4A-05

ソフトウェアセキュリティ知識ベースを活用した セキュアなソフトウェア開発事例ベースの提案

櫨山淳雄† 齊藤大仁† 吉岡信和‡ 小橋孝紀§ 鷲崎弘宜§ 海谷治彦* 大久保隆夫**
†東京学芸大学 ‡国立情報学研究所 §早稲田大学 *神奈川大学 **情報セキュリティ大学院大学

1. はじめに

インターネット上でのサービスの増加に伴い、セキュアなソフトウェア開発へのニーズが増大しており、ソフトウェアセキュリティ[7]に関する研究が活発に行われている。セキュアなソフトウェアを開発するためのプロセス(手法)、原則、パターン、ガイドライン等の様々な技術開発がなされてきた。

しかし、それらがソフトウェア開発ライフサイクル全体で体系的に整理されていないという問題認識から、セキュアなソフトウェア開発のための知識ベース（以下知識ベースと記す）の構築を進めてきた[6]。また、知識ベース中の知識を設計根拠としてセキュアなソフトウェア開発において作成される成果物と関連づける学習支援環境を提案した[6]。

さらに、ライフサイクル全体を扱ったセキュアなソフトウェア開発の事例が少ないという指摘がある[2]。そのような背景から大久保らによりソフトウェアセキュリティの共通問題を作成する 1 つの試みがなされている[10]。

本論文は、大久保らが作成したソフトウェアセキュリティの共通問題を事例として、その成果物と知識ベースとを関連付けた知識ベースを活用したセキュアなソフトウェア開発事例ベースを提案する。

2. 事例ベース管理システム

セキュアなソフトウェア開発で作成される成果物の管理、知識ベースの管理、成果物と知識ベースとの関連付けを行うためのシステムとして、事例ベース管理システムを開発した。

3. ソフトウェアセキュリティ知識ベース

図 1 にソフトウェアセキュリティ知識ベースの概念モデルを表している。このモデルは Barnum と McGraw によるソフトウェアセキュリティのための知識[3]を拡張したものである。知識ベースに登録されている知識は各クラスのインスタンスである。知識ベース構築にあたっては、Web 上で公開され入手可能な既存の成果を使用している。

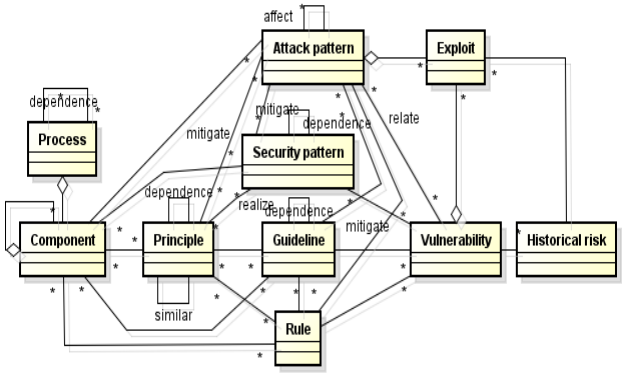

図 1.セキュリティ知識ベースのための概念モデル

- Process (プロセス) : ライフサイクル全体を扱っていることから CLASP (Comprehensive, Lightweight Application Security Process)[11]を、セキュリティ要求分析手法として大久保と田中の手法[9]を登録している。セキュリティ要求分析手法は CLASP プロセスの一部と位置付けている。
- Principle (原則) : CLASP の Principle[12]を登録している。
- Security pattern (セキュリティパターン) : Yoder と Barcalow のパターン[14]を登録している。
- Guideline (ガイドライン) : マイクロソフトの設計ガイドライン[8]、Mozilla のセキュアコーディングガイドライン[5]を登録している。
- Rule (ルール) : OWASP (The Open Web Application Security Project) Cheat Sheet[13]を登録している。
- Attack pattern (攻撃パターン) : CAPEC (Common Attack Pattern Enumeration and Classification)[4]を登録している。脅威分析においては脅威の分類である STRIDE を用いることがあるので、各攻撃を STRIDE と関連付けて登録している。

個々の知識の間に関連がある場合には、知識間の関連も登録している。

4. ソフトウェアセキュリティの共通問題

ソフトウェアセキュリティの共通問題は産学が共同で用いることのできるセキュアなソフトウェア開発手法の実験、評価用の標準的なデータの 1 つの試みである[10]。この共通問題は大学における教務システム (EMSSec と呼ばれる)であり、大学説明会参加者の管理、在学生の管理、卒業生に対する証明書発行管理の 3 つの主要機能から構成される。

この共通問題に対して、ユースケース図、クラス

Proposal of Case Bases of Secure Software Development using Software Security Knowledge Base, Atsuo Hazeyama, Masahito Saito, Nobukazu Yoshioka, Takanori Kobashi, Hironori Washizaki, Haruhiko Kaiya and Takao Okubo, Tokyo Gakugei University, National Institute of Informatics, Waseda University, Kanagawa University and Institute of Information Security.

図、脅威とその対策木、CakePHP [1]で記述されたソースコードが成果物として作成されている。これらはソフトウェアセキュリティの専門家により検討されたものである。

5. 事例ベース

5.1 事例ベース構築手順

セキュリティ要求分析に関しては、ユースケース図とクラス図 (識別されたアセット)を入力として、大久保と田中の手法と攻撃パターン CAPEC を知識として用い、ミスユースケース図を作成した。この時、共通問題の成果である攻撃とその対策木を参考にした。

実装に関しては、セキュリティ要求分析において関連付けた脅威に対する解決策 (ガイドライン、ルール等)とソースコードの関連付けを行った。

5.2 結果

表 1 に開発事例の成果物と知識ベースを関連付けた結果の抜粋を示す。共通問題の成果物である脅威とその対策木に記述されていた脅威はミスユースケース図に反映された。プログラミング言語として採用された CakePHP には Security コンポーネントや Auth コンポーネントなどセキュリティ対策機能が提供されている[1]。また、文字列のエスケープ処理を行うメソッドも用意されており、それらを使用したセキュリティ対策がなされていた。

表 1. 開発事例と知識ベースを関連付けた事例ベース (抜粋)

識別された脅威	知識ベース中の知識 (攻撃)	知識ベース中の知識 (対策)	共通問題の対策
XSS	CAPEC243	フィルタをかけていない入力を受け付けないようなライブラリやテンプレートを使用する (OWASP XSS Prevention Cheat Sheet)	分析：入力妥当性検査 実装：validation 機能で入力チェック
		入力の正規化、フィルタ、ホワイトリスト化 (OWASP XSS Prevention Cheat Sheet)	分析：出力のエスケープ 実装：CakePHP の h()関数を利用

5.3 議論

セキュアなソフトウェア開発の事例を提供している研究は少ない。その中で Apvrille と Pourzandi はインスタントメッセージサービスを対象に、セキュリティ要求分析から設計、実装、テストの各工程について成果物を示し、いくつかのトピックを解説している[2]。それぞれのトピックはセキュアなソフトウェア開発にとって重要な項目であるが、成果物とトピックを直接関連づけてはいない。

それに対して、本研究では、知識ベースにおいてセキュアなソフトウェア開発のために行うべき活動 (プロセス)と、それらの活動のための指針となる情報 (例えば脅威分析においては攻撃パターン、実装においてはガイドライン、ルール等)が関連付けて管理されており、何をどのように行うかということ、その具体的な成果物 (開発事例)が関連付けられており、セキュリティに詳しくないソフトウェア開発者に対するセキュアなソフトウェア開発への一助になると考えている。

6. おわりに

本稿では、セキュアなソフトウェア開発支援の一環として、ソフトウェアセキュリティの共通問題を事例として、そこで開発された成果物とソフトウェアセキュリティ知識ベースとを関連付けた事例ベースを構築した。

謝辞

本研究は平成 26 年度国立情報学研究所共同研究として行われた。記して謝意を表す。

参考文献

[1] 安藤祐介, 岸田健一郎, 新原雅司, 市川快, 渡辺一宏, 鈴木則夫, CakePHP2 実践入門, 技術評論社, 2012.
[2] Axelle Apvrille and Makan Pourzandi, Secure Software Development by Example, IEEE Security & Privacy, Vol.3, No.4, pp.10-17, 2005.
[3] Sean Barnum and Gary McGraw, Knowledge for Software Security, IEEE Security & Privacy, Vol.3, No.2, pp.74-78, 2005.
[4] CAPEC, http://capec.mitre.org.
[5] Michael Coates, Chris Lyon and Mark Goodwin, WebAppSec/Secure Coding Guidelines, https://wiki.mozilla.org/WebAppSec/Secure_Coding_Guidelines (Accessed 30 Dec. 2014).
[6] Atsuo Hazeyama and Masahito Saito, Preliminary Evaluation of a Software Security Learning Environment, International Journal of Software Innovation, Vol.2, No.3, pp.26-39, 2014.
[7] Gary McGraw, Software Security, IEEE Security & Privacy, Vol.2, No.2, pp.80-83, 2004.
[8] MSDN, セキュリティ保護された Web アプリケーションの設計ガイドライン, http://msdn.microsoft.com/ja-jp/library/ff648647 (d=printer).aspx (Accessed 30 Dec. 2014).
[9] 大久保隆夫, 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, Vol.50, No.10, pp.2484-2499, 2009.
[10] 大久保隆夫他, セキュリティとプライバシーを考慮したソフトウェア開発における共通問題の調査研究, 産学戦略的研究フォーラム, 2013.
[11] OWASP, CLASP Best Practice, https://www.owasp.org/index.php/Category:CLASP_Best_Practice (Accessed 30 Dec. 2014).
[12] OWASP, CLASP Security Principles, http://www.owasp.org/index.php/CLASP_Security_Principles (Accessed 30 Dec. 2014).
[13] OWASP Cheat Sheet Series, https://www.owasp.org/index.php/OWASP_Cheat_Sheet_Series (Accessed 30 Dec. 2014).
[14] Joseph Yoder and Jeffrey Barcalow, Architectural Patterns for Enabling Application Security, Proc. 4th Conference on Patterns Language of Programming (PLoP'97), 1997.