirements on how software can be used or redistributed Copyleft: Copyleft licenses (GPL, etc.) ensure that software using the original source code provides the same rights to the user, meaning any modified versions must also be freely available under the same license Open is not a licensing model.

21. B — Requirements Gathering
Answer: Requirements Gathering Security requirements must be defined and formalized during the Requirements Gathering stage of the software development lifecycle. Ensuring that security is built into the design from the beginning is crucial in mitigating supply chain risks. This process involves identifying potential security issues and addressing them through specific requirements that guide the entire development process. The main steps of incorporating security into the software development lifecycle are: Stage Description Requirements Gathering Define and formalize security requirements to mitigate supply chain risks; address potential security issues. Design Integrate security requirements into the software design; establish secure development practices. Implementation Follow secure coding practices; perform code reviews and security testing. Testing Conduct security testing and vulnerability assessments; ensure security requirements are met. Deployment Implement security measures in the deployment process; ensure secure configuration and environment. Maintenance Monitor and update the software to address new security threats; maintain the security posture.

22. A — Open-source
Answer: Open-source Maintenance of third-party code may be performed under one of two models: Open-source: Software under this model typically relies on community participation for maintenance, including support through forums and public websites. Proprietary: The vendor or developer of the third-party code, or an appointed organization, is responsible for addressing questions and resolving issues. Collaborative and unmanaged are not code maintenance models.

23. D — Cryptographically Hashed, Digitally Signed Components
Cryptographically Hashed, Digitally Signed Components: Ensuring that software updates are cryptographically hashed and digitally signed is crucial in verifying the authenticity and integrity of the updates received from third-party vendors. This practice helps safeguard against supply chain attacks where an attacker might insert malicious code. Secure Transfer: While transferring software via secure channels like TLS can protect the data during transfer, it does not guarantee the authenticity or integrity of the software itself. System Sharing/Interconnections: Securing interconnections is critical for protecting against attacks at network interfaces but does not inherently ensure the authenticity of software updates. Code Repository Security: Protecting code repositories against unauthorized changes is important but does not verify the

Unlock all 1010 questions + timed mock exams
→ **https://certs.theorypractice.app/csslp**
$2.99/week or $6.99/month · cancel anytime · scan to start
Unofficial study material · not affiliated with any certifying body

authenticity of third-party software updates.

24. C — Code integrity controls
Answer: Code integrity controls Code integrity controls ensure the integrity of the code by maintaining a traceable history of changes and contributions. These controls are implemented via code repositories and help verify the authenticity and security of the software supply chain. Vendor management controls are policies and processes to manage suppliers and contracts. Vendor traceability controls ensure that each component of the software can be traced back to its origin. Vendor authentication controls verify the identities of suppliers and contributors to the software.

25. A — Assess
Answer: Assess Conducting a security audit on third-party software components within the SDLC helps to assess potential security risks and vulnerabilities that these components might introduce to the overall software system. This audit is a crucial step in identifying and managing supply chain risks. Step Description Identify All third-party components that could potentially introduce risks are identified. Assess Each component is evaluated for security vulnerabilities and risks. Respond Strategies for mitigating identified risks, such as patching vulnerabilities or replacing risky components, are developed. Monitor Continuous monitoring ensures that the risks associated with third-party components are managed effectively over time.

26. B — Open Source
Answer: Open Source Proprietary Software: This software is licensed under strict terms that don't allow modification by the end user. Commercial off the Shelf (COTS): COTS software is mass-produced and available for sale to the general public but does not allow for end user modifications. Freeware: Freeware is software available at no cost but generally does not allow modification to its source code. Open Source: Open Source software is distributed with a license that allows the end user to view, alter, and distribute the source code.

27. A — Implement comprehensive code review practices
Implementing comprehensive code review practices can help identify and mitigate potential security vulnerabilities early in the development process, thereby reducing the risk of a security breach in the software supply chain. In contrast, using open-source libraries without verification, ignoring vendor security audits, or outsourcing critical components without proper risk assessment increases the likelihood of security breaches.

28. B — Conformance
Answer: Conformance Conformance refers to adherence to internal organizational standards and policies. Compliance, on the other hand, deals with adherence to external rules and laws.

29. B — Requirements management team
Answer: Requirements management team The requirements management team is responsible for overseeing and managing changes to software requirements to ensure the project stays on track and meets stakeholders' expectations. Configuration managers focus on managing the configuration of software, hardware, and documenting changes. Quality assurance teams ensure the final product meets quality standards, whereas project managers oversee the overall progress and provide leadership.

30. C — Marketing Strategies
Answer: Marketing Strategies The Secure Software Development Lifecycle (SDLC) typically includes the following phases: 1. Requirement Analysis 2. Design 3. Implementation Other common phases include testing, deployment, and maintenance, but marketing strategies are not typically part of the SDLC.

Unlock all 1010 questions + timed mock exams
→ **https://certs.theorypractice.app/csslp**
$2.99/week or $6.99/month · cancel anytime · scan to start
Unofficial study material · not affiliated with any certifying body

Ready to pass?

Unlock the full CSSLP Secure Software Prep bank, every explanation, and unlimited timed mock exams.

Scan to start practising
https://certs.theorypractice.app/csslp

Watch the full video walkthrough on YouTube @CertsQuizPrep

Unlock all 1010 questions + timed mock exams
→ **https://certs.theorypractice.app/csslp**
$2.99/week or $6.99/month · cancel anytime · scan to start

Unofficial study material · not affiliated with any certifying body