From Code to Structure: Enhancing Web and Application Security Through Structure-based Code Analysis Using AI Techniques

Rozi, Muhammad Fakhrur

(Degree)
博士（学術）

(Date of Degree)
2024-03-25

(Date of Publication)
2026-03-25

(Resource Type)
doctoral thesis

(Report Number)
甲第8934号

(URL)
https://hdl.handle.net/20.500.14094/0100490159

(Attachment: Form 3)

Dissertation Abstract

Name: MUHAMMAD FAKHRUR ROZI

Department: Electrical and Electronic Engineering

Dissertation Title

From Code to Structure: Enhancing Web and Application Security Through Structure-based Code Analysis Using AI Techniques

(コードから構造へ : AI 技術を使用した構造ベースのコード分析によるウェブおよびアプリケーションのセキュリティ強化)

Academic Supervisor: Seiichi Ozawa

In the contemporary digital era, the significance of web and application security has been magnified by the surge in internet usage. This increase enhances connectivity and elevates the potential for cyber threats. Various system vulnerabilities present opportunities for cybercriminals to launch attacks such as injection attacks, drive-by-downloads, and cross-site scripting. These attacks are executed either by inserting harmful code into websites and applications or exploiting existing source code weaknesses. The examination of source code is crucial in countering these threats. It involves identifying harmful elements and vulnerabilities within the code. However, existing methods often need to fully assess the practical functionality of the code, which is vital for understanding its true purpose and potential threats.

This dissertation proposes an innovative approach to source code analysis, focusing on its structure. An accurate assessment of the code's intent and functionality can be achieved by examining how different parts of the code are organized and interact. Incorporating artificial intelligence (AI) in this process aids in a deeper analysis of the structural elements, providing insights into the underlying semantics of the code. The objective of this dissertation is mainly explained into three chapters, each describing a different aspect of this structured analysis, showing how this method can provide a better understanding of source code and improve security.

In Chapter 2 of the dissertation, a detailed study emphasizes the significance of structure-based features in enhancing the detection of malicious web pages. This study advances the understanding of web page content, specifically focusing on JavaScript, by extracting structural features from the source code. These features provide deeper insights into the logic of the web page, helping to determine whether it involves malicious or benign activities.

The study introduces a novel structure-based feature combined with other features previously identified in related research. This integration of new and existing features significantly improves the performance of machine learning systems in detecting malicious web pages, as demonstrated by the experimental results. The study identifies the most practical combination of web page features that achieve optimal accuracy in detecting malicious content, regardless of whether the web page contains JavaScript.

Furthermore, the study analyzes the influence of different feature categories on detection performance. The newly proposed structure-based feature is found to have the highest influence compared to other categories. This influence remains consistent across different cluster groups of web pages, with the proposed feature consistently ranking highest in impact.

However, the study acknowledges certain limitations. One essential limitation is that the proposed method focuses on individual web pages without considering their interrelationships, which could be crucial in identifying linked malicious activities. Future research is recommended to extend beyond single web pages, exploring connections between multiple pages to provide a more

comprehensive understanding of web-based threats. Additionally, the study suggests the need for real-world implementation to evaluate the actual performance of the proposed method. Implementing the method as a browser plugin could allow for real-time data collection, feature extraction, and model application, enhancing the practical utility of the research. The study recognizes the potential challenge of adversarial attacks, especially given the reliance on machine learning models. It highlights the need for ongoing research and development to ensure the robustness and effectiveness of the proposed detection method in the face of evolving cyber threats.

Chapter 3 of the dissertation focuses on detecting malicious JavaScript, introducing a novel structure-based analysis method utilizing Graph Neural Networks (GNN). This approach is designed to analyze the complete structural representation of source code through its graph representation. The primary goal is to enhance detection performance by applying a deep learning model to this structure-based analysis.

JavaScript code is converted into an Abstract Syntax Tree (AST), a structure commonly used in the compilation process of JavaScript engines. The GNN in this approach extracts structural features from the AST graph, considering the attributes of individual nodes. It is achieved through neural message passing and neighborhood aggregation, allowing the method to encode both the local structure and the node attributes of the AST graph. This dual focus helps in capturing the semantic meaning of the source code and identifying unique structures within the AST representations. The effectiveness of this method is demonstrated through extensive experiments on two different datasets, achieving high accuracy scores of 99.4% and 96.92%. These results highlight the method's capability in detecting various types of malicious JavaScript, with over 81% success in identifying different attack types and showing a significant performance improvement, particularly in detecting JS-Droppers, compared to sequence-based approaches.

The study also observes that the AST graph structure effectively represents the semantic meaning of the code, with distinct patterns and signatures that the proposed method can capture. However, there are limitations to this approach. One significant limitation is its sensitivity to flow-based obfuscation, meaning that its effectiveness depends on the techniques encountered during training. It could pose challenges in detecting novel obfuscation methods attackers use, including zero-day attacks. Another critical aspect is the need for reliable, accurately labeled datasets that realistically mimic actual attacks. Datasets that do not adequately represent real-world scenarios could negatively impact the performance of the detection model. These limitations present opportunities for further research and improvements in the field.

Chapter 4 of the dissertation delves into detecting vulnerabilities in source code, particularly focusing on the C/C++ programming languages widely used in software development. Recognizing the complexity and context-sensitive nature of many code vulnerabilities, the study proposes a more nuanced graph representation approach.

Traditional structure-based representations are often insufficient to capture the intricacies of source code vulnerabilities. These vulnerabilities often depend on specific contextual factors, such as variable names and data types, which are crucial for accurately identifying potential security issues. To address that, the study introduces an enhanced method of representing source code called ContextCPG (Contextual Code Property Graph). This novel representation incorporates detailed information about each node in the graph, including names and data types.

ContextCPG is designed as a multi-relational graph encompassing three types of edges. A Relational Graph Convolutional Network is employed to process this complex structure effectively. This network is adept at capturing the comprehensive information presented in ContextCPG, considering the distinct types of edges. Such an approach ensures that the learning process avoids overlapping, thereby maintaining the integrity of the final representation of each node.

The research demonstrates that combining structure-based analysis with natural language processing (NLP) techniques improves performance in detecting three common source code vulnerabilities: buffer overflow, invalid input, and use-after-free. Including contextual information enhances the CPG representation and can improve other types of graph representations.

However, the study acknowledges a significant limitation in this new representation: its inability to handle obfuscation effectively. This sensitivity is particularly pronounced when the source code contains unfamiliar words or characters, often occurring when developers use randomization or obfuscation techniques to protect their code from reverse engineering. While obfuscation may have less impact on purely structural representations, its influence becomes more evident when detailed contextual information is incorporated into the graph. This limitation highlights the need for further research to refine the ContextCPG approach and enhance its resilience against such obfuscation techniques.

In conclusion, this dissertation proposes a structure-based analysis approach to source code, offering significant advancements in the security of web and software applications. By focusing on the structural aspects of source code, this research contributes to a deeper understanding of potential vulnerabilities and malicious activities within web and application environments. The use of GNN for detecting malicious JavaScript, incorporating structure-based features for identifying malicious web pages, and the innovative ContextCPG model for vulnerability detection in C/C++ programming languages collectively demonstrate the efficacy of this approach. Each of these methodologies leverages the structural nuances of source code, enhanced by integrating AI techniques, to achieve more accurate and reliable detection of security threats.

This dissertation presents a new perspective in source code analysis and signifies a substantial contribution to AI in web and application security. It paves the way for developing more sophisticated, automated detection systems capable of responding to the evolving landscape of cyber threats. The insights and methodologies presented here could serve as a foundation for future

research to continually improve the robustness and effectiveness of security measures in the digital domain.

（別紙１）　　論文審査の結果の要旨

氏名	MUHAMMAD FAKHRUR ROZI		
論文題目	From Code to Structure: Enhancing Web and Application Security Through Structure-based Code Analysis Using AI Techniques （コードから構造へ：AI 技術を使用した構造ベースのコード分析によるウェブおよびアプリケーションのセキュリティ強化）		
審査委員	区　分	職　名	氏　　名
	主　査	教　授	小　澤　　誠　一
	副　査	教　授	寺　田　　努
	副　査	准教授	白　石　　善　明
	副　査		
	副　査		印

要　　旨

第１章は、ウェブやアプリケーション開発における急速な技術発展が様々な脆弱性を露呈し、深刻なサイバーリスクにつながっている現状を述べ、サイバー空間での安全性を担保する上で、機械学習が重要な役割を果たすことを述べている。

第２章では、悪意あるウェブサイト（悪性サイト）の検知精度を向上させる上で、URL やウェブコンテンツ（テキストや画像など）に加えて、JavaScript や PHP などのソースコードの構造に基づいた特徴量を加えることが有効であることを主張している。提案手法では、JavaScript から構造的特徴を抽出するため、まず、ソースコードを AST（Abstract Syntax Trees）表現に変換してから、graph2vec による埋め込みベクトルへの変換を行う。そして、URL の文字数や特徴的な文字列などに注目した Lexical 特徴量や WHOIS などのホスト情報、ウェブコンテンツ関連情報、ウェブ画像、ウェブコンテンツの埋め込みベクトル、URL 埋め込みベクトルなどを組み合わせて、高精度に攻撃者の意図を推定する方法を提案した。提案手法の悪性 JavaScript に対する検知性能を評価するため、悪性ウェブページ判定を行う大規模ベンチマークである GAWAIN（良性 61,080 件、悪性 44,405 件）を利用した。このデータセットにおいて、JavaScript が使われているページは 45,199 件であり、そのうち良性が 24,637 件、悪性が 20,482 件である。実験を行った結果、JavaScript を含まないウェブページに対しては 84.75%、JavaScript を含むページに対しては 84.84% という高い検知精度が達成された。

第３章では、悪意のある JavaScript の検知に焦点を当て、Graph Neural Networks（GNN）を利用した新しい構造ベースの解析手法を提案している。この手法はソースコードをグラフ表現に変換して、その構造を深層学習である GNN の表現学習で高精度な分析が可能となるように設計されている。分析対象の JavaScript コードは、まず前述した AST に変換され、そのノード属性も考慮して、構造的特徴を埋め込みベクトルとしてエンコードする。これはユーラル・メッセージ・パッシングと近傍集約によって実現され、AST グラフの局所構造とノード属性の両方をエンコードするのが特徴であり、ソースコードのセマンティクス獲得と AST グラフ内のユニーク構造の抽出に役立つ。提案手法の有効性を悪性と良性の JavaScript で構成される２つの異なるデータセットを使って性能評価を行い、それぞれ 99.40%と 96.92% という高い精度が達成された。さらに、81%以上の成功率で攻撃タイプを識別し、特に JS-Dropper の検出においては、従来手法であるシーケンスベースのアプローチに対して顕著な性能向上が見られた。

氏名	MUHAMMAD FAKHRUR ROZI

第 4 章では、ソースコードに含まれる脆弱性を検知するための機械学習手法を提案している。特に、ソフトウェア開発で広く使われている C/C++プログラミング言語に焦点を当て、ソースコードが含む脆弱性の複雑さと文脈依存な特性を認識して、ソースコードが含む微妙な文脈の違いを表現できるグラフ構造を使ったアプローチを提案している。ソースコードの脆弱性は、変数名やデータ型のような特定の文脈的要因に依存することが多く、従来の構造ベース表現では、その脆弱性の特徴を捉えることが難しかった。これに対し、本章では、CPG（Contextual Code Property Graph）と呼ばれるソースコードの構造表現力を

強化したアプローチと、CPG にグラフノードに変数名やデータ型などの詳細情報をグラフ畳み込みネットワークでエンコードした ContextCPG を採用している。ContextCPG はソースコードの包括的な情報を捉えることに長けており、本章では、構造ベースの解析と自然言語処理技術を組み合わせることで、バッファオーバーフロー、不正入力、use-after-free という 3 つの一般的なソースコードの脆弱性を検出する手法を提案している。提案手法の性能を、共通脆弱性タイプと呼ばれる Common Weakness Enumeration (CWE)のうち、特に CWE-119, CWE-20, CWE-672 の脆弱性を含む評価データセット（合計 5,578 件）で評価している。ContextCPG を特徴量として Relational Graph Convolution Network（RGCN）で学習したモデルは 0.75～0.81 の F1-Score を実現し、CPG を使ったときよりも検知性能が顕著に向上することが示されている。

第 5 章では、ウェブサービスやソフトウェア・アプリケーションが含む脆弱性を利用した攻撃に対して、3 つの機械学習を用いたソシューションを提案し、実際の攻撃データを使った性能評価の結果をまとめている。これらの結果より、ソースコードの構造ベース特徴量を生成する高度な深層学習モデルの導入により、従来の機械学習手法を上回る性能が達成可能であると結論している。また、提案手法の限界や今後の課題についても言及している。

本論文では、ソースコード解析における新たな視点を提供し、ウェブサービスやソフトウェア・アプリケーションのセキュリティに対して、AI、特に深層学習モデルが実用レベルで有効であることを実証した。データ駆動型である AI モデルはサイバー脅威の進化に対応可能であり、より洗練された自動検出システムに道を開くものである。本論文で提案された手法やその考察は、サイバーセキュリティ対策の堅牢性と有効性の継続的向上に大きく寄与すると考えられ、機械学習やサイバーセキュリティ分野における本論文の貢献度は大きい。提出された論文は工学研究科学位論文評価基準を満たしており、学位申請者の MUHAMMAD FAKHRUR ROZI は、博士（学術）の学位を得る資格があると認める.