

セキュアコーディングを学習させるためのオンラインコンパイラの開発

Development of Online Execution Environment to Learn Secure Coding

森田 浩平^{*1}, 松本 慎平^{*1}

Kohei MORITA^{*1}, Shimpei MATSUMOTO^{*1}

^{*1} 広島工業大学情報学部

^{*1}Faculty of Applied Information Science, Hiroshima Institute of Technology.

Email: {b214205, s.matsumoto.gk}@cc.it-hiroshima.ac.jp

あらまし: 本稿では、セキュアなコーディングを学習させるためのオンラインコンパイラの詳細を報告することを目的とする。著者らはこれまで、デバッグ機能、コーディング過程の自動記録機能、オンラインジャッジメント機能を備えたオンラインコンパイラを開発しており、提案システムは、これを拡張した仕組みである。提案システムでは、Web 開発で最も一般的な技術である PHP, JavaScript を対象とする。提案システムを用いることで、学習者は、Web ブラウザさえあれば、問題の提示やコーディング、自動採点を受けることができる。提案システムでは、学習者から提出されたコードに対して、静的解析ツールとテストツールを使用してコードの精査を行うことで、自動採点を実現させている。

キーワード: プログラミング, Web, 開発環境, セキュアコーディング, 学習支援

1. はじめに

現在日本国内では、10 秒に 1 人の被害者が出るほどの数のサイバー攻撃が行われている。シマンテック社は、2015 年に検出されたマルウェアが 4 億 3000 万を超えたと発表している。WEB に対する攻撃も、1 年で 2 倍以上に増加している⁽¹⁾と言われているが、既に数値では想像が出来ない規模のサイバー攻撃が日々行われているのが現状である。

サイバー攻撃が世界規模で猛火を振るう一方で、それに対応するセキュリティ人材不足が深刻化している。IT 専門家の 82%がセキュリティの人材不足を認識し、危機感を持っている。事実、国内でもセキュリティ人材が不足していることが IPA の試算により明らかにされている。セキュリティ人材不足の主な要因として、以下の 3 点があげられる。

- ✓ セキュリティ教育の困難さ
- ✓ セキュリティ人材の能力判定の困難さ
- ✓ セキュリティ知識の更新の早さ

セキュリティに関する知識は多岐にわたるが、業務形態や水準に応じて異なったセキュリティの知識が必要とされる。ソフトウェア開発に一番近いところでは、セキュアなアプリケーションを開発する知識、技能が重要となる。しかし、セキュアなコーディングを手軽に学習できる機会は少なく、また、高等教育機関においても十分に教育できていないのが現状である。そこで本稿では、セキュアなコーディングを学習させるためのオンラインコンパイラを提案すること、その詳細を報告することを目的とする。著者らはこれまで、デバッグ機能、コーディング過程の自動記録機能、オンラインジャッジメント機能を備えたオンラインコンパイラを開発しており⁽²⁾、提案システムである CureVuln は、これを拡張した仕組みである。提案システムでは、セキュリティに関する知識が強く求められる Web エンジニアリング

の中で広く用いられている PHP, JavaScript を対象とする。提案システムを用いることで、学習者は、Web ブラウザさえあれば、問題の提示やコーディング、自動採点まで受けることができるようになる。提案システムでは、学習者から提出されたコードに対して、静的解析ツールとテストツールを使用してコードの精査を行うことで、自動採点を実現させている。テストツールでは、攻撃コードを含んだ値を入力値として渡し、戻り値が適切にエスケープされているかをチェックする。テストを実行した結果、すべてのテストケースを passing できた場合は正解、できなかった場合は不正解として、フィードバックを学習者に提示することで、セキュアコーディングの学習環境を実現している。

2. セキュアコーディング学習システム

頑健なソフトウェアを開発するための技能は非常に重要である^(3,4)。ソフトウェアの頑健性を評価することを目的とした研究は、既にいくつか進められている。児島らは、Java アプレットが悪用される問題に対処するためのソースコード監査支援ツールを開発している⁽⁵⁾。鈴木らは、Self-Protecting 技術を用いて、DOM based XSS を防止するガイドラインに対する本方式の適用および有用性を検証した⁽⁶⁾。本研究と同様に、セキュアなアプリケーションを開発できる技能を養成することを目的とした取り組みもある。菱田らは、Java のセキュアコーディングを対象とし、インスタンスの状態を変更できないという特徴をもつ不変クラスのセキュアコーディングを利用した学習支援システム及びコードのチェッカを設計・実装した⁽⁷⁾。しかしながら、PHP, JavaScript のセキュアコーディングを学習させる仕組みについては、十分に行われていない。PHP, JavaScript は、Web システム開発で最も利用される言語であるものの、デザイナーなど専門家以外にも開発に携わることが多くあるた

XSS Escape

HTML特殊文字をエスケープして返す関数h(\$str)を作成してください

🔍 Hint

リスト 1 問題例

```

1 <?php
2
3 require "php_xss_problem_good.php";
4 assert_options(ASSERT_CALLBACK, 'assert_failure');
5
6 function assert_failure($file, $line, $code, $desc = null) {
7     $output = array(
8         'success' => 'fail',
9         'detail' => $desc,
10    );
11    echo json_encode($output, JSON_UNESCAPED_UNICODE);
12    exit(1);
13 }
14
15 function check() {
16     assert('h(">")' === ">", ">がエスケープされていません");
17     assert('h("<")' === "<", "<がエスケープされていません");
18     assert('h(""")' === """, ""が安全にエスケープされていません");
19     assert('h("\'")' === "'", "'が安全にエスケープされていません");
20     assert('h("&")' === "&", "&がエスケープされていません");
21 }
22
23 check();
24
25

```

リスト 2 リスト 1 の問題に対するテストの例

め、提案システムの有用性は高いと考えられる。

3. CureVuln の提案と実装

CureVuln は、PHP と JavaScript のセキュアコーディングを Web ブラウザのみで実際にコーディングを行いながら学習できるプラットフォームである。CureVuln は、Django REST Framework 3.3.3, NodeJS 4.2.6, Vuejs 1.0.24, Electron 1.3.8 を用いて開発されている。学習者は、提示された問題に対して、解答コードを記述し、実行ボタンを押下することで、そのコードをシステムに送信できる。送信されたコードは LXC コンテナへと転送され、問題に対応した攻撃コードを含む Unit テストを行う。例えば、リスト 1 で示した問題例に対し、リスト 2 の Unit テストが実行される。安全でない関数を使用していないかなどを確認するために、Lint を実行している。JavaScript の Lint には ESLint を採用し、主に DOM Based XSS を検知するために eslint-plugin-domxss を実装している。ESLint は、JavaScript コードを文字列として読み込み、AST にパース後、定義したルールを実行することができる。eslint-plugin-domxss では、DOM Based XSS を引き起こす可能性があるシンクやソースを定義したルールを記述することで、DOM Based XSS の検知を行っている。

自動採点機能については、Unit テストによる動的検査と Lint による静的解析の双方が passing すれば正解とし、できなければ対応したフィードバックを返すことで実現している。例えば、リスト 1 の問題にて「&」を「&」にエスケープできていない場合は、リスト 3 のようなフィードバックが返る。

CureVuln は、学習者が問題に対してどのような解答をしたかの過程を記録できるため、言語やフレームワークごとに、どのような脆弱性を作り込みやすいかを分析することが可能となる。

Fail: & は & にエスケープする必要があります

リスト 3 フィードバックの例

4. おわりに

本稿では、セキュアなコーディングを学習させるためのオンラインコンパイラ CureVuln を提案し、その詳細を報告した。デバッグ機能、コーディング過程の自動記録機能、オンラインジャッジメント機能を CureVuln に実装した。試験運用及び評価結果の詳細については、当日報告において明らかにする。なお、入力されたコードの詳細な評価方法については、今後検討すべき課題であると考えている。たとえば、プログラマにコーディング規約や規則に従わせることを目的とし、違反しているコードの構造を木パターンで表現し、ソースコード中からパターンにマッチする箇所を探すことで違反を検出するツールが提案されている⁽⁸⁾。こういった知見を踏まえて、いくつかの方法の有効性を比較・検証する予定である。

謝辞

本研究は、独立行政法人日本学術振興会科学研究費助成事業(基盤研究(C)16K01147, 17K01164)の助成を受けて実施した成果の一部である。ここに記して謝意を表します。

参考文献

- (1) シマンテック, インターネットセキュリティ脅威レポート 第 21 号 (2016).
- (2) K. Morita, S Matsumoto, Developing a Cloud-Based Programming Learning Support Tool - Aiming to the Most Accessible Development Environment for University Students -, Proc. of AROB 2017, GS6-2, pp.143-146 (2017).
- (3) 久保正樹, 脆弱性を作りこまない C/C++ セキュアコーディングプラクティス, 研究報告コンピュータセキュリティ, No.19, pp.1 (2009).
- (4) 村上正志, In Detail サイバー攻撃に強い制御システムを構築するには(Part6)攻撃側のスキルと高セキュアコーディングについて, 計装, Vol.56, No.5, pp. 74-83 (2013).
- (5) 児島尚, 中山裕子, 川島悟, 藤川亮子, セキュアな Java プログラム作成のためのソースコード監査支援ツール, 情報処理学会研究報告. マルチメディア通信と分散処理研究会報告, Vol. 2003, No. 18, pp.83-88 (2003).
- (6) 鈴木富明, 白井丈晴, 小林真也, 川端秀明, 西垣正勝, セキュリティガイドラインに準拠したアプリケーション作成支援に関する一提案, 研究報告コンピュータセキュリティ, Vol. 2015, No. 8, pp.1-8 (2015).
- (7) 菱田昂宏, 早川智一, 疋田輝雄, Java セキュアコーディングを促進する不変クラスチェッカの提案と実装, 第 76 回全国大会講演論文集, pp.537-539 (2014).
- (8) 中村真也, 鶴川始陽, 馬谷誠二, 則違反コードの構造を反映した木パターンを用いるコード検査器, 情報処理学会論文誌プログラミング, Vol.9, No.4, pp.1-15 (2016).