



コースコード : GX-SWAD

税込価格 : 132,000円 (税抜価格 : 120,000円)

日数 : 1日間

トレーニング内容

Webアプリケーションにおいては新しい攻撃手法が次々に登場するわけではないため、セキュアに構築するためのセオリーがあります。つまり、攻撃に対応した安全なWebサイトを構築するためのセキュリティ要件は明確になっていて、それに対応した設計を施し、実装することで、攻撃の大半は防ぐことができます。

セキュリティの問題を修正する場合、開発後のフェーズになるほど高いコストが必要になってきます。要件定義や、設計段階での不備は、後の全フェーズに影響します。そのため、セキュリティの問題は設計段階までには解消しておくことが重要です。

セキュアWebアプリケーション設計士トレーニングでは、インターネットまたは内部ネットワーク向けに公開するWebアプリケーションをセキュアに構築するための要件や設計を学ぶことを目的としています。

(「認定セキュアWebアプリケーション設計士試験」の受験チケットがトレーニングとセットで提供されます。)

ここに注目!!

本トレーニングで学ぶことができる要件や設計方針は、特定非営利活動法人日本ネットワークセキュリティ協会の日本セキュリティオペレーション事業者協議会のセキュリティオペレーションガイドラインWG (WG1) と、OWASP Japan主催の共同ワーキンググループである『脆弱性診断士スキルマッププロジェクト』が公開している『Webシステム / Webアプリケーションセキュリティ要件書』に基づいています。

本トレーニングの目的や概要を以下から動画でご覧いただけます。

[SecuriST「認定セキュアWebアプリケーション設計士」ご紹介
\(グローバルセキュリティエキスパート社YouTubeチャンネル\)](#)

動画投稿時からアップデートがある可能性がございます。トレーニングの最新情報は本ページ記載内容をご参照ください。

ワンポイントアドバイス

本トレーニングで学ぶことができる要件や設計方針は、特定非営利活動法人日本ネットワークセキュリティ協会の日本セキュリティオペレーション事業者協議会のセキュリティオペレーションガイドラインWG (WG1) と、OWASP Japan主催の共同ワーキンググループである『脆弱性診断士スキルマッププロジェクト』が公開している『Webシステム / Webアプリケーションセキュリティ要件書』に基づいています。

受講対象者

このコースの受講対象者は次の通りです。

インターネット / 内部ネットワーク向けのWebシステム / Webアプリケーションに関わる下記の方が主な対象者となります。

- ・Webシステムの発注者
- ・Webアプリケーションの設計者・開発者

PCI DSSなどで要求されるOWASP Top 10などに基づいた安全なコーディング技法に関するトレーニングが要求されている場合などにも最適です。

前提条件

このコースを受講する前に受講者が習得しておく必要がある知識およびスキルは次のとおりです。

ご受講に際して特に前提条件はございませんが、下記のようなご経験がありますとよりこのコースの理解度が高まります。

- ・開発言語を使ったプログラム経験（例：VB, C++, PHPなど）
- ・OSのインストール経験（例：WindowsやLinux、MacOSなど）
- ・ホームルーターの設定経験

目的

このコースを修了すると次のことができるようになります。

本トレーニングでは下記のスキルを習得することができます。

- ・セキュアなWebシステム / Webアプリケーションを構築するために必要な知識
- ・発注者・開発者に必要なWebシステム / Webアプリケーションのセキュリティ要件
- ・上記セキュリティ要件を満たす設計の具体例

アウトライン

セキュリティ要件

セキュリティ要件の原則

Webアプリケーションのセキュリティ要件

セキュアWebアプリケーションの構築

認証の目的

NIST SP800-63B、Authenticatorのタイプ、AAL

Webアプリケーションで使う主な認証の種類

BASIC認証、DIGEST認証

フォームベース認証

認証を行うべき箇所

強いパスワードとは

パスワードのハッシュ化、salt、ストレッ칭

パスワードの作成について

ユーザーへのパスワード通知方法

パスワードの変更機能

パスワードリセット機能

秘密の質問について

認証実行時のエラー処理、ログ記録

アカウントロック

パスワードリスト攻撃対策

二要素認証、リスクベース認証

認可

認可の目的

アクセス制御の失敗例

アクセス制御方法

OpenID, OAuth, シングルサインオン

限定公開URL

FIDO認証

セッション管理

セッションIDの役割

Cookie

設定すべきCookieの属性値

セッションIDを利用した攻撃を防ぐ設計

セッションタイムアウトの設計



ログアウト機能

セッションIDの生成

CSRF対策

トークン方式

SameSite属性

入力処理

クライアント側での入力値チェック

Webアプリケーション側でのチェック

パラメーターについて

入力値の文字種や文字長の検証

文字エンコーディングの統一

入力値としてファイルを扱う場合

XMLファイルを扱う場合

デシリアライズについて

出力処理

出力処理で必要なこと

特殊文字のエスケープ処理

HTMLを生成する際の処理

HTMLのエスケープ処理

その他のスクリプト埋め込み原因の排除

クライアント側でHTMLを生成する際の処理

SQL文を組み立てる際の処理

JSONの生成

OSコマンドを呼び出す処理

HTTPレスポンスヘッダーについて

リダイレクタを使う際の注意事項

HTTPS

SSL/TLS

HTTPSの仕組み

証明書に対する攻撃

HTTPSを使う際の注意

証明書の種類、用途による使い分け

安全なプロトコルと暗号アルゴリズム

フィッシングサイトに対抗するには？

その他

エラーメッセージハンドリング

暗号アルゴリズムと乱数について

疑似乱数生成器

言語・フレームワーク・ミドルウェア・ライブラリなどの選定

ログの記録

ユーザーへの通知

Access-Control-Allow-Originヘッダーについて

クリックジャッキング対策

キャッシュ制御について

CAPTCHAについて

言語環境のセキュリティ設定

用意すべきドキュメント