

試験Secure-Software-Design基礎訓練 & 一生懸命に Secure-Software-Design合格受験記 | 便利なSecure- Software-Design前提条件

Secure Software Design (D487) In-Depth Study Guide

Subtitle: Based on Practical Core Software Security - A Reference Framework

Format: Chapter-by-Chapter Breakdown with Exam Readiness, Terminology, Threat Models, Deliverables, and Real-World Applications

Chapter 1: Introduction

- Differences between software security and application security
- Secure code vs quality code
- CIA Triad with definitions, examples, and violations
- Threat modeling overview (STRIDE, DREAD, DFDs, trust boundaries)
- Attack surface validation: entry/exit points, expansion via feature additions

Chapter 2: The Security Development Lifecycle (SDL)

- SDL vs SDLC distinction
- SDL phases mapped to SDLC phases
- Maturity models: BSIMM vs OpenSAMM
- ISO/IEC 27034 standard
- Tools and talent (SAST, DAST, manual reviews, SDL roles)
- Principle of Least Privilege and privacy integration
- Metrics: vulnerabilities caught pre-release, time to patch, training coverage

Chapter 3: Security Assessment (SDL Phase A1)

- Discovery session outcomes and stakeholders
- PIA planning, privacy questions, initial risk classification
- SDL Project Outline, Product Risk Profile, Threat Profile
- Metrics Template deliverable

Chapter 4: Architecture (SDL Phase A2)

- Policy compliance analysis with internal/external controls
- Threat modeling artifacts (DFDs, STRIDE threat mapping, data classification)
- Architecture Threat Analysis and mitigation planning
- Open-source evaluation (licensing, vulnerability exposure)
- Privacy information gathering

無料でクラウドストレージから最新のJpshiken Secure-Software-Design PDFダンプをダウンロードする: https://drive.google.com/open?id=1_ITKVgZ-R4BT9s6tK2wfeHW-zqbgqqNT

あなたが失敗した場合、あなたのレッスンを学ぶことを忘れないでください。それでも自分でテストの準備をしていて、何度も失敗する場合は、有効なSecure-Software-Designスタディガイドを選択してください。これは、試験をクリアして認定を取得するための最良の方法です。優れたSecure-Software-Design学習ガイドは、効率的な準備と効率的な練習への近道となります。無駄な努力を避け、興味深いことをします。Jpshikenは、受験者が最初の試行で100%合格することを保証する100%合格率Secure-Software-Designスタディガイドファイルをリリースします。

WGU Secure-Software-Design 認定試験の出題範囲:

トピック	出題範囲
トピック 1	<ul style="list-style-type: none">デザインパターンの選択と実装: このセクションでは、ソフトウェア開発者およびソフトウェアアーキテクトのスキルを測定し、適切なデザインパターンの選択と実装について学びます。受講者は、一般的なデザインパターンとソフトウェア開発におけるその応用について考察します。この教材は、繰り返し発生する設計上の問題を解決し、コード構成を改善するために、特定のパターンをいつ、どのように適用するかを理解することに重点を置いています。

トピック 2	<ul style="list-style-type: none"> ソフトウェアアーキテクチャの種類: この試験セクションでは、ソフトウェアアーキテクトのスキルを評価し、大規模ソフトウェアシステムで使用される様々なアーキテクチャの種類を網羅します。受講者は、システム設計の意思決定を導く様々なアーキテクチャモデルとフレームワークを学習します。このコンテンツでは、特定のプロジェクト要件と組織のニーズに最適なアーキテクチャパターンを特定し、評価する方法を学びます。
トピック 3	<ul style="list-style-type: none"> ソフトウェアシステム管理: この試験セクションでは、ソフトウェアプロジェクトマネージャーのスキルを測定し、大規模ソフトウェアシステムの管理について学びます。受講者は、構想から導入までソフトウェアプロジェクトを監督するためのアプローチを学びます。教材は、複雑なソフトウェアソリューションの確実な提供を実現するための調整戦略と管理手法に重点を置いています。
トピック 4	<ul style="list-style-type: none"> 大規模ソフトウェアシステム設計: この試験セクションでは、ソフトウェアアーキテクトのスキルを評価し、大規模ソフトウェアシステムの設計と分析について学びます。受講者は、変化する要件に合わせて拡張・適応できる複雑なソフトウェアアーキテクチャを計画するための手法を探求します。この試験内容では、成長に対応し、増加するワークロード需要に対応できるシステム設計を作成するための手法を取り上げます。

>> Secure-Software-Design基礎訓練 <<

Secure-Software-Design試験の準備方法 | 効率的なSecure-Software-Design基礎訓練試験 | 一番優秀なWGUSecure Software Design (KEO1) Exam合格受験記

JpshikenはWGU試験問題集を提供するウェブサイトで、ここによく分かります。最もよくて最新で資料を提供いたします。こうして、君は安心してSecure-Software-Design試験の準備を行ってください。弊社の資料を使って、100%に合格を保証いたします。もし合格しないと、われは全額で返金いたします。

WGUSecure Software Design (KEO1) Exam 認定 Secure-Software-Design 試験問題 (Q12-Q17):

質問 # 12

Which security assessment deliverable identifies possible security vulnerabilities in the product?

- A. Threat profile
- B. List of third-party software
- C. SDL project outline
- D. Metrics template

正解: A

解説:

A threat profile is a security assessment deliverable that identifies possible security vulnerabilities in a product. It involves a systematic examination of the product to uncover any weaknesses that could potentially be exploited by threats. The process typically includes identifying the assets that need protection, assessing the threats to those assets, and evaluating the vulnerabilities that could be exploited by those threats. This deliverable is crucial for understanding the security posture of a product and for prioritizing remediation efforts.

References: The importance of a threat profile in identifying security vulnerabilities is supported by various security resources. For instance, Future Processing's blog on vulnerability assessments outlines the steps involved in identifying security vulnerabilities, which align with the creation of a threat profile¹. Additionally, UpGuard's article on conducting vulnerability assessments further emphasizes the role of identifying vulnerabilities as part of the security assessment process².

質問 # 13

Which software control test examines an application from a user perspective by providing a wide variety of input scenarios and inspecting the output?

- A. Black box
- B. Dynamic
- C. White box
- D. Static

正解: A

解説:

The software control test that examines an application from a user perspective by providing a wide variety of input scenarios and inspecting the output is known as black box testing. This testing method focuses on the functionality of the application rather than its internal structures or workings. Testers provide inputs and examine outputs without knowing how and where the inputs are worked upon. It's designed to test the system's external behavior.

* Black box testing is used to verify that the system meets the requirements and behaves as expected in various scenarios, including edge cases and incorrect input data. It helps in identifying discrepancies between the system's actual functionality and its specified requirements.

* This type of testing is applicable across various levels of software testing, including unit, integration, system, and acceptance testing. It is particularly useful for validating user stories and use cases during the software development process.

* Since black box testing treats the software as a "black box", it does not require the tester to have knowledge of the programming languages or the system's implementation. This allows testers to objectively test the software's behavior and performance.

: The concept of black box testing is well-documented and is a standard practice in secure software design, as outlined by sources such as LambdaTest1 and other industry best practices.

質問 # 14

Which secure software design principle states that it is always safer to require agreement of more than one entity to make a decision?

- A. Separation of Privileges
- B. Psychological Acceptability
- C. Least Privilege
- D. Total Mediation

正解: A

質問 # 15

Which secure coding best practice ensures sensitive information is not disclosed in any responses to users, authorized or unauthorized?

- A. Authentication and Password Management
- B. System Configuration
- C. Error Handling and Logging
- D. Input Validation

正解: C

質問 # 16

Which mitigation technique is used to fight against an identity spoofing threat?

- A. Require user authorization
- B. Encryption
- C. Filtering
- D. Audit trails

正解: A

解説:

To combat identity spoofing threats, a mitigation technique that is often used is requiring user authorization.

This involves implementing strong authentication methods to verify the identity of users before granting access to sensitive information or systems. Techniques such as two-factor authentication (2FA) or multi-factor authentication (MFA) are effective in reducing the risk of unauthorized access, as they require users to provide multiple pieces of evidence to confirm their identity, making it much

2026年Jpshikenの最新Secure-Software-Design PDFダンプおよびSecure-Software-Design試験エンジンの無料共有: https://drive.google.com/open?id=1_ITKVgZ-R4BT9s6tK2wfeHW-zqbgqqNT