セキュア開発チェックリスト

システム開発現場における、セキュア開発のためのチェックリストです。

開発するシステムにより対応すべき事項が異なるため、すべてに対応することを目的とするものではなく、本来対応すべき事項の抜け漏れを軽減する事を目的としています。

本チェックリストは、以下のガイドラインを参照して作成しています。

- OWASP Application Security Verification Standard（OWASP ASVS）
- OWASP Mobile Application Security Verification Standard（OWASP MASVS）
- OWASP Top 10 Proactive Controls
- OWASP Cheat Sheet Series

一般	1. SAST（Static Application Security Testing）診断を受けており、必要な指摘はすべて対応している
認証	1. ユーザを変えて認証機能のテストを実施している
	2. ログイン成功時および失敗時に、システムがログを記録している
	3. 認証にはCognitoなどの信頼性の高い認証サービスを使用している
	4. 認証サービスを使用していない場合は、フレームワークの標準機能を使用している
	5. 一定回数認証に失敗した場合、ロックアウトする
	6. 多要素認証を設定できる
	7. パスワードはハッシュ化されて保存されている
	8. パスワード変更、普段と異なるIPアドレスからのログイン、アカウントロックアウト時にユーザと管理者に通知される
	9.ユーザのパスワードまたは権限が変更された際に、操作を実施したユーザを特定できるログが記録される
	10. パスワードリセットに「秘密の質問」を使っていない
認可	1. ユーザおよび権限を変更して認可機能のテストを実施している
	2. 権限違反でアクセスを拒否した場合に拒否したユーザ名等がログに残っている
	3. 権限マトリクスなどで、ユーザ、ロールごとの許可される操作（例えば、他人ユーザ情報の閲覧）が文書化されている
	4. 例外が発生した場合、アクセスを拒否している
	5. 最小権限の原則にのっとっている
	6. 改ざん耐性のあるメタデータを用いて認可を判断している
	7. 認可制御は、サーバ側で実装しておりフロント側の制御のみに頼っていない
	8. 管理画面は、信頼できるエンドポイントまたは場所（例えば特定のIPアドレス）からのみアクセスできる
セッション管理	1. セッション管理機能は、フレームワークやライブラリの実装を使っていて、独自実装していない
	2. 信頼できるバックエンドで、セッショントークンを検証している
	3. セッション開始から12時間経過後に再認証が必要となるよう設定されている（12時間は当社ベースライン。各システムの固有の数字に置き換えてください）
	4. 30分間操作がない場合、再認証が必要となるよう設定されている（30分は当社ベースライン。各システムの固有の数字に置き換えてください）
入力値検証	1. プログラムで扱う入力値をすべて検証している（POSTのみならず、GETやPathパラメータ、Cookie、ファイルなどを含む）
	2. 入力値の検証は、可能な限り許可リスト（ホワイトリスト）方式を採用している
	3. 許可リストで検証できない場合、サイズ、文字種、フォーマットの各観点で入力値を検証している
	4. 入力値について、境界値テストを実施している
	5. 入力検証で、NGと判定した場合にNG内容がログに残っている
	6. 出力値が定義されたプロパティおよび型に準拠していることを検証している

エンコードとエスケープ	1. HTMLに出力するときは、自動的にエンコードするライブラリやフレームワークを使っている
	2. DBアクセスは開発チームで決められたライブラリを使って実装している
	3. SQLやHTML、URL、ファイルパス、OSコマンド、JSONなど構造化されたテキストを文字列置換や文字列結合で構成せず、専用のライブラリや関数で実装している
暗号化	1. すべての通信で、SSL/TLS通信（https、SSH、SFTPなど）などの暗号通信を強制している
	2. TLSバージョン1.2以上を使っている（SSL 1.0,2.0,3.0およびTLS 1.0,1.1は非推奨）
	3. SSL/TLSのCipher Suiteなどの設定値は、推奨の設定値を使っている
	4. Cipher Suiteなどの設定を手動で構成する場合は、Mozilla SSL Configuration Generatorを使用して推奨設定を採用している
	5. システムが扱う機密データを特定し、保管場所、削除タイミング、アクセスログの保管場所、操作履歴の保管場所を文書化している
	6. 機密データは、暗号化して保管している
	7. 機密データに対するアクセスや操作履歴は、ログとして記録している
	8. 暗号化は、可能な限りプログラムで実装せず、RDSやS3などのインフラサービスの暗号化機能を利用している
	9. プログラムで暗号化している場合、その暗号鍵はKMSなどのシークレット管理ソリューションを使っている
シークレット管理	1. シークレットは、フロントWebアプリケーションのソースコードや環境変数に書き込まず、シークレットが必要な処理は、バックエンドで実装している
	2. シークレットは、ソースコードに書き込まず、環境変数、パラメータストア、Secret Managerなどのソリューションで管理している
	3. Gitにシークレットをコミットせず、アクセス制限された安全な手段（例：SharePoint）で共有している
過度な使用	1. 下記のように重要な機能には、過度に使用されないように一定の制限を設けている a. ECであれば1人のユーザの1日あたりの購入数や購入金額の上限を設けている b. ストレージサービスであれば、1人がアップロードできるファイル数やサイズに上限を設けている
ファイル関連	1. ファイルの入力検証は、ファイルサイズ、拡張子、mime-typeを検証している
	2. ファイルを保存する際は、予測困難なランダムなファイル名に変更して保存している
	3. オリジナルのファイル名が必要な場合は、入力検証をしたうえでメタデータやDBなどで保管する
	4. 圧縮ファイル (zip, gz, docx, odt など) を展開する前に最大許容非圧縮サイズおよび最大ファイル数を検証している
	5. ユーザから送信されたファイルを実行していない（ユーザから受け取ったexeファイルなど）
ログ	1. 機密データをログに記録していないことを確認している
	2. 記録されているログの種別、保管場所、保管期限、機密データの有無が文書化されている
	3. 少なくとも、アクセスログ、アプリケーションエラーログ、DBのエラーログ、各コンポーネントのエラーログが残っており、有事の際に調査できる
	4. ログから脆弱性診断の実施状況を把握できる（例：ロックアウト発生、認可拒否／認証失敗／入力検証の失敗の急増など）
バックアップ	1. 必要なデータのバックアップを取得している
	2. バックアップの保管場所、タイミング、保持期間が文書化されている
	3. バックアップからのデータ復元が可能であり、復元訓練を実施している
	4. 必要に応じて、別のアカウントや地域にバックアップを保持している

モニタリング	1. システムの死活監視を実施しており、異常値を検知した際に、即時にアラート通知が行われるよう設定している a. httpレスポンス b. CPU,メモリ,Disk使用率 c. エラーログ（アプリケーションやDBエラー）
	2. パフォーマンス目標にかかわる指標を定期的に観測している a. エラー率,停止時間 b. レイテンシ c. スループット
コンポーネントの脆弱性	1. コンポーネントの脆弱性をチェックし、危険な脆弱性がないことを確認している
	2. 簡単にコンポーネントの脆弱性の有無を可視化することができる（コマンド実行やInspectorなどのツールを利用して）
インフラ	1. インフラ構成図が最新の状態で文書化されている
	2. インフラ構成は、リファレンスアーキテクチャをもとに設計されている
	3. インフラ構成の作成、変更時には、セカンドオピニオンとして誰かのレビューを受けている
	4. インフラを管理する管理コンソールへのログインは、多要素認証を設定している
	5. インフラを管理する管理コンソールへのログインは、アクセス元IPアドレスで制限されている
	6. インフラを管理する管理コンソールへのログインは、個人ごとのアカウントを発行している
	7. WAFが導入されており、既知の悪意あるIPアドレス（例：AWSのReputation List）からのアクセスを遮断している
	8. AWSの場合、以下を利用するリージョンでONにしている a. Guard Duty b. Security Hub c. Inspector d. Cloud Trail
その他	1. 本番環境にテストコードが含まれていない
	2. アプリケーションのビルドとデプロイはが自動化されており、再現性のある安全な手順で実施できる
	3. テストが自動化されており、再現性のある手順で簡単に実施できる

チェックリストの項目に不安、懸念がある場合は、セキュア開発トレーニングの受講をおすすめします。

セキュア開発トレーニングはこちら

https://www.proactivedefense.jp/services/training/secure-dev-training

免責事項

本チェックリストは、セキュアなソフトウェア開発を支援する目的で提供されるものであり、その内容の正確性、完全性、有効性を保証するものではありません。

本チェックリストの利用により発生したいかなる損害（直接的・間接的・偶発的・結果的損害を含む）についても、当社は一切の責任を負いません。

利用者は自己の責任において本チェックリストを使用するものとし、必要に応じて専門家の助言を受けるなど、適切な判断のもとでご利用ください。

ライセンス