テストポリシーを考慮したセキュリティ要件獲得手法の提案

野口 睦夫†‡　　田中 英彦§

†NEC ソリューションイノベータ株式会社
136-8627 東京都江東区新木場一丁目 18 番 7 号
mut-noguchi@uf.jp.nec.com
‡情報セキュリティ大学院大学 客員研究員
221-0835 神奈川県横浜市神奈川区鶴屋町 2 丁目 14– 1
mgs115502@iisec.ac.jp
§情報セキュリティ大学院大学 情報セキュリティ研究科
221-0835 神奈川県横浜市神奈川区鶴屋町 2 丁目 14– 1
tanaka@iisec.ac.jp

あらまし 近年，Webアプリケーションに対する不正アクセスが日常的に発生し，開発現場でもセキュリティ対策の重要性が認識されて来ている．しかし，開発プロセスの上流工程では十分なセキュリティ対策が行われず，下流工程の受入試験でも適切に脆弱性の有無を確認できていない現状がある．そのため，上流工程と受入試験を担当する発注者を対象に，システム機能ベースのセキュリティパターンを拡張し，テストポリシーを考慮しながらセキュリティ要件を獲得する手法を提案する．この提案により，定義したセキュリティ要件を受入試験時にどのような方針で確認すれば良いかが識別可能となり，サービス公開時の安全性向上に繋がることが期待される．

Proposal of security requirements acquisition technique for considering test policies

Mutsuo Noguchi†‡　　Hidehiko Tanaka§

†NEC Solution Innovators, Ltd.
1-18-7 Shinkiba, Koto-ku, Tokyo, 136-8627, JAPAN
mut-noguchi@uf.jp.nec.com
‡Institute of Information Security, Visiting Researcher
2-14-1 Tsuruya-cho, Kanagawa-Ku, Yokohama-Shi, Kanagawa 221-0835 JAPAN
mgs115502@iisec.ac.jp
§Institute of Information Security, Graduate School of Information Security
2-14-1 Tsuruya-cho, Kanagawa-Ku, Yokohama-Shi, Kanagawa 221-0835 JAPAN
tanaka@iisec.ac.jp

Abstract In recent years, unauthorized access to the Web application is occurring on a daily basis. Therefore, there is recognized the importance of security measures. However, sufficient security measures are not carried out in the upstream process of the development process. And, it is not be detected the vulnerability in acceptance testing. Therefore, targeting orderer responsible for upstream process and acceptance testing, we propose the available techniques, security requirements acquisition technique for considering test policies that extends the system function based security pattern. With this proposal, it would be seen the test policy of pre-defined security requirements at the time of acceptance testing.

1　はじめに

近年，インターネットバンキングやスマートフォンを用いた証券･外国為替証拠金取引など，個人や経済の根幹部分にまで情報サービスが普及し，暮らしが便利になっている．[1] 加えて，スマートフォンやタブレット端末などのインターネットに接続可能なスマートデバイスが急速に普及し，個人の生活とインターネットの結びつきはますます強くなっている．

しかし，インターネット上では，不正アクセスが日常的に行われており，Web サイトの改ざんやサービス停止などのインシデントも多数報告されている．[2]それにともない，Web サイトを構築する OS やネットワーク機器，セキュリティ機器などのプラットフォームだけではなく，Web サイト上のサービスを実現するソフトウェア，すなわち Web アプリケーションに対する情報セキュリティ対策技術の重要性が認識されてきた．[3] 一度 Web サイト上で公開されたサービスは，幾度もの更新を繰り返して，継続的にサービス提供されるが，その更新サイクルのなかで，セキュリティ対策の重要性が認識されていても，Web アプリケーションに脆弱性が埋め込まれることも多い．筆者が脆弱性診断を実施してきた経験のなかでも，新規に開発された Web アプリケーションでは存在しなかった脆弱性が，更新時にセキュリティ対策の漏れにより，脆弱性が埋め込まれてしまう事例を体験している．その際に要件定義書や設計書を確認する機会もあるが，機能要件は設計書並に詳細に記載されていても，セキュリティ要件が明確かつ詳細に定義されていることは，ほとんど存在していないのが現状である．また，セキュリティ要件が定義されていても，開発試験フェーズや受入試験フェーズにおいて，システム機能要件と異なり，セキュリティ要件に対して，テストポリシーの設計ができないという相談を受けることもある．

本稿で提案するテストポリシーを考慮したセキュリティ要件獲得手法では，システム機能要件をベースにセキュリティパターンを使うことでセキュリティ要件を獲得するとともに，各セキュリティ要件どおりに実装されているかを各試験フェーズで確認するためのテストポリシーを導出していく．

以降，筆者が本研究を行う動機となった背景やセキュリティ要求分野における関連研究について説明し，提案手法の必要性について言及する．次に，本稿の提案手法について説明し，実在するサイトに対する Web アプリケーション脆弱性診断を通して，受入試験におけるテストポリシーを考慮することの意義について言及する．最後に，本稿での成果と今後の方針について説明する．

2　背景と関連研究

セキュリティ開発ライフサイクルについてはマイクロソフト社の文書[4]が著名であるが，その中でも筆者の経験上，Web アプリケーションの開発プロセス内に組み込まれることがあるセキュリティ対策は下図のとおりである．

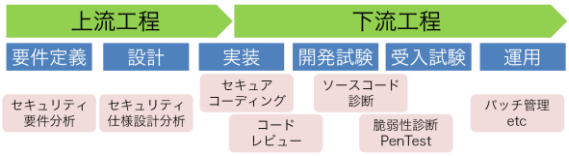

図 1 開発プロセスとセキュリティ対策の対応

しかし，主に実施されているセキュリティ対策は，「設計」「実装」「開発試験」「運用」の各開発フェーズに対応するものだけであり，「要件定義」や「受入試験」ではあまりセキュリティ対策が実施されていない．そして，実施されているセキュリティ対策は，Web アプリケーション開発を請け負う開発者側の知見によって行われていることが多いのが現状である．

筆者の経験上，Web アプリケーションの開発プロジェクトにおいて，各開発プロセスにおける発注者と開発者の責任範囲は図２に示すような関係となっていることが多い．発注者側は『実現したい機能を決める要件定義』[5]『受注者側によって実装された Web アプリケーションに対して要件の充足度を確認する受入試験』[6]，運用の各プロセスに責任を持っている．そのため，セキュリティ要件を明確に示し，受入試験でセキュリティ上問題のない実装になっているか確認の必要があるが，一章で述べたようにこれらは殆ど実施されていないのが現状である．

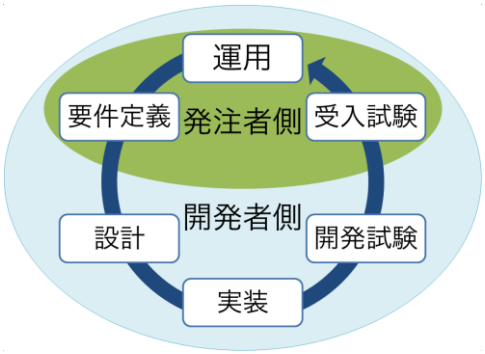

図 2 発注者と受注者の関係図

大久保らはセキュリティ要求分析が実際に行われない主な理由として，以下を挙げている．[7]

1. 要求分析においてセキュリティが重要視されていない
2. セキュリティ要求分析のためのノウハウを持つ開発者がいない
3. 要求分析のために用意された時間が短く，セキュリティ要求分析を行う時間が確保出来ない

1.に関しては，開発に関わるステークホルダーの情報セキュリティに対する意識や知識不足，2.や 3.についても，開発体制のなかで情報セキュリティ対策に必要な知識を持つ人材が不足していることが原因として考えられる．加えて 3.については，セキュリティ要求分析に必要な脅威分析などの作業が，時間と費用の両面で大きな負担となるため，従来の要求分析や要件定義に与えられた時間や体制のなかでは，完遂が困難になっていることが挙げられる．

したがって，現状の問題を解決するためには，発注者側が利用可能であり，情報セキュリティの知識や作業期間があまり求められない手法が必要であると考える．

専門的な知識の不足に対しては，パターン[8]を用いる手法があり，セキュリティに対しては，セキュリティパターン[9]を用いた手法が数多く提案されている．

上流工程で利用可能な多様なセキュリティパターンがまとめられた文献として，Markus Schumacher らの[10]があるが，専門知識やかなりの時間が要求されることが課題として指摘されている．他にもセキュリティパターンについては，様々な提案がなされており，吉岡らによる[11]では，セキュリティパターンをソフトウェア開発のライフサイクルの観点で以下の 3 つに分類している．

1. 要求分析に関するセキュリティパターン
2. 設計のためのセキュリティパターン
3. 実装に関するセキュリティパターン

そして，以下の傾向が読み取れるとしている．

- 要求分析工程では，攻撃に関するパターンが多く，対策に関するパターンが少ない
- アーキテクチャ設計や詳細設計工程では，攻撃に関するパターンと次にセキュリティ仕様に関するパターンが少ない
- 実装工程では，攻撃や対策に関するパターンが多いが，セキュリティ仕様に関するパターンが少ない

吉岡らの[11]では，下流工程を意識した改善アプローチが必要としており，要件定義フェーズへの言及がされていない．

大久保らの[12][13]では，一般の Web アプリケーションでは，典型的な脆弱性や攻撃が解決策も含め知識として整理され公開されている点に着目し，Web 領域に特化したセキュリティ要求パターンが提案されている．しかし，大久保らの提案するユースケース図を拡張した AsseMis 記法を用いるため，提案手法を含めた情報セキュリティに関する専門性を有した担当者が必要となるため，知識不足の課題解消には至っていない．

前述の知識不足という課題に対応するため，宇野は機能要求分析で得られるシステム機能要件に基づき，セキュリティ要件を獲得するためのセキュリティパターンを提案している．[14]システム機能要件に基づくことで，非機能要求分析の作業工程であるセキュリティ要求分析を専門の担当者を置かずに，機能要求分析と連動してセキュリティ要件が獲得出来るため，情報セキュリティに関する専門性を有する必要性が軽減できることを利点としている．

本稿が対象とする利用者である発注者は，専

門的な知識の不足から開発者の支援に依存する傾向があるため，宇野の提案手法の利点は，本稿の目的と一致している．

一方で，セキュリティ要件の獲得とその発注者側での確認作業となる受入試験を連動させた研究は，調査した範囲では確認できなかった．そのため，受入試験でセキュリティ要件が正しく実装されているかを確認するため，各セキュリティ要件に対応するテストポリシーを考慮した手法が必要であると考える．以下，3 章でこの課題に対応した手法として，提案手法について説明する．

3　提案手法について

3.1　提案手法の考え方

宇野の提案手法[14]では，セキュリティ要求分析を行う担当者に求められる情報セキュリティ知識の低減を図っている点で，他の研究手法よりも利用価値が高いと考える．しかし，導出されたセキュリティ要件が，要件定義フェーズ以降の各開発フェーズにおいて，どのように反映されるかを正確に把握することは困難と想定される．そのため，受入試験フェーズにおいて，発注者側が機能要件の観点からテストポリシーを作成することは可能であっても，適切にセキュリティ要件に沿ったテストポリシーを作成することは困難であると推測される．

本稿の提案手法では，前述の懸念事項を考慮し，以下の流れでセキュリティ要件とともにテストポリシーの作成を行う．また，図 3 に本稿提案手法の流れを図示する．

1. 要件定義フェーズでシステム機能要件に基づいてセキュリティ要件を獲得する際に，各セキュリティ要件に関連するベースとなるテストポリシーを導出する
2. 設計フェーズや実装フェーズにおいて開発者に確定した仕様をフィードバックさせる
3. フィードバック情報を考慮し，テストポリシーを更新する

各開発フェーズにおける変更内容をテストポリシーに反映できるため，受入試験フェーズで適切に試験を実施するためのテストポリシーが獲得可能となり，そのテストポリシーを利用して発注者が必要なテストケースを設計することが可能になるため，受入試験フェーズでセキュリティ要件が正しく実装されているかを効果的に確認可能になることが期待される．

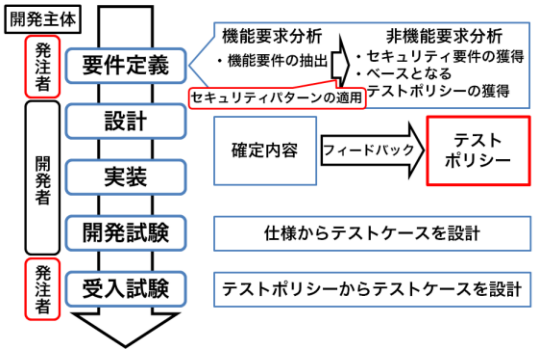

図 3 提案手法の適用イメージ

3.2　パターンの記述内容

本手法で用いるセキュリティパターンに記述する項目とその内容の定義を以下に示す．

- 名称
  セキュリティパターンの名称を記述
- 状況
  セキュリティパターンを適用する状況を記述
- 問題
  想定する状況において発生する問題を記述
- 解法
  問題を解決するために有効な方法や考え方を記述
- テストポリシー
  解法に沿って実装されているかを確認するために試験を実施する際に必要となる方針を記述

上記定義に従い作成した「外部プログラム呼び出し」に関するパターンの例を以下に示す．

- 名称
  外部プログラム呼び出し
- 状況
  本パターンは，Web アプリケーションから直接サーバ上のプログラムやシェルコードを実行するような実装が想定される

場合に適用される.

– 問題
  ① CWE-78：OS コマンドインジェクション
  ② CWE-94：コードインジェクション
  ③ CWE-200：情報漏えい
– 解法
  A： OS コマンドインジェクション対策 (①, ③)
  B： コードインジェクション対策 (②, ③)
– テストポリシー
  ● データ妥当性確認テスト（A, B）
    ※外部入力を元に任意のコマンドがサーバ上で実行されないことを確認

宇野の提案するセキュリティパターン[14]に，テストポリシーを追加するとともに，問題にCWE[15]を用いるように拡張することで，曖昧性を排除している.

3.3 パターン作成の流れ

「外部プログラム呼び出し」に関するパターンをはじめとして，提案手法で用いるセキュリティパターン作成の流れを以下に示す.

① 情報収集
② Web アプリケーションの機能抽出
③ 各機能の問題と解法の収集
④ 問題および解法に対するテストポリシーの収集
⑤ セキュリティパターンの作成
⑥ パターンの確認及び修正
⑦ 用語集の作成

②においては, Web アプリケーション機能モデルを作成した.

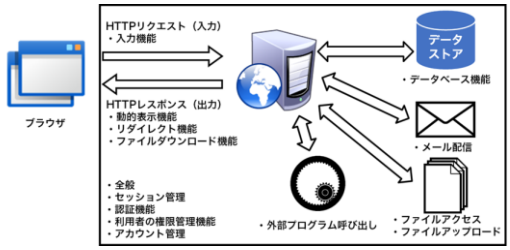

図 4 Web アプリケーション機能モデル

この機能モデルに従って，各システム機能の処理の流れを整理することで，図 5 のように処理の流れを把握し，セキュリティパターンの適用容易性を高めている.

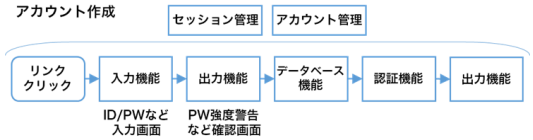

図 5 機能モデルベースの[アカウント作成]機能の処理

なお，テストポリシーに対して開発者が各開発フェーズでフィードバックすべき内容については，用語集で詳細を記載している.

3.4 作成したセキュリティパターン

最終的に作成した「テストポリシーを考慮したシステム機能ベースセキュリティパターン」一覧の一部を以下に示す.

パターン名称	説明
全般	Web アプリケーションを開発する上での基本的な方針に関するパターン
パッケージシステムの適用	Web アプリケーションを開発する際に，パッケージシステムを利用する上での基本的な方針に関するパターン
入力機能	ブラウザから受け取るパラメータの受付処理機能
セッション管理	同一利用者による一連の複数のリクエストを一意に識別するための機能（Cookie の利用含む）
Cookie の利用	
認証機能	利用者を識別するための機能
ログイン認証機能	利用者に ID とパスワードを入力させ，利用者本人であることを確認する機能
クライアント認証機能	クライアント証明書を要求し，システムから利用者向けに発行されたかを確認することで，利用者本人であることを確認する機能
利用者の権限管理機能	認証された利用者の権限を確認し，付与する機能
外部プログラム呼び出し	シェル経由での OS コマンド実行など，Web アプリケーション外のプログラムを実行する機能
データベース機能	データベースにアクセスする機能
メール送信機能	登録あるいは指定されたメールアドレス宛にメールを送信する機能

表 1 セキュリティパターン一覧（一部）

4 評価

本稿では, テストポリシーを考慮することで，受入試験フェーズのセキュリティ対策にどの程度の効果があるか評価を行った.

受入試験フェーズにおけるセキュリティ対策としては，図 1 で示した脆弱性診断が挙げられる．そこで商用脆弱性検査ツールである IBM Security AppScan を用い，AppScan 評価用サイト『Altoro Mutual』を対象とした受入試験を想定し，本提案手法の効果測定を実施した．評価項目としては，「脆弱性検出結果」「検査時間」「検査時の総リクエスト数」としている.

評価では，対象の『ログイン』機能および『申請手続き』機能について，各機能の処理の流れを分析し，図 5 と同様に処理の流れを整理し，セキュリティパターンの適用を行った.

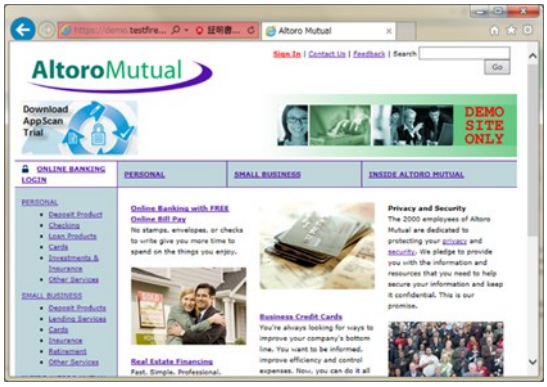

図 6 『Altoro Mutual』のトップ画面

セキュリティパターンによって得られたテストポリシーをもとに，以下の 2 通りの検査手法で効果を測定し，結果を表 2 および 3 に示す．なお，表 2 については，両検査手法で同じ結果が得られたことを示している．

– 従来の検査手法
  ブラックボックステストが前提となるため，全ての検査用テストポリシーを有効に設定し，脆弱性診断を実施
– 本提案手法を適用した場合の検査手法
  獲得したテストポリシーに基づき，必要なテストポリシーを把握．必要な検査用テストポリシー以外を従来の検査手法で定義した設定から除外し，脆弱性診断を実施

検出されたテスト項目	重大度	件数
SQL インジェクション	高	2
SQL インジェクションを使用した認証バイパス	高	2
クロスサイト・スクリプティング	高	2
予測可能なログイン証明書	高	1
暗号化されたセッション（SSL）Cookie のセキュア属性の不備	中	4
キャッシュ可能な SSL ページの発見	低	4
セッション Cookie に HttpOnly 属性がありません	低	3
データベース・エラー・パターンを検出	低	3
パスワード・フィールドで，HTML の autocomplete 属性が無効になっていません	低	2
暗号化が実行されていない	低	4
管理ページへの直接アクセス	低	1
HTML コメントによる秘密情報の開示	情報	2
アプリケーション・エラー	情報	2
アプリケーション・テスト・スクリプトを検知	情報	2

表 2 脆弱性検出結果

評価パターン	検査時間	送信された HTTP リクエスト数
従来の検査手法	34 分 6 秒	7097 リクエスト
本提案手法を適用した場合の検査手法	19 分 43 秒	4675 リクエスト

表 3 検査時間と検査時の総リクエスト数

今回実施した効果測定では，提案手法を適用した場合に，検査時間が約 40%短縮し，検査時の総リクエスト数が約 35%削減する効果を確認することができた．

このことから，脆弱性診断の品質を低下させずに，一定時間内に検査可能なページ数や機能数の増加という効果があったと判断出来る．

このことは，Web アプリケーション全体に対して，脆弱性診断を実施することが理想であるが，脆弱性診断に割り当てられる期間やコスト面から部分的にしか行われていなかった現状に効果的に作用すると考えられる．

5 まとめと考察

本稿では，テストポリシーを考慮したセキュリティ要件獲得手法が必要な背景を提示し，提案手法の考え方やセキュリティパターン作成の流れについて説明した．その上で，作成したセキュリティパターンを適用した場合の評価結果について示し,本手法が受入試験フェーズにおいて有効に働くことを明らかにした．

2015 年に入り，Web システムに関係するシステム損害賠償請求事件の判決[16]のように開発者にとって厳しい判決が話題になったが，これによって，開発者側が自衛策として発注者責任を明確に求めるようになると，本提案手法の有用性も高まると考える．

加えて，開発者にとっても，明確にセキュリティ要件が提示され，テストポリシー作成のためのフィードバックの過程で，開発者自らも開発試験で実施すべき項目を明確にすることができ，実装時に予め必要な対策を講ずることができるため, Web アプリケーション全体のセキュリティ向上と，それに伴う訴訟リスクの低減という利点があると考察可能である．

一方で, 実際の開発現場に適用するためには，開発プロセス全体をとおしての本提案手法を活用するためのアプローチの整理や，システム更新時への対応，セキュリティ要件毎の優先度付けが必要であると考えられるため，今後の検討課題としたい．

参考文献

[1] 吉岡信和. 1. セキュリティ要求工学の概要と展

望 (<特集> セキュリティ要求工学の実効性). 情報処理, 2009, 50.3: 187-192.
[2] JPCERT/CC インシデント報告対応レポート [2015 年 4 月 1 日～2015 年 6 月 30 日], https://www.jpcert.or.jp/pr/2015/IR_Report20150714.pdf
[3] McGraw, Gary. "Software security." Security & Privacy, IEEE 2.2 (2004): 80-83.
[4] Microsoft Security Development Lifecycle (SDL) Process Guidance - Version 5.2, http://www.microsoft.com/en-us/download/details.aspx?id=29884, Microsoft, 2012-5-23
[5] SEC BOOKS：実務に活かす IT 化の原理原則 17 ヶ条, http://www.ipa.go.jp/sec/publish/tn10-001.html,独立行政法人情報処理推進機構, 2010-10-12
[6] ソフトウェアテスト標準用語集 日本語版 Version 2.2.J03, http://jstqb.jp/dl/JSTQB-glossary.V2.2.J03.pdf, International Software Testing Qualifications Board 用語集作業班
[7] 大久保隆夫, 田中英彦 : セキュアなアプリケーション開発のための要求・デザインパターンの提案, 情報処理学会研究報告. CSEC, [コンピュータセキュリティ] 2009(20), 241-246, 2009-02-26
[8] 鷲崎弘宜. (2011). ソフトウェアパターン概観. 情報処理, 52(9), 1119-1126.
[9] Markus Schumacher, Eduardo Fernandez, Duane Hybertson, and Frank Buschmann. 2006. Security Patterns: Integrating Security and Systems Engineering. John Wiley & Sons.
[10] Markus Schumacher, Eduardo Fernandez, Duane Hybertson, and Frank Buschmann. 2006. Security Patterns: Integrating Security and Systems Engineering. John Wiley & Sons.
[11] 吉岡信和, 鷲崎弘宜, & 丸山勝久. (2007). セキュリティパターン技術に関する研究動向 (検証/セキュリティ). 情報処理学会研究報告. ソフトウェア工学研究会報告, 2007(107), 39-46.
[12] 大久保隆夫, and 田中英彦. "効率的なセキュリティ要求分析手法の提案." 情報処理学会論文誌 50.10 (2009): 2484-2499.
[13] Okubo, Takao, and Hidehiko Tanaka. "Web security patterns for analysis and design." Proceedings of the 15th Conference on Pattern Languages of Programs. ACM, 2008.
[14] 宇野健二（2011）. Web アプリケーション開発におけるシステム機能ベースセキュリティ要求分析. 情報セキュリティ大学院大学情報セキュリティ研究科修士論文（未公刊）
[15] CWE - CWE List List Version 2.8, http://cwe.mitre.org/data/index.html，2015 年 1 月 11 日閲覧
[16] 情報流出に係るシステム損害賠償請求事件（東京地裁 平成 26.1.23），http://www.softic.or.jp/semi/2014/5_141113/op.pdf
[17] 野口睦夫（2014）.テストポリシーを考慮したセキュリティ要件獲得手法の提案. 情報セキュリティ大学院大学情報セキュリティ研究科修士論文（未公刊）