Web アプリケーション更新時における セキュリティ要件獲得手法の検討

野口睦夫† 大久保隆夫† 田中英彦†

あらまし 近年，インターネットに公開されている Web サイトに対して，日常的に不正アクセスが行われている．それにともない，情報セキュリティ対策技術の重要性が認識されてきた．しかし，発注者が Web アプリケーションのセキュリティ対策を受注者に任せっきりの状況があり，発注者側には，セキュリティ要件を受注者に提示し，それらが正しく実装されているかを確認する責任があるが，現実的に開発プロセスの上流からセキュリティを考慮した対策が行われていないことが，脆弱性を生む一因となっている．筆者らは，必要とされるセキュリティ知識を極力減らすことを目的としたシステム機能ベースセキュリティパターンを拡張し，セキュリティ要件獲得時に受入試験での試験方針を獲得可能な手法を提案している．それにより，専門的なセキュリティ知識に頼らずに発注者が開発プロセスに能動的に関わるようになることが期待できる．本稿では，提案手法を紹介するとともに，Web アプリケーション更新時のセキュリティ要件を獲得するためのアプローチに関する検討状況を示す．

1. はじめに

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 サイトの改ざんなどのようなインシデントも多く報告されている．[1] それにともない，Web サイトを構成する 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．[2]

また，ベータ版サービスとして開始される事例の多いソーシャル・ネットワーキング・サービスに代表されるように，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，継続的にサービスが提供されている．そのサイクルのなかで，特に独自に作り込まれた機能が修正される際に，セキュリティ対策が漏れ，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中でも，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，明確なセキュリティ要件が定義されていることは，ほとんど存在しない．

そこで，本稿では，システム機能ベースセキュリティパターンを拡張し，セキュリティ要件の獲得とともに受入試験時の試験方針を獲得出来るようにすることで，セキュリティ要件の定義と確認を一体化させ，システム機能ベースセキュリティパターンを開発現場で利用する際の有効性を高めることの検討を進め，その上で Web アプリケーション更新時に対象とすべきセキュリティ要件の獲得手法についての検討を行う．

† 情報セキュリティ大学院大学
Institute of Information Security

2. 背景と関連研究

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ける各開発プロセスにおける発注者と受注者の責任範囲の関係は図 1 のようになっている．

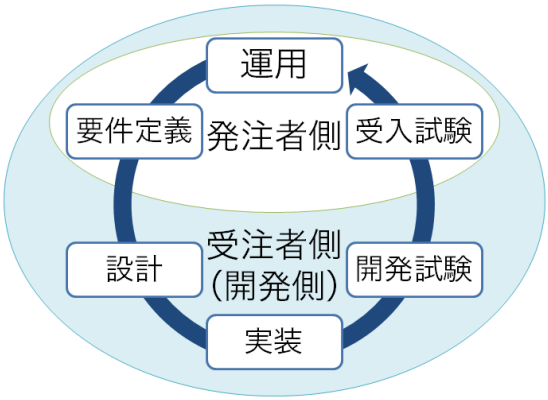


図 1 各開発プロセスにおける
発注者と受注者の責任範囲

発注者側は『実現したい機能を決める要件定義』[3]『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』[4]の各プロセスに責任を持つ．加えて，インターネット上に公開された Web サイトであれば，Web アプリケーションに脆弱性が存在すると，Web サイトの提供元である発注側の責任が問われることから，『実際に利用する運用』プロセスに対しても責任を持たなければならない．そして，受注者側は開発のプロフェッショナルとして，発注者が責任を持つ各プロセスに対しても，支援という形で関わることになる．しかし，開発現場の実態として，発注者は実現したい要求を受注者側に伝えるだけで，要件定義は支援の枠を越えて受注者側任せになっていることも少なくない．

しかしながら，不正アクセスによる Web サイトの改ざん

などのインシデントが身近な存在になっている現在，発注者側が Web サイトに対してセキュリティ面での安全性が確保されていることを確認する責任の重要性が増している．そこで重要になるのがセキュリティ対応の要件を決める『要件定義』と要件に従って実装されたかを発注者側が最終確認する『受入試験』の各開発プロセスである．これは[5]でも紹介されているようにシステム全体のコストに大きく影響することからも，発注者側が開発プロセスに関わることの重要性がわかる．

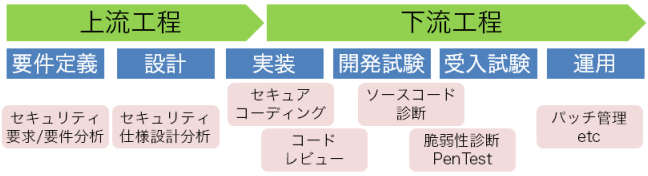


図 2　各開発プロセスにおけるセキュリティ対策

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[6]が著名であるが多いが，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を図 2 に示す．これらセキュリティ対策の内，実際に現場レベルで浸透しているものは，実装プロセスと運用プロセスに実施されるセキュアコーディングとコードレビュー，パッチ管理くらいである．次いで脆弱性診断やソースコード診断が行われており，すべて受注者側が主に責任を持つ下流工程の開発プロセスにおけるセキュリティ対策となっている．

上流工程に位置する要件定義で実施するセキュリティ対策として，セキュリティ要件分析があるが，実際に行われているプロジェクトを筆者の経験ではほとんど見たことがない．大久保らは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)は，‘‘開発に関係するステークホルダ（利害関係者）のセキュリティに対する意識や知識不足’’に起因するとしているが，不正アクセスが常態化している近年の状況では少数派になって来ていると想定される．2)3)は，‘‘開発体制の中でセキュリティの開発に必要な知識を持つ人材が不足していることが原因’’としているが，セキュリティ対策技術が通常の開発作業とは別の特別な技術として受け取られていることも，知識が浸透していかない原因だと考えられる．

要件定義プロセスを対象としたセキュリティ対策に関連する研究を調査した結果[8][9]を見ても，セキュリティ対策のために共通化されていない特別なアプローチの提案ばかりとなっており，実際の開発現場で利用可能な内容となっていないのが現状である．

大久保らによる研究では，セキュリティパターンを用いることで要求されるセキュリティ知識を最小化する手法が提案されているが，システム機能要件を定義する期間とは独立して非機能要件であるセキュリティ要件を導き出す期間が必要になり，最小化したとしてもセキュリティ知識を持つ人材不足への解とはなっていないと想定される．

また，セキュリティパターンを利用する手法については，吉岡らによって示されている研究動向[10]の中で以下の傾向が読み取れると述べている．

- 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

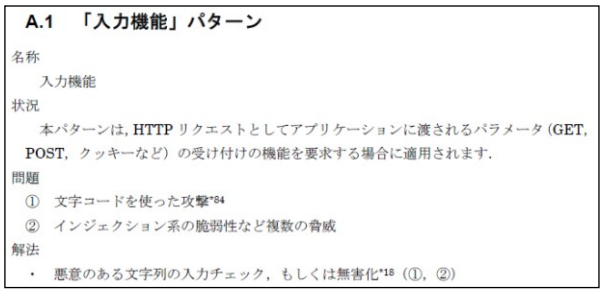
不足しているパターンを開発プロセス間の繋ぎ部分に注目して今後充実すべきとしており，発注者側よりも受注者側である下流工程を担当する開発者のために，設計工程で攻撃に関するパターンが必要であるとしている．しかし，発注者側が責任を持つ要件定義プロセスにおいては，言及されていない．他にもセキュリティパターンに関する提案はあるが，設計プロセスに着目したものが[11]をはじめとして，多く見られた．

吉岡らのセキュリティパターンを取り巻く研究状況や，大久保らのセキュリティパターンを利用することで必要となるセキュリティ知識を省力化できることに着目し，発注者側や受注者側にとって共通的な知識であるシステム機能をベースにしたセキュリティパターンを宇野が提案している．[12]

提案された手法では，図 3 に示すように，Web アプリケーションに求められる機能（例示では，「入力機能」）に着目し，

- その機能を利用する上で，発生・存在するセキュリティ上の問題は何か
- その問題を解決するために必要となる解法は何か

を事前にパターンとして準備して示すことで，専門的なセキュリティ知識が乏しい技術者であっても，セキュリティ要件を導き出すことが可能になることを目的としている．



A.1 「入力機能」パターン

名称

入力機能

状況

本パターンは，HTTP リクエストとしてアプリケーションに渡されるパラメータ（GET，POST，クッキーなど）の受け付けの機能を要求する場合に適用されます．

問題

① 文字コードを使った攻撃*84

② インジェクション系の脆弱性など複数の脅威

解法

・ 悪意のある文字列の入力チェック，もしくは無害化*18（①，②）

図 3 「入力機能」パターン

3. 検討手法について

既存研究では，新規にシステムを開発することを想定しており，ソーシャル・ネットワーキング・サービスをはじめとしたサービスを提供しながら，新たな要件を獲得し，Web アプリケーションの更新を繰り返すことが想定された研究はあまり存在しない．

そのため，短い開発期間でサービスを開始し，更新を繰り返す開発スタイルを意識した上で，セキュリティ対策に漏れが生じにくい手法が必要であると考える．

また，公開後に脆弱性が見つかり，炎上による風評被害など組織運営にまで影響を及ぼさないためにも，設計前に定義したセキュリティ対策が確実に実施されていることを確認するための手法も必要であると考える．

本研究では，多くの研究が受注者側である開発者を対象としているが，開発プロセスの上流工程において責任の多くを担う発注者を主な対象と定め，宇野が提案しているシステム機能ベースセキュリティ要求分析を拡張することで，要件定義プロセスにおいて，受入試験プロセスで必要となるセキュリティ対策が実施されていることを確認する方法を導き出す手法を提案する．サービスの運用開始後に脆弱性が見つかることによって生じる発注者側の損失を未然に防ぐために，受入試験でセキュリティ要件を満たした実装がされているかを確認することは，発注者側にとって有益なことであると考えられる．そのため，セキュリティに関して専門的な知識が必要であるために多くの開発現場で積極的に関わろうとしない発注者側が，能動的に関わるきっかけとして提案手法は活用可能であると考える．

また，個々のセキュリティ上の問題によって引き起こされる影響度は異なるため，前述の提案と同時に各パターンの『問題』に着目し，優先的に対応すべき問題は何かを分析した上で重み付けを行い，Web アプリケーション更新時のセキュリティ要件獲得の際に，受入試験で重点的に確認すべきセキュリティ要件を導き出す手法の検討を行う．

3.1 システム機能ベースセキュリティパターンの拡張

宇野の提案するセキュリティパターンの内，A.22「動的な表示機能」パターンを例に本研究における拡張方法について示す．

下記が宇野の提案するセキュリティパターンである．

‘‘A.22「動的な表示機能」パターン

■名称

動的な表示機能

■状況

本パターンは，外部からの入力に応じて，Web アプリケーション上で HTML や JavaScript を動的に生成する機能が要求され場合に適用される．

■問題

① クロスサイト・リプティング（XSS）

② エラーメッセージからの情報漏えい

■解法

・ XSS 対策（①）

・ 詳細なエラーメッセージの抑止（②）’’

このセキュリティパターンは「名称」「状況」「問題」「解法」の 4 つの部品から構成されており，ここに「試験方針」を追加することで，下記のように拡張することが可能である．

■試験方針

・ XSS 文字列に対する無害化処理の確認（①）

・ エラーが発生する文字列を入力した際に，エラーメッセージが出力されないことの確認（②）

なお，この「試験方針」に出てくる『XSS 文字列』や『エラーが発生する文字列』が実際に受入試験を実施する際に必要となってくるが，発注者側としては結果が確認出来れば良いことや，Web アプリケーションの環境に依存することからパターン化が困難なため，実際の開発現場においては，適宜 OWASP Testing Guide[13]に代表されるような文献を参照して試験が実施されることを期待し，本研究の対象としては扱わない．

システム機能ベースセキュリティパターンは，例示したように「問題」あるいは「解法」に対応する形で定義することで拡張が可能であり，個別の Web アプリケーションに特化しないように汎化することでパターンとして示すことが可能であると考える．

しかし，既存のセキュリティパターンに「試験方針」を拡張するだけでは，更新にともなう機能差分の影響範囲を把握することが出来ない．そのため，「問題」に対して重要度の設定を行うことで，既存のシステム機能に対するセキュリティ要件を導き出した場合と更新後のシステム機能に対するセキュリティ要件を導き出した場合とで，重要度による試験方針の見直しが可能になり，更新にともなう影響度や影響範囲の把握が可能にならないか検討を進めている．

現在検討中の「問題」に適用する重要度は大中小で表現し，以下の内容で影響度や影響範囲の把握が可能か確認中である．

・ 大：試験方針の見直しが必要
・ 中：既存試験方針を踏襲可能
・ 小：更新による影響なし

3.2 ケーススタディの検討

本研究手法の検証を行うために現在ケーススタディを作成中である．

ケーススタディで用いるシステムの要件としては，システム機能要件が具体的に導き出せ，かつ，システムの機能拡張によって更新が行える汎用的なシステムであることが挙げられる．そして，導き出されたシステム機能要件を基に従来手法と提案手法によってセキュリティ要件を獲得し，受入試験の項目設定や実施における作業量を分析し，比較することで評価を行うことを想定している．

ケーススタディに用いるシステムとして，システム機能要件が公開されている事例の多い図書館システムを考えている．図書館システムは，図書館内に閉じたシステムではなく，インターネット上に公開された Web アプリケーションである事例が増えており，他の図書館と連携することで，予約システムの更新や検索可能な図書館数の拡大に伴う更新など，Web アプリケーション更新時を想定する上で公開情報をもとに設計出来る．そして，広く横展開可能な汎用的なシステムとして設計可能であるため，ケーススタディで用いるには最適だと考える．

現在，具体的かつ現実的なシステム機能要件を基にシステム構成と更新時のシステム機能要件の検討を進めている段階であるため，評価結果などの詳細については記載出来ない．しかし，現段階でケーススタディを行う上で意識すべき課題として，インターネットバンキングやショッピングサイト，ソーシャル・ネットワーキング・サービスなどの図書館システム以外のシステムにおいても提案するセキュリティパターンが有効であるかを検証する必要があると認識している．

また，新規に Web アプリケーションを開発するよりも，要件獲得までのアプローチが複雑になることが想定されるため，パターン化に適さない事例が出てくる可能性があるが，その場合にどのようにパターンを活用可能かも検討が必要になるのではないかと考える．

4. おわりに

本稿では，Web アプリケーションの開発プロセスにおける発注者と受注者の責任分担や関係性を示した上で，発注者側が開発プロセスにおいてセキュリティ対策に関わる重要性を説明し，セキュリティ知識に乏しい発注者側の技術者が能動的に開発プロセスに関わる環境を構築するために利用可能なシステム機能ベースのセキュリティパターンを拡張した手法の提案を行った．この提案の中で，発注者側が主体的に関わるべきである受入試験プロセスを要件定義プロセスと結びつけることで，セキュリティ要件を基に受入試験の試験方針を獲得が出来るため，受入試験において，システム機能要件だけでなく，セキュリティ要件の充足度を確認することが可能になり，発注者に有益であることを提示した．

その上で，近年のインターネットへ公開後にシステムの更新を繰り返す Web アプリケーションを対象としたアプローチについて検討状況を示した．

今後は，セキュリティパターンの拡張と有効性の分析を行いながら，ケーススタディで用いるシステム構成を完成させ，システムの機能拡張によって得られるシステム機能要件のさらなる検討・精査を行う．また，ケーススタディを実施することで，既存研究に対する有効性について検証を行う．

参考文献

1) JPCERT/CC インシデント報告対応レポート [2014 年 1 月 1 日～2014 年 3 月 31 日]
https://www.jpcert.or.jp/pr/2014/IR_Report20140415.pdf
2) McGraw, Gary. "Software security." Security & Privacy, IEEE 2.2 (2004): 80-83.
3) 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
4) 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
5) J.Stecklein, Error cost escalation through the project life cycle, http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20100036670_2010039922.pdf, NASA, Tech. Rep., 2004.
6) 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
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) Tondel, Inger Anne, Martin Gilje Jaatun, and Per Håkon Meland. "Security requirements for the rest of us: A survey." Software, IEEE 25.1 (2008): 20-27.
9) Elahi, Golnaz, et al. "Security requirements engineering in the wild: A survey of common practices." Computer Software and Applications Conference (COMPSAC), 2011 IEEE 35th Annual. IEEE, 2011.
10) 吉岡信和, 鷲崎弘宜, and 丸山勝久. "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 (2007): 39-46.
11) Ferraz, Felipe Silva, Rodrigo Elia Assad, and S. R. Lemos Meira. "Relating security requirements and design patterns: Reducing security requirements implementation impacts with design patterns." Software Engineering Advances, 2009. ICSEA'09. Fourth International Conference on. IEEE, 2009.
12) 宇野健二 : 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 2011 年度修士論文
13) OWASP Testing Guide v3, http://www.owasp.org/images/5/56/OWASP_Testing_Guide_v3.pdf