大学共同利用機関法人 情報・システム研究機構
国立情報学研究所
National Institute of Informatics

トップエスイー ソフトウェア開発実践演習

Webアプリケーションにおける セキュアデザインパターンを使用した 設計の実践と効果の分析

NECソリューションイノベータ株式会社　林　昌紀　mas-hayashi@ys.jp.nec.com

開発における問題点

セキュリティ設計により，機能面の設計にも後戻りが発生し得るため，セキュリティ設計は早く実施することが望ましい．
しかし，設計がある程度進まないと保護するポイントが決まらないため，設計できないという課題がある．

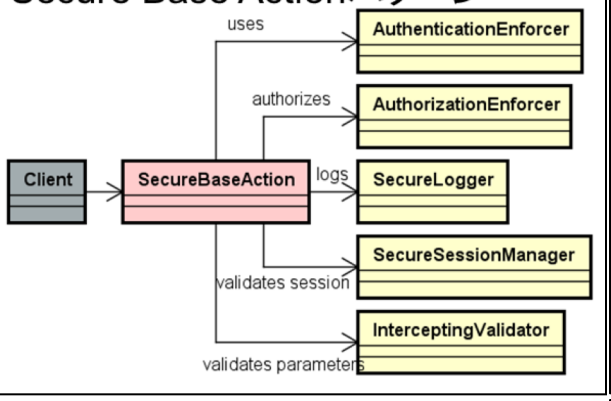
手法・ツールの適用による解決

セキュリティについての設計ノウハウをまとめた『セキュアデザインパターン』を使用することで課題を解決する．

セキュアデザインパターンの使用

今回使用したセキュアデザインパターン
- Secure Base Actionパターン
  セキュリティ対策のとりまとめとなるクラスを作ることで，対策の実施漏れを防ぐ．
- Obfuscated Transfer Objectパターン
  パスワード等の重要なデータをシステム内で適切に扱うようにできる．
- Account Lockoutパターン
  規定回数のログイン失敗により，アカウントをロックする．
- Client Data Storageパターン
  クライアント側で重要なデータを持つ場合の安全なデータの保持方法．

Secure Base Actionパターン

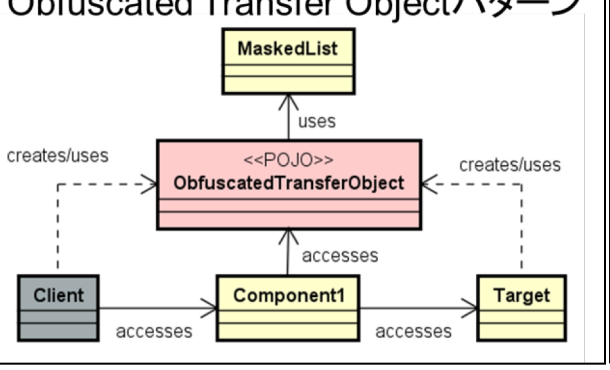


Account Lockoutパターン

- ログイン失敗回数を記録する
- ログイン成功時に0に戻す
- ログイン失敗回数が閾値を超えると，アカウントをロックする
- ロックされたアカウントに対して認証が試行されるとシステムは認証を処理せず，ログに記録する

…

Obfuscated Transfer Objectパターン



Client Data Storageパターン

- データを送信する時に暗号化を行う
- 共通鍵暗号を使用し，共通鍵をクライアントとサーバで共有する
- データを送る際には暗号化し，受信側は復号化して使用する

…

セキュアデザインパターンを調査　⇒　セキュリティ課題が明確になる
セキュアデザインパターンを適用　⇒　セキュリティ課題に対して対策できる

考察

セキュアデザインパターンの効果	課題
• セキュアデザインパターンを調査・適用することにより，設計フェーズの早期からセキュリティ設計が可能になる	• セキュリティ課題を洗い出すためにセキュアデザインパターンを一通り調査する必要がある • GoFのデザインパターンほど整理されておらず，知っておくべき必要十分なパターンが不明である