様式Ｃ－１９

# 科学研究費助成事業（科学研究費補助金）研究成果報告書

平成 24 年５月１４日現在

機関番号：３４３１５
研究種目：基盤研究(C)
研究期間：2009～2011
課題番号：21500043
研究課題名（和文）　ソフトウェア変更が安全性に与える影響の解明

研究課題名（英文）　Investigating the impact of software modification on its security characteristics

研究代表者
丸山　勝久（MARUYAMA KATSUHISA）
立命館大学・情報理工学部・教授
研究者番号：30330012

研究成果の概要（和文）：
本研究では、プログラム変更(ソースコードの変化)がソフトウェアのセキュリティ特性にどのような影響を与えるのかに着目し、脆弱性の混入への影響をプログラム内部のデータに関するアクセスの容易さの変化で判断する基準を提案した。さらに、この基準を用いることで、リファクタリングの適用時におけるソースコード変更が、そのコードの脆弱性に与える影響をプログラマに警告するリファクタリング支援ツールの開発に成功した。

研究成果の概要（英文）：
In this research study, we focused on the impact of program modifications (source code changes) on security characteristics of software, and proposed a criterion assessing the changes of accessibility of data stored in the target program. With this criterion, we achieved success in developing a novel tool that implements security-aware refactoring. It provides information so that programmers can easily know the impact of the applied refactoring on the security vulnerabilities of the modified code.

交付決定額

（金額単位：円）

| | 直接経費 | 間接経費 | 合　計 |
|---|---|---|---|
| 2009 年度 | 1,000,000 | 300,000 | 1,300,000 |
| 2010 年度 | 1,200,000 | 360,000 | 1,560,000 |
| 2011 年度 | 1,200,000 | 360,000 | 1,560,000 |
| 年度 | | | |
| 年度 | | | |
| 総　計 | 3,400,000 | 1,020,000 | 4,420,000 |

研究分野：　ソフトウェア工学
科研費の分科・細目：情報学・ソフトウェア
キーワード：　ソフトウェア保守、ソフトウェア変更、プログラム理解、ソフトウェアセキュリティ、リファクタリング、コード脆弱性

## １．研究開始当初の背景

ソフトウェアが社会において重要な役割を果たす現状において、ソフトウェアの脆弱性は深刻な問題である。特に、ネットワークを通してダウンロードされたコードをローカルホスト上で実行させたり、Web サービスのマッシュアップにおいて複数のサービスを連携させたりする状況では、ソフトウェアの安全性を保証する技術は必須である。たとえば、JavaScript を実行するコンテナ(Web ブラウザなど)に脆弱性が含まれると、遅延束縛(動的束縛)を悪用したメソッド呼び出しにより、攻撃者が悪意のあるコードを実行できる可能性が高くなる。また、Web サービスの

API 実装において、データを保持するインスタンスやその参照を外部に不用意に公開してしまうと、機密データの漏洩が発生し、また攻撃者に機密データの改ざんを許すことになる。

これらの問題に対して、セキュリティ要件定義やリスク管理をソフトウェア開発の初期段階から積極的に取り込むことで、脆弱性を排除しながら安全性の高いソフトウェアを開発する方法論の研究は活発である。しかしながら、従来の研究は安全性の高い新規ソフトウェアをどのように開発するのかに注力しており、既に稼働しているソフトウェアの安全性をどのように維持していくかという研究はほとんど行われていない。

現状では、稼働中のソフトウェアの安全性が維持されているかどうかを確認する手段として、セキュアプログラミングガイドラインに照らして保守者が手動で検査するか、あるいは、非常に限られた規則に基づき脆弱性を自動的に検出するツールを頻繁に適用するしかない。このため、開発当初は安全であったソフトウェアに対して、保守者がプログラム変更中に、無意識に不具合(セキュリティホール)を埋め込んでいる可能性があり、そのようなプログラムは攻撃に対して無防備である。

## ２．研究の目的

本研究では、ソフトウェア変更が、そのソフトウェアの安全性(脆弱性)の増減にどのような影響を与えるのかを、ソースコード解析技術に基づき明らかにする。ここで、本研究における安全性とは、自分で開発したコードと他人が開発した信頼できないコードが混ざりあって動作する環境において、内部データが勝手に他人に漏れない(機密性: confidentiality)、内部データが勝手に他人によって書き換えられない(完全性: integrity)、プログラムが勝手に停止しない(可用性: availability) という 3 つの特性を指す。これらの特性に違反するコードを脆弱なコードと定義し、ソフトウェア内部に存在する脆弱コードの数や規模(コード量)の変化で安全性の増減を判断する。

本研究では、次に示す 3 つの研究項目の実現を目指す。

（１）リファクタリングにおけるソースコード変換により、プログラムの安全性(脆弱性)がどのように増減するのかを評価する手法を確立する。

（２）リファクタリングをはじめとする数多くのソースコード変更を、安全性の増減という観点で、パターンとして分類・整理する手法を確立する。

（３）パターンを検査規則として扱い、プログラム変更後ではなく変更途中に、リアルタイムに脆弱性を警告するツールを構築する。

変更と安全性の増減とを対応付け、パターン化を行うことで、保守者は実際に変更を適用する前に、安全性の変化を容易に知ることができ、脆弱性が無意識に埋め込まれる可能性は大きく減少する。また、安全性の変化を常時評価・提示する仕組みをツールとして開発・保守環境に導入することで、保守者は、ソフトウェア変更の途中において、安全性に関するリスクを強制的に知ることになり、ソフトウェアの脆弱性が取り除かれる可能性は飛躍的に高くなる。

## ３．研究の方法

従来の研究が、変更後(あるいは変更前)のコードを検査することで、コード自体の安全性を評価していたのに対して、本研究では適用される変更(変更前後のコードの差分)から、安全性の増減を評価することが大きな特徴である(図１参照)。

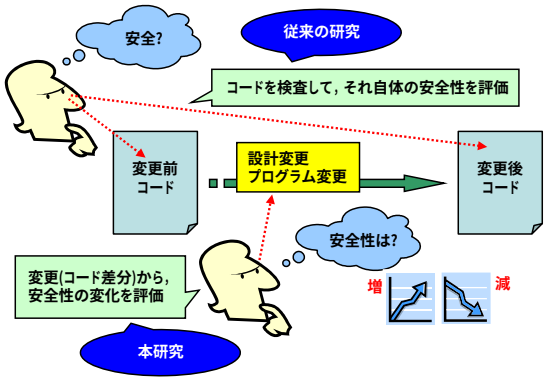


図 1: 本研究の特徴

コードの差分とセキュリティレベル(脆弱性の混入や除去)との関係に着目した研究は現時点では存在しない。このため、本研究では、ソフトウェア変更を広範囲に扱うことはせず、変更過程が比較的明確であるという理由から主にリファクタリングに着目し、安全性の影響の解明に取り組んだ。リファクタリングとは、ソフトウェアの理解性や拡張性の向上を目的として、既存ソフトウェアの外部から見た振る舞い(利用者から見た挙動)を変えずに内部構造を改良することを指す。これは、近年の(オブジェクト指向)開発におけるソフトウェア改善において必須の技術であり、多くの統合開発環境(たとえば、Eclipse)に標準で装備されている。

各研究項目に対する取り組みを次に示す。

（１）リファクタリングの実施手順が簡潔かつ明確にまとめられている Fowler のカタログを対象とし、リファクタリングの適用によりコード内部の脆弱性がどのように変化するのかを明らかにする。特に、セキュリティ特性における機密性と完全性に着目し、コー

ド内部のデータへのアクセスのしやすさの変化で、その安全性の増減を評価する基準を決定する。
（2）実際のソフトウェア変更履歴に含まれるコード差分を抽出し、どのようなソースコード変更が行われているのかを調査する。変更履歴の収集には、研究代表者の研究室で開発した Eclipse プラグインツールを活用する。一般的に、ソースコード変更を特定するためには、単純にコード差分を収集するだけでは不十分であり、収集したコード差分を適切に分類および抽象化する必要がある。本研究では、コード差分の分類および抽象化手法を確立することで、コード変更と安全性の増減の対応をパターンとして分類・整理することを試みる。
（3）研究項目（1）（2）の成果を受けて、ソースコード変更時の脆弱性の混入をリアルタイムに検出するツールを開発する。まずは、リファクタリング操作のみを対象とし、その脆弱性の増減を評価する基準を統合開発環境(Eclipse)上で実装したリファクタリング機能に組み込む。その後、ソースコード変更と安全性の増減に関するパターンを検査規則として扱うことで、リファクタリング以外のソースコード変更にも適用可能なツールを構築する。

## 4．研究成果

（1）ソースコード変更として、その手順が明確なリファクタリングを取り上げ、アクセス修飾子の変化と情報フローの概念を組み合わせた評価基準を提案した(図 2)。さらに、この手法に基づくリファクタリング支援ツール（Jsart: Java Security-Aware Refactoring Tool）の構築に取り組み、その実装に成功した（図 3)。このツールは、統合開発環境 Eclipse のプラグインとして動作し、2つのリファクタリング操作(Pull Up Method と Push Down Method)に対する脆弱性の混入を検出し、開発者に警告する。

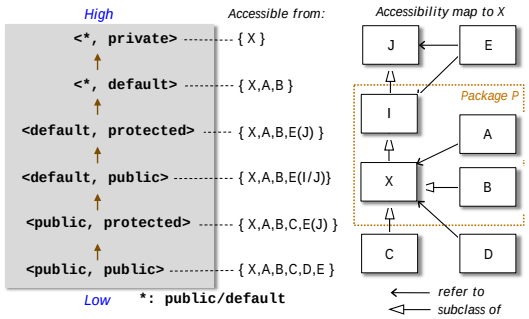


図 2: アクセス修飾子の変化に基づく脆弱性混入の評価基準

また、機密データの存在範囲(格納場所あるいは観測可能場所)やサブクラス化の有無による脆弱性混入を検出する評価基準の検討に取り組み、この基準において検出可能な脆弱性が混入する可能性を持つリファクタリングの洗い出しも実施した。

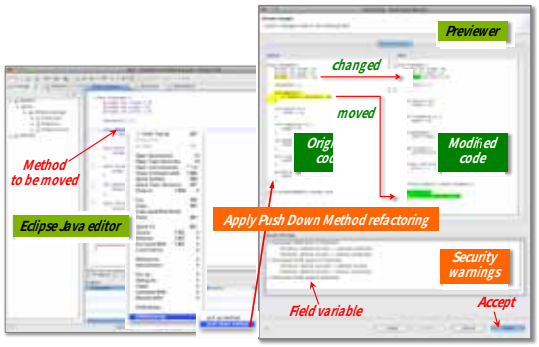


図 3: リファクタリング時に脆弱性の混入を検出するツール Jsart

（2）Jsart が検出可能な脆弱性の混入が実際のソフトウェア開発において発生する場面(パターン)を見つけ出し、実際のオープンソース・プロジェクトを用いて脆弱性混入検出の精度に関する評価実験を行った。これにより、ツールが有用であることがわかった。さらに、実験を通して、検出精度を向上させるためには、①プログラム解析の正確さの向上、②情報フローに関する十分な対応、③アクセスレベル基準の形式化に関する洗練を達成する必要があることが明らかになった。また、検出時間を短縮させるために、内部データの形式を再検討する必要があることもわかった。
（3）アクセス修飾子の変化と情報フローに基づく評価基準を、ソースコード中の機密性違反の検出に応用し、アスペクト指向プログラミング言語のポイントカットとして、機密性違反を宣言的に記述できる手法の提案に成功した。さらに、この手法を、AspectJ のコンパイラとして実現した。提案するポイントカットをプログラム記述に導入することで、ソースコード変更と脆弱性混入の検出を切り離してソフトウェアを設計できる可能性があることを立証した。
（4）ソフトウェア変更と脆弱性の混入との関係を明確にするという観点から、ソースコードの変更履歴を細粒度で記録するツール OperationRecorder の改良を行った。同時に、脆弱性の増減に関わる変更パターンの収集が迅速に実施できるように、記録した編集操作を再生(リプレイ)する OperationReplayer ツールの開発にも成功した（図 4)。このツールでは、過去の編集における特定の範囲を俯瞰的に表示するハイライト機能を提供する。このハイライトは、保守者により柔軟にカス

タマイズ可能となっている。このようなハイライトを用いることで、特定の条件を満たすソースコード変更だけを、膨大な履歴から探し出すことが容易になる。さらに、実際のソースコード変更を細粒度で記録した編集操作履歴を分析することで、コード差分から過去に行われたソースコード変更をより効率的かつ正確に検出する手法の検討を行った。この手法を実装したツール用いて簡単な実験を行った結果、細粒度な編集操作履歴を用いることにより、従来のコード差分に基づく分析に比べて、より正確なソースコード変更が特定できることが確認できた。

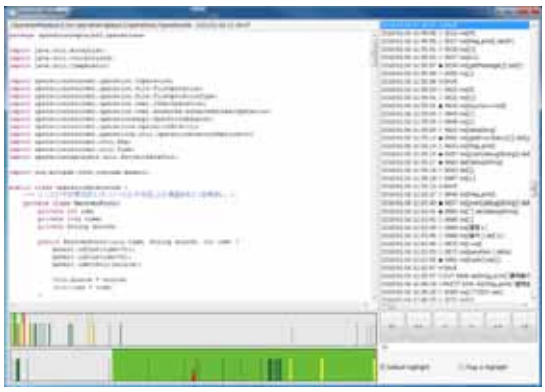

図 4: 編集操作再生ツール
OperationReplayer

５．主な発表論文等
（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計 5 件）

① 発表者名：木津栄二郎、大森隆行、丸山勝久、発表題目:ソースコード編集履歴を用いたプログラム変更の検出、雑誌名：コンピュータソフトウェア、査読：有、巻：29、発行年：2012、ページ：168-173
② 発表者名:鷲崎弘宜、坂本一憲、大杉直樹、権藤克彦、服部哲、久保淳人、小林隆志、大月美佳、丸山勝久、榊原彰、発表題目:デザインパターンへのソフトウェア工学的取り組み、雑誌名：コンピュータソフトウェア、査読：有、巻：29、発行年：2012 ページ：130-146
③ 発表者名：大森隆行、丸山勝久、発表題目:プログラム開発履歴調査のための編集操作再生器、雑誌名：コンピュータソフトウェア、査読：有、巻：28、発行年：2011、ページ：371-376
④ 発表者名：丸山勝久、戸子田健祐、大森隆行、発表題目:脆弱性に関する影響の可能性を警告するリファクタリング、雑誌名：情報処理学会論文誌、査読：有、巻：51、発行年：2010、ページ：1777-1793
⑤ 発表者名：丸山勝久、平井孝、 発表題目:プログラム依存グラフの等価性に基づくアスペクトの干渉検出、雑誌名：情報処理学会論文、査読：有、巻：50、発行年：2009、ページ：3108-3126

〔学会発表〕（計 16 件）

① 発表者名：轟大樹、大森隆行、丸山勝久、発表標題：CodeForest:ソフトウェア構造・特性・依存の視覚化によるプログラム理解支援、学会名等：電子情報通信学会技術研究報告、発表年月日：2012 年 3 月 14 日、発表場所：てんぶす那覇（沖縄県）
② 発表者名：丸山勝久、木津栄二郎、大森隆行、林晋平、発表標題：プログラム理解支援を目的とした編集操作スライスとその再生、学会名等：日本ソフトウェア科学会ソフトウェア工学の基礎 XVIII、発表年月日：2011 年 11 月 25 日、発表場所：海扇閣（青森県）
③ 発表者名：林晋平、大森隆行、善明晃由、丸山勝久、佐伯元司、発表標題：ソースコード編集履歴のリファクタリング手法、学会名等:日本ソフトウェア科学会ソフトウェア工学の基礎 XVIII、発表年月日：2011 年 11 月 25 日、発表場所：海扇閣（青森県）
④ 発表者名：木津栄二郎、大森隆行、丸山勝久、発表標題：ソースコード編集履歴を用いたプログラム変更の検出、学会名等：日本ソフトウェア科学会第 28 回大会発表年月日：2011 年 9 月 28 日、発表場所：沖縄産業支援センター（沖縄県）
⑤ 発表者名：Takayuki Omori、Katsuhisa Maruyama 、発表標題：An Editing-operation Replayer with Highlights Supporting Investigation of Program Modifications、学会名等：12th International Workshop on Principles on Software Evolution and 7th ERCIM Workshop on Software Evolution (IWPSE-EVOL'11)、発表年月日：2011 年 9 月 6 日、発表場所：Szeged (Hungary)
⑥ 発表者名：Katsuhisa Maruyama 、Takayuki Omori、発表標題：A Security-Aware Refactoring Tool for Java Programs、学会名等：4th Workshop on Refactoring Tools (WRT'11)、発表年月日：2011 年 5 月 22 日、発表場所：Hawaii (USA)
⑦ 発表者名：Takayuki Omori、Katsuhisa Maruyama、発表標題：A Software Development Environment Maintaining Fine-grained Code Metadata by Using Editing Operations、学会名等： IASTED International Conference on Software Engineering (SE'11)、発表年月日：2011 年 2 月 17 日、発表場所：Innsbruck (Austria)

⑧ 発表者名：Takayuki Omori、Katsuhisa Maruyama、発表標題：Flexibly Highlighting in Replaying Operation History、学会名等： International Workshop on Empirical Software Engineering in Practice (IWESEP'10)、発表年月日：2010 年 12 月 7 日、発表場所：NAIST (奈良県)
⑨ 発表者名：Ken-ichi Nakatani、Takayuki Omori、Katsuhisa Maruyama、発表標題：A Programming Environment Consisting of Web Services、学会名等：14th IASTED International Conference on Software Engineering and Applications (SEA'10)、発表年月日：2010 年 11 月 8 日、発表場所：Marina del Rey (USA)
⑩ 発表者名：大森隆行、丸山勝久、発表標題 プログラム変更履歴調査のための編集操作再生器、学会名等：日本ソフトウェア科学会ソフトウェア工学の基礎 XVII、発表年月日：2010 年 11 月 19 日、発表場所：越後のお宿 いなもと (新潟県)
⑪ 発表者名：Phan The Dai、Katsuhisa Maruyama、発表標題：Automatically Finding Web Documents Related to a Code Sample、学会名等：情報処理学会研究報告、発表年月日：2010 年 3 月 18 日、発表場所：国立情報学研究所（東京都）
⑫ 発表者名：Takayuki Omori、Katsuhisa Maruyama、発表標題：Identifying Stagnation Periods in Software Evolution by Replaying Editing Operations、学会名等：16th Asia-Pacific Software Engineering Conference (APSEC'09)、発表年月日：2009 年 12 月 3 日、発表場所：Penang (Malaysia)
⑬ 発表者名：伊三野直志、丸山勝久、発表標題：プログラム中の機密性違反を扱うセキュリティポイントカットとアドバイスの提案、学会名等：日本ソフトウェア科学会ソフトウェア工学の基礎 XVI、発表年月日：2009 年 11 月 21 日、発表場所：ホテルおかだ（神奈川県）
⑭ 発表者名：大森隆行、丸山勝久、発表標題 ソースコード編集履歴を用いた開発停滞期検出、学会名等：日本ソフトウェア科学会ソフトウェア工学の基礎 XVI、発表年月日：2009 年 11 月 19 日、発表場所：ホテルおかだ（神奈川県）
⑮ 発表者名：伏井洋平、大森隆行、丸山勝久、発表標題：プログラム変更支援のための再利用コンテキスト収集ツール、学会名等：情報処理学会研究報告、発表年月日：2009 年 11 月 5 日、発表場所：名古屋大学（愛知県）
⑯ 発表者名：Hironori Washizaki、Eduardo B. Fernandez、Katsuhisa Maruyama、Atsuto Kubo、Nobukazu Yoshioka、発表標題：Improving the Classification of Security Patterns、学会名等：3rd International Workshop on Secure Systems Methodologies Using Patterns (SPattern'09) 発表年月日：2009 年 9 月 2 日、発表場所：Linz (Austria)

〔図書〕（計 2 件）
① 著者名：丸山勝久、出版社名：CQ 出版、書名：組込みソフトウェア開発技術(第 3 章と第 5 章を担当)、発行年：2011、総ページ数：80
② 著者名：高橋直久、丸山勝久、出版社名：森北出版、書名：ソフトウェア工学、発行年：2010、総ページ数：192

６．研究組織
(1)研究代表者
丸山 勝久（MARUYAMA KATSUHISA）
立命館大学・情報理工学部・教授
研究者番号：30330012

(3)連携研究者
大森 隆行（OMORI TAKAYUKI）
立命館大学・情報理工学部・助教
研究者番号：90532903

桑原 寛明（KUWABARA HIROAKI）
立命館大学・情報理工学部・講師
研究者番号：30432222