氏　　名	宗藤 誠治
学位(専攻分野)	博士(情報学)
学 位 記 番 号	総研大甲第 1831 号
学位授与の日付	平成28年3月24日
学位授与の要件	複合科学研究科 情報学専攻 学位規則第6条第1項該当
学 位 論 文 題 目	アジャイル Web アプリケーション開発におけるソフトウェアセキュリティ保証のための反復型評価手法に関する研究
論 文 審 査 委 員	主　査　准教授 吉岡 信和 教授 胡 振江 教授 中島 震 准教授 鷲崎 弘宜 早稲田大学 准教授 石川 冬樹 国立情報学研究所

(別紙様式 2)
(Separate Form 2)

論文内容の要旨
Summary of thesis contents

ウォーターフォール型からアジャイル型へとソフトウェア開発プロセスが変化する中、ソフトウェアのセキュリティ保証の新しい仕組みが必要とされている。ウォーターフォール型と呼ばれる従来の開発プロセスでは、開発の各段階（要求、設計、実装、テスト）に対応したセキュリティ保証に、セキュリティ専門家を含む多くのリソースを費やすことで、脆弱性のないソフトウェアの実現を目指してきた。こうした取り組みは、特に大規模なソフトウェアやシステムの新規開発に広く実践されている。一方、アジャイル型と呼ばれるソフトウェア開発では、小規模な開発チームが短いリリースサイクルで反復的に開発を進める。そのため、脅威分析や網羅的なセキュリティテストのように手間と時間のかる作業を頻繁に実施することは難しい。また、セキュリティに関する専門家の不在、開発者が十分にセキュリティに関する知識を持っていない事も課題として挙げられる。

本研究では、反復的な開発プロセスにセキュリティ保証を効率的に組み込む方法として、アジャイルソフトウェア開発手法の一つであるテスト駆動開発を拡張し、開発者自身によるセキュリティ要求の把握とテストの記述と実行を実現する。その際の課題である、設計及びコード実装に依存する脆弱性への対応と、セキュリティテストで必要とされる網羅性の保証を実現するために、開発者の作業を、知識と作業効率の両方の観点から補佐する新しい手法を提案する。

提案手法では、アプリケーションの実装コードの中に現れるコマンドについて、そのセキュリティに関する情報を抽象化、集約、知識化する。具体的には、検査対象のアプリケーションを、アジャイル開発の中での実装部分（アプリケーションコード）と、フレームワークのようにアプリケーションが利用している部分 (フレームワークコード) の２つに分離し、後者の機能を抽象化した「コマンド抽象化ライブラリ」を作成する。ここで言うコマンドとは、アプリケーションの実装コードが呼び出すアプリケーションフレームワークの提供する様々な機能を示す。次に、コマンド抽象化ライブラリを用いて、実装コードの静的解析から、アプリケーションの振る舞いモデル生成、セキュリティ要求の抽出、(静的な)セキュリティ解析、(動的な)セキュリティテストカバレッジの計測をツール化することで、開発者のセキュリティ保証作業を補佐する。

ここで作成するライブラリにはフレームワークが提供するすべてのコマンドを登録する。その中でアプリケーションの振る舞いと、セキュリティに関係するコマンドについては、その特性を分類することでフレームワークの提供するセキュリティ機能を抽象化する。セキュリティに関係するコマンドを、Security Command, SC：セキュリティ機能を実現するコマンドと、Risky Command, RC：その使用がセキュリティ上のリスクとなる可能性のあるコマンドの 2 種類に分類する。SC に分類されるコマンドは、例えばアクセス制御に関するコマンドであり、これらは主にアプリケーションのセキュリティ要求及び設計（デザインの脆弱性）に関係する。RC に分類されるコマンドとしては、インジェクション攻撃の対象となるコマンドであり、主にコード実装(実装の脆弱性)に関係

する。以上のように、コマンドを 2 つに分類することで、提案手法は設計と実装の双方のセキュリティの問題に対応する。

設計に関する脆弱性に対応するためには、アプリケーションの動的な振る舞いを把握する必要がある。そのため、その振る舞いに関するコマンドもライブラリに登録する。この情報を元に、アプリケーションコードの静的検証からセキュリティ検証モデルを効率的に生成する。モデルは制御フローモデルとデータフローモデルから構成され、アプリケーションのセキュリティ機能の検証には制御フローモデルを、インジェクション攻撃などの攻撃可能性の検証にはデータフローモデルを利用する。開発者は、ツール化された本手法を用いて、実装したアプリケーションのセキュリティ機能、問題箇所を迅速に把握し、セキュリティテストケースを作成する。

セキュリティテストのカバレッジについては、コード中の SC と RC に対するテストの有無から、テストカバレッジを計測する。開発者は SC については、セキュリティ機能の確認(サンプリング)、RC については、その存在が脆弱性に繋がっていないことの確認のテストケースを作成する。これらは、カバレッジ情報を元に効率的にテストを配置する。

最後に、SC と RC にソフトウェアに関する体系的なセキュリティ知識を対応付ける。これにより、開発者がセキュリティ要求とその対策及びテストに関する知識に、実装コード視点から参照することが出来るようになり、セキュリティ知識が不十分な開発者をツールがサポートする。

以上のように、提案手法では、セキュリティの観点から開発者が柔軟かつ迅速に要求、実装、テストの関係を把握することで、アジャイルソフトウェア開発に適合したセキュリティ保証を実現する。本研究で提案する実装コードレベルでのセキュリティ知識のライブラリ化は、アプリケーションのセキュリティ保証の新しい手法であり、反復開発及び開発者間でのセキュリティ知識の共有と利用を、ライブラリを介して実現する。

提案手法の評価にあたり、アジャイルソフトウェア開発との整合性の確認の観点から次の 3 つの目標を設定した。1）要求や設計に起因するセキュリティの問題と、実装に起因するセキュリティの問題とを統一した手法で取り扱えること(ツール化)、2）コマンド抽象化ライブラリの作成と保守が開発者にとって大きな負担とならないこと、3）アプリケーション付随の回帰テストでセキュリティ保証を行える（テスト駆動開発にセキュリティテストが組み込める）ことである。提案手法を用いることによってこれらの目標が実現できることを、アジャイル型開発を代表する Web アプリケーションフレームワーク、Ruby on Rails 用のセキュリティツール（RailroadMap）の開発を通して確認した。

(別紙様式 3)
(Separate Form 3)

博士論文の審査結果の要旨
Summary of the results of the doctoral thesis screening

近年広く普及している Web アプリケーションの開発では，迅速かつ適応的にソフトウェアするアジャイルソフトウェア開発が広く用いられる。しかしながら，アジャイルソフトウェア開発では，主要な機能の開発を優先させるため，網羅的に実施し，保証する必要があるセキュリティテストとの整合性が悪い。そこで本論文では，セキュリティの保証を自動化するための手法を提案する。具体的には，プログラム中のセキュリティに関連したコマンドに対して,セキュリティのタイプや制御およびデータフローなどの情報を付加し，その情報をもとにセキュリティテストの網羅性を検証するためのセキュリティ検証モデルを構築する。その検証モデルとインターネットで公開されているセキュリティ情報知識(セキュリティデータベース）を用いることで，必要なセキュリティテストが実施されているか，どのようなテストが不足しているのかといったセキュリティテストの必要十分性を自動的に確認できるようにした。

1 章では，ソフトウェア開発におけるセキュリティの現状を述べるとともに，論文で扱うソフトウェアの種類とセキュリティ問題について整理している。そして，提案手法を提案する動機を示し，手法の要約を述べている。

2 章では，研究の技術的背景と関連研究について述べている。まず，ソフトウェアの開発プロセスとセキュリティ保証プロセスの関係を整理し，アジャイルソフトウェア開発に適したセキュリティ保証の取り組みについてその課題を整理している。そして，Web アプリケーション開発に関するセキュリティ技術および従来のセキュリティ保証手法について述べている。次に，モデル駆動開発とセキュリティ保証の取り組みと，今回実装対象とする Rails について述べ，最後に既存手法の課題について整理している。

3 章では，提案手法であるコマンド抽象化ライブラリを活用したセキュリティ保証の詳細について述べ
ている。具体的には，まず，提案手法の概要について述べ，提案の中心となるセキュリティに関するコマンドレベルの抽象化と，それを用いた検証のためのモデルの生成について説明している。そして，セキュリティテストのテストカバレッジの計測手法について，次に，コマンドレベル抽象化ライブラリを用いたセキュリティ知識との自動リンクについて説明している。最後に，提案手法のアジャイルソフトウェア開発への組み込みについてまとめている。

4 章では提案手法のツール化，Rails に対応したセキュリティテストツール，RailroadMap の機能及び実装の詳細と，RailroadMap を実際の Web アプリケーションに適用し,提案手法の有効性及び実効性について実験した結果を示している。具体的には，セキュリティ要求と実装の一貫性の確認と静的テストとの比較実験，セキュリティテストケースの生成についての実験，テストケースカバレッジの計測結果，セキュリティ知識との連携に関する実験結果について示し，最後に，既存のセキュリティテストツールとの機能比較を行っている。

5 章では本提案手法の有効性についてまとめるとともに，今後のセキュリティ保証のあり方について議論している。

最後の 6 章で本研究をまとめ，本論文を結んでいる。

出願者は,以上の研究成果を,査読付きジャーナル論文（International Journal of Secure Software Engineering）1 件，査読付き国際会議 1 本などにまとめている。 本論文の提

(別紙様式 3)
(Separate Form 3)
案内容は，新規性・有効性・信頼性において十分であると判断でき，学術的にも社会的にも価値があり，博士の学位論文として十分であるものと認められる。