# A Study on Dynamic Detection of Web Application Vulnerabilities

August 2011

Yuji Kosuga

# 主　論　文　要　旨

| 報告番号 | ㊙ 乙 第　　号 | 氏　名 | 小菅　祐史 |
|---|---|---|---|

主 論 文 題 目 :
A Study on Dynamic Detection of Web Application Vulnerabilities
（動的解析による Web アプリケーションの脆弱性検出手法に関する研究）

（内容の要旨）

Web 関連技術の発達により，Web アプリケーションは動的にコンテンツを生成することによってショッピングサイトやソーシャルネットワークサイトなど，高機能なサービスを提供している．その一方，Web アプリケーションは大規模化し，その構造も複雑になり，脆弱性が混入しやすくなっている．WhiteHat Security によると，83%の Web アプリケーションに少なくとも 1 つの脆弱性が存在する．Web アプリケーションに存在する脆弱性を検出する手段として，動的解析技術を用いた脆弱性検出手法が近年多く利用されている．この手法では，動作している Web アプリケーションに対して実際に攻撃を送信し，このとき Web アプリケーションが生成する HTTP レスポンスなどの出力を解析することにより，攻撃が成功したか判断することで脆弱性の有無を判定する．攻撃は無害な HTTP リクエストに悪意ある文字列（攻撃コード）を挿入することによって生成する．動的解析を用いた既存の脆弱性検出手法では，攻撃コードはあらかじめ定義されており，どの Web アプリケーションに対しても同じ攻撃コードを用いて攻撃を生成する．そのため，Web アプリケーションによっては必要な攻撃が不足している場合や無駄な攻撃を多く実行してしまう場合があり，検出精度が低いことが問題となっている．

本論文では，Web アプリケーションごとに攻撃コードを自動生成することによって，必要な攻撃のみ実行する手法を提案する．攻撃を成功させるために有効な攻撃コードは，HTTP レスポンスや SQL クエリなどに攻撃コードが現れる箇所の構文によって異なる．そのため，この出力の構文解析を行うことによって攻撃コードを埋め込む箇所の構文を調べ，その構文に合わせて攻撃コードを自動生成する．この仕組みにより，有効な攻撃を生成することができるだけでなく，構文に合わない攻撃コードを使用しないため，無駄な攻撃の実行を抑制することができる．攻撃コードは，本手法で用意した攻撃コード生成規則を参照することによって生成する．攻撃コード生成規則は，対象とする攻撃ごとに用意しており，構文の種類に応じて，有効な攻撃を生成するために使用する文字列の組み合わせ方法を定義している．本提案機構は，SQL インジェクションとクロスサイト・スクリプティング（XSS）に対する脆弱性検出手法を実現している．これらの手法は既存の評価の高い脆弱性検出ツールと検出能力の比較を行うことで，本提案手法の有効性を示した．また，実際に使用されている Web アプリケーションやオープンソースの Web アプリケーションに対して本提案機構を利用することで，これまでに 131 件の脆弱性を検出を行った．

また，次々と出現する攻撃に対して脆弱性検出手法を容易に実装することができるように，脆弱性検出用プラグインにより機能拡張が可能な脆弱性検出用フレームワークである Amberate を提案する．Amberate は動的解析を用いた脆弱性検出手法に共通する機能を隠蔽し，各攻撃に対する脆弱性検出手法に依存する処理を独自に実装可能な API（Application Programming Interface）を提供している．Amberate と同じく機能拡張が可能な従来の脆弱性検出ツールに対し，Amberate の XSS に対する脆弱性検出用プラグインと同様の機能を実装したところ，Amberate の方が 500 行少ないコード数（比較対象のツールの 82%の実装量）でプラグインを実装することができた．

# SUMMARY OF Ph.D. DISSERTATION

| School<br>Science for Open and Environmental Systems | Student Identification Number | SURNAME, First name<br>KOSUGA, Yuji |
|---|---|---|

Title

A Study on Dynamic Detection of Web Application Vulnerabilities


Abstract

With the evolution of web technologies, web applications have come to provide a wide range of web services, such as online stores, e-commerce, social network services, etc. The internal mechanism of web applications has gotten complicated as the web technologies evolve, which has also led web applications to the increase in the potential to contain vulnerabilities. WhiteHat Security reported that 83 percent of web applications they have audited during 2010 had at least one vulnerability. Vulnerability scanners that perform dynamic analysis are often used for detecting vulnerabilities in a web application. The dynamic analysis sends attacks to the web application to check to see if the output from the web application contains the implication of success of the attacks. An attack is an HTTP request that contains a maliciously crafted string. The existing vulnerability scanners that perform dynamic analysis define several malicious strings, and generate attacks using the same malicious string against different web applications. As a result, they tend to have a precision problem because of the absence of malicious strings necessary to exploit the web applications and the execution of useless attacks that can never be successful.

In this dissertation, we present our technique that performs efficient and precise vulnerability detection by dynamically generating effective attacks through investigating the output issued by the web application, such as an HTTP response or SQL query. By analyzing the syntax of the point into which attack is injected, our technique is able to generate only effective attacks as well as to prevent making useless attacks. An attack is generated by referencing to the attack rule that we prepared for each syntax of the point into which malicious string is injected. With this approach, we implemented a prototype Sania for discovering SQL injection vulnerabilities, and Detoxss for Cross-site Scripting (XSS) vulnerabilities. We demonstrate that these techniques find more vulnerabilities and performed more efficient testing than existing popular vulnerability scanners do. In our empirical study, we discovered 131 vulnerabilities in total in web applications currently in service and open source web applications so far.

Additionally, we present Amberate a framework for web application vulnerability scanners, which supports the plugin system to facilitate a new vulnerability detection technique. Amberate encapsulates functions commonly used for vulnerability detection techniques that perform dynamic analysis, and provides Application Programming Interface (API) for implementing functions different by each vulnerability detection technique. We demonstrated the ease of extending a new vulnerability detection technique by comparing the actual lines of code of the Amberate plugin for an XSS vulnerability detection with a plugin we implemented the same functions as Amberate XSS vulnerability detection plugin on an extensible vulnerability scanner. This result revealed that Amberate plugin required 500 fewer lines of code, which accounts for 82 percent of lines of code of the plugin for the other scanner.