

No	脆弱性の種類	対策の性質	状態	実施項目	解説
1	SQLインジェクション	根本的解決	■対応済み □未対応 □対応不要	SQL文の組み立ては全てブレースホルダで実装する	1-(i)-a
		根本的解決	■対応済み □未対応 □対応不要	SQL文の構成を文字列連結により行う場合は、アプリケーションの変数をSQL文のリテラルとして正しく構成する	1-(i)-b
		保険的対策	■対応済み □未対応 □対応不要	ウェブアプリケーションに渡されるパラメータはSQL文を直接指定しない	1-(ii)
		保険的対策	■対応済み □未対応 □対応不要	エラーメッセージをそのままブラウザに表示しない	1-(iii)
		保険的対策	■対応済み □未対応 □対応不要	データベースアカウントに適切な権限を与える	1-(iv)
2	OSコマンド・インジェクション	根本的解決	■対応済み □未対応 □対応不要	シェルを起動できる言語機能の利用を避ける	2-(i)
		保険的対策	□対応済み ■未対応 ■対応不要	シェルを起動する場合は引数を全てチェックする	2-(ii)
3	パス名パラメータ未チェック/ ディレクトリトラバーサル	根本的解決	■対応済み □未対応 □対応不要	外部からのパラメータでウェブサーバ内のファイル名を直接指定する実装を避ける	3-(i)-a
		根本的解決	□対応済み ■未対応 ■対応不要	ファイルを開く際は、固定のディレクトリを指定し、かつファイル名にディレクトリ名が含まれないようにする	3-(i)-b
		保険的対策	□対応済み □未対応 ■対応不要	ウェブサーバ内のファイルへのアクセス権限の設定を正しく管理する	3-(ii)
		保険的対策	□対応済み □未対応 ■対応不要	ファイル名のチェックを行う	3-(iii)
4	セッション管理の不備	根本的解決	■対応済み □未対応 □対応不要	セッションIDを推測が困難なものにする	4-(i)
		根本的解決	■対応済み □未対応 □対応不要	セッションIDをURLパラメータに格納しない	4-(ii)
		根本的解決	■対応済み □未対応 □対応不要	HTTPS通信で利用するCookieにsecure属性を加える	4-(iii)
		根本的解決	■対応済み □未対応 □対応不要	ログイン成功後に、新しくセッションを開始する	4-(iv)-a
		根本的解決	■対応済み □未対応 □対応不要	ログイン成功後に、既存のセッションIDとは別に秘密情報を発行し、ページの遷移ごとにその値を確認する	4-(iv)-b
		保険的対策	■対応済み □未対応 □対応不要	セッションIDを固定値にしない	4-(v)
		保険的対策	■対応済み □未対応 □対応不要	セッションIDをCookieにセットする場合、有効期限の設定に注意する	4-(vi)
		根本的解決	■対応済み □未対応 □対応不要	ウェブページに出力する全ての要素に対して、エスケープ処理を施す	5-(i)
		根本的解決	■対応済み □未対応 □対応不要	URLを出力するときは「http://」や「https://」で始まるURLのみを許可する	5-(ii)
		根本的解決	□対応済み □未対応 ■対応不要	<script>要素の内容を動的に生成しない	5-(iii)
		根本的解決	■対応済み □未対応 □対応不要	スタイルシートを任意のサイトから取り込めるようにしない	5-(iv)

5	クロスサイト・スクリプティング (XSS)	保険的対策	■対応済み □未対応 □対応不要	入力値の内容チェックを行う	5-(v)
		根本的解決	□対応済み □未対応 ■対応不要	HTMLテキストから構文解析木を作成しスクリプトを含まない要素のみ抽出する	5-(vi)
		保険的対策	□対応済み □未対応 ■対応不要	HTMLテキストからスクリプトに該当する文字列を排除する	5-(vii)
		根本的解決	■対応済み □未対応 □対応不要	HTTPレスポンスヘッダのContent-Typeフィールドに文字コード(charset)の指定を行う	5-(viii)
		保険的対策	■対応済み □未対応 □対応不要	CookieにHttpOnly属性を加え、TRACEメソッドを無効化する	5-(ix)
		保険的対策	■対応済み □未対応 □対応不要	XSS潜在的脆弱性対策としてブラウザ機能を有効にするレスポンスヘッダを返す	5-(x)
6	CSRF (クロスサイト・リクエスト・フォージェリ)	根本的解決	■対応済み □未対応 □対応不要	処理を実行するページをPOSTメソッドでアクセスするようにし、hiddenパラメータに秘密情報を挿入し検証する	6-(i)-a
		根本的解決	□対応済み □未対応 ■対応不要	処理を実行する直前のページで再度パスワードの入力を求め、正しい場合のみ処理を実行する	6-(i)-b
		根本的解決	■対応済み □未対応 □対応不要	Refererが正しいリンク元かを確認し、正しい場合のみ処理を実行する	6-(i)-c
		保険的対策	□対応済み □未対応 ■対応不要	重要な操作を行った際にその旨を登録済みメールアドレスに送信する	6-(ii)
7	HTTPヘッダ・インジェクション	根本的解決	■対応済み □未対応 □対応不要	ヘッダの出力を直接行わず、ウェブアプリケーションの実行環境や言語に用意されているヘッダ出力用APIを使用する	7-(i)-a
		根本的解決	■対応済み □未対応 □対応不要	改行コードを適切に処理するヘッダ出力用APIを利用できない場合は、改行を許可しないよう適切な処理を実装する	7-(i)-b
		保険的対策	■対応済み □未対応 □対応不要	外部からの入力の全てについて、改行コードを削除する	7-(ii)
8	メールヘッダ・インジェクション	根本的解決	□対応済み □未対応 ■対応不要	メールヘッダを固定値にして、外部からの入力はずべてメール本文に出力する	8-(i)-a
		根本的解決	□対応済み □未対応 ■対応不要	ウェブアプリケーションの実行環境や言語に用意されているメール送信用APIを使用する(8-(i)-aを採用できない場合)	8-(i)-b
		根本的解決	□対応済み □未対応 ■対応不要	HTMLで宛先を指定しない	8-(ii)
		保険的対策	□対応済み □未対応 ■対応不要	外部からの入力の全てについて、改行コードを削除する	8-(iii)
9	クリックジャッキング	根本的解決	■対応済み □未対応 □対応不要	X-Frame-Optionsヘッダフィールドを出力し、他ドメインのサイトからのframe要素やiframe要素による読み込みを制限する	9-(i)-a
		根本的解決	□対応済み □未対応 ■対応不要	処理を実行する直前のページで再度パスワードの入力を求め、正しい場合のみ処理を実行する	9-(i)-b
		保険的対策	□対応済み □未対応 ■対応不要	重要な処理は、一連の操作をマウスのみで実行できないようにする	9-(ii)
10	バッファオーバーフロー	根本的解決	■対応済み □未対応 □対応不要	直接メモリにアクセスできない言語で記述する	10-(i)-a
		根本的解決	■対応済み □未対応 □対応不要	直接メモリにアクセスできる言語で記述する部分を最小限にする	10-(i)-b
		根本的解決	■対応済み □未対応 □対応不要	脆弱性が修正されたバージョンのライブラリを使用する	10-(ii)

11	アクセス制御・認可制御の欠落	根本的解決	■対応済み □未対応 □対応不要	アクセス制御機能によるパスワード等の秘密情報入力が必要とする認証機能を設ける	11-(i)
		根本的解決	■対応済み □未対応 □対応不要	認可制御を実装し、ログイン中の利用者が他人になりすましてアクセスできないようにする	11-(ii)