3B-3

Webアプリケーション開発のためのソフトウェアセキュリティ知識ベースの開発

櫨山 淳雄†
東京学芸大学

1. はじめに

インターネット上でのサービスは増大の一途を辿っている。それに伴い、コンピュータセキュリティの重要性が高まってきている。特に近年、多くのサービスがソフトウェアで実現され、その複雑さが増大しているため、ネットワークセキュリティ技術(アクセス制御や暗号技術など)のみでなくソフトウェアセキュリティ技術の重要性が認識されてきた[7]。ソフトウェアセキュリティはソフトウェア開発過程全体でセキュリティを扱い、セキュアなソフトウェアを開発することを目指す[7]。近年ソフトウェアセキュリティに関する標準、開発プロセス、パターン等が活発に開発されてきた。しかし、それらの間の関係性が整理されておらず、利用に困難を来している[6]。

そこで本論文では、これまでに提案されてきたソフトウェアセキュリティに関する知識を関連付けた知識ベースを提案する。

2. ソフトウェアセキュリティ知識ベースのための概念モデル

Barnum と McGraw はソフトウェアセキュリティのための知識を、7 つの知識カタログ(principle, guideline, rule, attack pattern, vulnerability, exploit, historical risk)とその関連としてモデル化している[1]。また櫨山は Barnum と McGraw のモデルを拡張した概念モデルを提案している[6]。図 1 に櫨山が提案した概念モデルを UML のクラス図として表している。

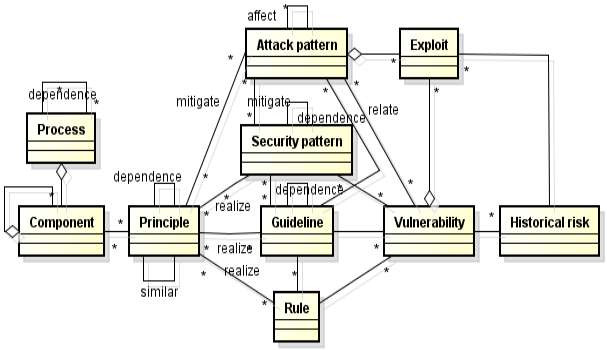

図 1.セキュリティ知識ベースのための概念モデル

†Development of Software Security Knowledge Base for Web Application Development, Atsuo HAZEYAMA, Tokyo Gakugei University.

3. ソフトウェアセキュリティ知識ベース

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することとする。

プロセスとして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)[9] を使用する。Principle として CLASP の Principle[10]、Saltzer と Schroeder の Principle [12]、NIST の Principle[13]を使用する。セキュリティパターンとして Yoder と Barcalow のパターン[15]を使用する。Guideline としてマイクロソフトの設計ガイドライン[8]と Mozilla のセキュアコーディングガイドライン[4]を使用する。Rule として OWASP (The Open Web Application Security Project) Cheat Sheet（例えば[11]）を使用する。Attack pattern として CAPEC (Common Attack Pattern Enumeration and Classification)[3]を使用する。Vulnerability として CWE[5]を使用する。

知識の関連付けには、CLASP プロセスと Principle の関連に着目した Buyens らの研究[2]、Principle とセキュリティパターンを関連付けた Wassermann と Cheng の研究[14]、Attack Pattern と Principle、Guideline、Vulnerability を関連づけた CAPEC[3]を参考にした。また、著者が各知識の記述を精査して関連付けを行った。

4. Webアプリケーションに対する攻撃

Web アプリケーションに対する主たる攻撃としてクロスサイトスクリプティング (Cross-Site Scripting。以下 XSS)と SQL インジェクション (SQL Injection。以下 SQLI)がある (例えば Top 10 Most Critical Web Application Attacks, http://www.websitedefender.com/web-security/owasp-top-10/参照)。

XSS は妥当性の確認されていないデータに基づいて HTML ページ中のリンクのようなコンテンツを生成する Web アプリケーションを悪用する攻撃である。SQLI は悪意のある SQL 文により、データベースに不正にアクセスする攻撃である。

5. ソフトウェアセキュリティ知識ベースの検証

本節では XSS と SQLI に対して知識ベースが対応可能かを検証する。これらの攻撃の詳細について CAPEC を参照する。

(1)XSS に対する検証

XSS に関して CAPEC によると、入力の妥当性確認を行うことや出力のエンコーディングを行う

ことが記述されている。

入力の妥当性確認について、CLASP プロセスでは Activity “Implement validation and error handling on function or method input”で言及されており、CLASP Principle でも“Input validation”が言及されている。また、MSDN の設計ガイドライン並びに Mozilla のセキュアコーディングガイドラインではともに“Input validation”として具体的に何を行えばよいかを提示している。MSDN の設計ガイドラインでは、「すべての入力を悪意あるものと仮定すること」、「アプローチの集中」、「クライアント側での検証に依存しないこと」、「正規化の問題に注意すること」、「入力を制限・拒否・校正すること」を挙げている。Mozilla のセキュアコーディングガイドラインでは、「受入れ可能な文字タイプを定義すること」、「(最小/最大)データ長を規定すること」などを挙げている。「アプローチの集中化」に関して Yoder と Barcalow は“Single Access Point”を提案している。出力のエンコーディングについては、CLASP プロセスでは Activity “Implement validation on function or method outputs”で言及されている。また、Mozilla セキュアコーディングガイドラインでは出力エンコーディングが XSS やインジェクション攻撃を防止する基本的な手段であると述べている。さらに、Mozilla セキュアコーディングガイドラインの出力エンコーディングの記述から OWASP の XSS Prevention Cheat Sheet にリンクが張られており、そこでは具体的なルール(例えば「HTML のタグを表す<>を&lt や&gt に置き換える」等)が記述されている。

(2)SQLI に対する検証

SQLI に関して CAPEC によると、「入力の妥当性確認を行うこと」や「エラーページのカスタム化を行うこと」が記述されている。

入力の妥当性確認は XSS と同じである。Mozilla セキュアコーディングガイドラインの出力エンコーディングの記述から OWASP の SQL Injection Prevention Cheat Sheet にリンクが張られており、そこでは具体的なルールが記述されている。エラー処理については、Mozilla セキュアコーディングガイドラインの“Error handling”に、エラー処理の推奨として、「表示するメッセージを一般的な内容にすること」、「ログファイルを記録すること」を挙げている。

6. おわりに

本稿ではソフトウェアセキュリティのための知識ベースをソフトウェアセキュリティの分野で開発されてきた知識を関連付ける形で構築した。そして Web アプリケーションの主たる攻撃である XSS と SQLI に対して知識ベースに蓄積されている知識を用いてプロセスから始まりコーディングの具体的な対応方法まで関連付けて把握できることを示した。今後は知識ベースを活用したセキュアなソフトウェア開発の実例を蓄積していきたい。

謝辞

本研究の一部は科学研究費補助金基盤研究 (C) 22500910 の助成のもとで行われた。

参考文献

[1] Sean Barnum and Gary McGraw, Knowledge for Software Security, IEEE Security & Privacy, pp.74-78, No.2, 2005.
[2] Koen Buyens, Riccardo Scandariato, and Wouter Joosen, Process Activities Supporting Security Principles, Proc. 31st Annual International Computer Software and Applications Conference (COMPSAC2007), Vol.2, 2007.
[3] CAPEC, http://capec.mitre.org.
[4] Michael Coates, Chris Lyon and Mark Goodwin, WebAppSec/Secure Coding Guidelines, https://wiki.mozilla.org/WebAppSec/Secure_Coding_Guidelines (Accessed 3 Jan. 2012).
[5] CWE, http:// http://cwe.mitre.org.
[6] Atsuo Hazeyama, Survey on Body of Knowledge Regarding Software Security, Proc. 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel & Distributed Computing (SNPD2012), pp.536-541, IEEE CPS, 2012.
[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 8 Oct. 2012).
[9] OWASP, CLASP Activities, https://www.owasp.org/index.php/Category:Activity (Accessed 3 Jan. 2012).
[10] OWASP, CLASP Security Principles, http://www.owasp.org/index.php/CLASP_Security_Principles (Accessed 3 Jan. 2012).
[11] OWASP XSS(Cross Site Scripting) Prevention Cheat Sheet, https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet (Accessed 1 Dec. 2012).
[12] Jerome H. Saltzer and Michael D. Schroeder, The Protection of Information in Computer Systems, Proc. the IEEE, 63(9), pp. 1278–1308, 1975.
[13] Gary Stoneburner, Clark Hayden, and Alexis Feringa, NIST Special Publication 800-27 Rev A, Engineering Principles for Information Technology Security (A Baseline for Achieving Security), Revision A, 2004, http://csrc.nist.gov/publications/nistpubs/800-27A/SP800-27-RevA.pdf (Accessed 3 Jan. 2012).
[14] Ronald Wassermann and Betty H.C. Cheng, Security Patterns (Accessed 3 Jan. 2012), http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.118.7818&rep=rep1&type=pdf.
[15] Joseph Yoder and Jeffrey Barcalow, Architectural Patterns for Enabling Application Security, Proc. 4th Conference on Patterns Language of Programming (PLoP’97), 1997.