L-032

ウェブサイト脆弱性診断対象ページ検出システムの提案

Proposal of website vulnerability diagnostic target page detection system

王　森宇†　　嶋　久登†
Senyu Wang　　Hisato Shima

1. はじめに

現在、ウェブサイトの安全性を確認するために、脆弱性診断が広く行われている[1]。その際、脆弱性診断の作業量を把握するため、一度簡易診断を行うことが一般的である。簡易診断では、ウェブサイトの動的ページ数や各ページにどんな脆弱性の可能性があるかなどの情報を取得する。しかし、手作業で行う簡易診断には、以下の問題がある。Ⅰ.時間がかかる。Ⅱ.情報を見落とす可能性がある。

そこで、本研究では、簡易診断を自動化し、ウェブサイト脆弱性診断対象ページ検出システムを提案する。本システムはページの HTML コンテンツを解析することで診断対象ページを検出するため、対象サイトに影響を与えることがない。また、サイト内のページを自動的に網羅することにより、脆弱性診断対象となるページを見逃すことなく検出することが可能である。

2. システムの目標

ウェブサイトに対する脆弱性診断の作業量を把握するために、本番診断の前に行う簡易診断を自動化することを目標とする。

本提案では、第一段階として各ページにどんな脆弱性の可能性があるかという分析をする時に、独立行政法人情報処理推進機構 IPA の「ソフトウェア等の脆弱性関連情報に関する届出状況 2014 年第 4 四半期（10 月～12 月）」[4]を参考し、一番影響が大きい SQLInjection と一番数が多い CorssSiteScripting を対象とした。

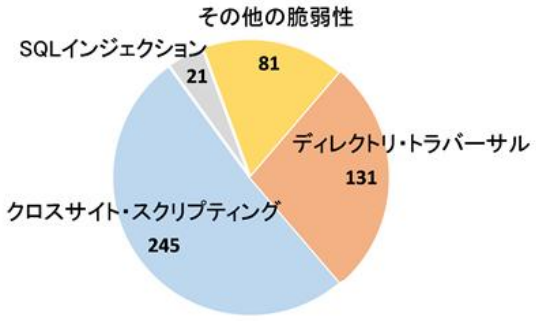

図1　脆弱性の件数の割合[4]

本システムはの対象となるウェブサイトのトップページの URL のみで自動的に分析を行い、以下の情報を洗い出す。

① サイト全体のページ数
② サイト内の全てのページの URL とタイトル
③ 各ページに対する SQLInjection と CorssSiteScripting の可能性の有無

3. システムの構成

図 2 に示すように、本システムはクローラー、各ページの分析スクリプト、分析結果メモリ、URL メモリの 4 つの部分で構成する。以下で各部分の動作を説明する。

① クローラーが URL メモリから URL を一つ取り出す
② クローラーがウェブサイトにリクエストを送る
③ クローラーがウェブサイトから HTML コンテンツを取得する
④ 取得した HTML コンテンツを分析し、その中の URL を URL メモリに書き込む
⑤ 取得した当ページの URL と HTML コンテンツのデータを分析スクリプトに渡す
⑥ 分析スプリクトが HTML コンテツを分析し、入力フォームに適切な値を入力しクローラーを経由しウェブサイトにリクエストを送る
⑦ 分析スクリプトがそのページに含まれる可能性がある脆弱性を分析し、結果を分析メモリに書き込む

このような動作を最後の URL を分析するまでに繰り返し、分析結果を基にレポートを作る。

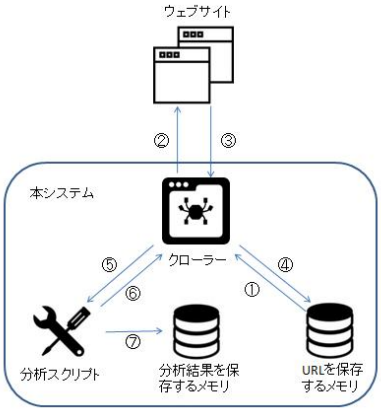

図２　システムの各部分の関係

3.1 クローラー

クローラーを利用し、サイト内の各ページ内の全ての URL を探し出す。本システムのクローラーは木構造の深さ優先探索の前順アルゴリズムを使用する。サイトのトップページから最後のページまで探索しながら、分析スクリプトで各ページを分析し、取得した情報をメモリで保存する。図３では、A をサイトのトップページとし、クローラーは A→B→C→D→E→F→G→H→I という順番で探索する。

† 神戸情報大学院大学 Kobe Institute of Computing; Graduate School of Information Technology

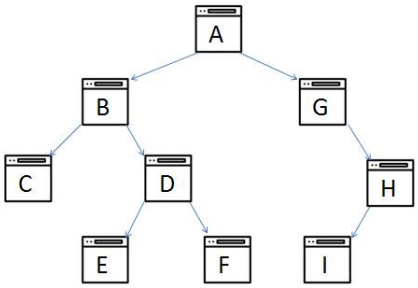

図３　ウェブサイトの構成例

3.2 各ページの分析スクリプト

検査スクリプトはクローラーから取得した対象ページの情報を分析する。その結果によって、分析されたページが脆弱性の可能性があるかどうかを判断する。そして最後の結果をメモリに書き込み保存する。

3.2.1 SQL Injection に対する分析

SQL Injection とはサイトが想定しない SQL 文を実行させ、データベースシステムを不正に操作する攻撃方法のことである[2]。従って、データベースと連携するページならば SQL Injection 脆弱性の可能性があると考えられる。

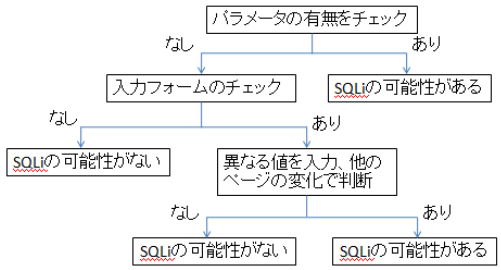

図４　SQLInjection に対する分析の流れ

データベースの連携は以下の手順で判断する。

① ページの URL にパラメータが付いている場合、データベースを利用する可能性があり、SQL Injection 脆弱性の診断対象ページとなる。

② ページの URL にパラメータが付いていない場合、ページの中の入力フォームの有無で判断する。入力フォームがない場合、本ページは SQL Injection 脆弱性がないと考える。

③ 入力フォームがある場合、入力フォームに数字或いはアルファベットで複数の値を入力し、それぞれに次のページの変化で SQL Injection 脆弱性の可能性があるかどうかを判断する。次のページの内容が変化した場合、可能性があると考える。内容が変化していない場合、SQL Injection 脆弱性がないと考える。

3.2.2 CrossSiteScripting（XSS）に対する分析

Cross Site Scripting（以下は XSS で示す）とはユーザーからの入力をそのままエコーバックすることを利用し、ページに悪意のあるスクリプトを注入する攻撃のことである[2]。つまり、前のページに入力した内容が他のページに表示されるならば、前のページは XSS 脆弱性の可能性があると考えられる。なお、本システムは URL のパラメータと入力フォームが悪意のあるスクリプトの入り口となる二パターンの SQLInjection に対する判断を行う。

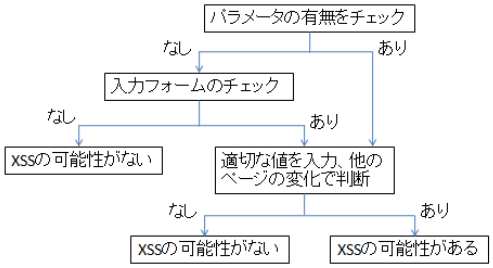

図５　CrossSiteScripting に対する分析の流れ

①ページの URL にパラメータが付いている場合、各パラメータに数字或いはアルファベットで複数の値を入力し、他のページに同じ値が表示されたら XSS 脆弱性の可能性があると考えられる。

②ページの URL にパラメータが付いていない場合、ページの中に入力フォームの有無で分析を行う。入力フォームがない場合、XSS 脆弱性の可能性がないと考えられる。

③入力フォームがある場合、①と同じように適切な値を入力し、他のページの変化で判断する。他のページに同じ値が表示されたら XSS 脆弱性の可能性があると考えられる。表示されない場合、XSS 脆弱性の可能性がないと考えられる。

4. おわりに

本研究はウェブサイトに対する脆弱性診断の作業量を把握するための簡易診断を自動化するシステムを提案した。今後、本システムを実装し手作業で行う簡易診断と比較する。しかし、判断の精密度や分析できる脆弱性パターンの数が少ないなどの問題がある。より多く脆弱性パターンに対する検査と検査の精密度の上昇させることが今後の課題である。

参考文献

[1]大森 雅司,亀山 友彦,関口 竜也:”「ウェブサイトにおける脆弱性検査手法の紹介」に関するレポート”. 独立行政法人情報処理推進機構 IPA . 入手先 < http://www.ipa.go.jp/files/000036973.pdf>(参照 2015-06-15).

[2]谷口 隼祐, 扇沢 健也,山下 勇太,徳丸 浩,高木 浩光:”安全のウェブサイトの作り方”. 独立行政法人情報処理推進機構 IPA.入手先< https://www.ipa.go.jp/files/000017316.pdf >(参照 2015-06-20).

[3]久山　真宏 . “WEB アプリケーションにおけるセキュリティ診断の検討”,日本教育情報学会,第 29 回年会,2013.

[4]独立行政法人情報処理推進機構.” ソフトウェア等の脆弱性関連情報に関する届出状況 ”. 入手先 < http://www.ipa.go.jp/security/vuln/report/vuln2014q4.html >(参照 2015-06-20).