改ざんサイト自動検知システムDICE(Detection of Injected Site using Cyber search Engine)の開発

田中 達哉† 田村 佑輔†† 甲斐 俊文＊ 佐々木 良一†

†東京電機大学
〒101-8457 東京都千代田区神田錦町 2-2
tanaka@isl.im.dendai.ac.jp

††セイコーエプソン株式会社 ＊パナソニック電工株式会社

あらまし 近年, SQLインジェクションを用いてWebサイトに不正スクリプトを埋め込む改ざん攻撃が急増している．この攻撃は，サイトを閲覧したユーザにマルウェアを感染させることを目的としており，感染源が正規サイトであることから一般ユーザ側での対策が困難となっている．著者らは先に，実際に改ざんサイトや不正スクリプトの調査を通して得たサイトタイトルやスクリプトの記述パターンの分析データを元に，ユーザがサイトを閲覧する際に改ざんサイトであるか自動で検出する方式を提案した．本論文では，実データをさらに入手し分析することにより方式の詳細化を図るとともに，その機能を持つシステムDICE(Detection of Injected Site using Cyber search Engine)をプロキシサーバ上に試作開発するとともに，精度や速度を，実験を通して評価したので報告する．

Development of Defaced Sites Automatic Detection System DICE

Tatsuya TANAKA† Yusuke TAMURA†† Toshifumi KAI＊ Ryoichi SASAKI†

†Tokyo Denki University
2-2, Kanda-Nishiki-cho, Chiyoda-ku, Tokyo, 101-847 JAPAN
tanaka@isl.im.dendai.ac.jp

††SEIKO EPSON CORPORATION ＊Panasonic Electric Works Co.,Ltd

Abstract Recently, the manipulation attack to website embedding malicious script using SQL injection vulnerability is increasing. Purpose of this attack is to infect users PC which browse the site with the malware. It is difficult for users to protect this attack because source of infection is website that has high evaluation. In the former paper the authors proposed the method with the function of automatic detect whether there was a defaced site. In this paper, we propose the refined method using the increased data, and we describe the developed system named DICE (Detection of Injected Site using Cyber search Engine) system on proxy server. We also present evaluated results on precision and speed of DICE through an experiment with this proxy server.

1. はじめに

2000 年に発生した中央省庁の Web サイト改ざんに代表されるように，Web サイトへの改ざん攻撃は以前から行われていた．これらの改ざん攻撃は，サイトデザインを改ざんすることで，攻撃者自身の主張・メッセージを訴えることが目的であったといえる．しかし近年，Web サイトを閲覧した一般ユーザの PC にマルウェアを感染させることを目的として，サイトに不正なスクリプトを挿入する新たな Web サイト改ざん攻撃が増加している．2008 年 3 月には，日本をターゲットとした大規模な攻撃が行われ，多数の正規サイトが改ざんの被害に遭った．現在でも攻撃は継続しており，個人 HP や地方自治体などの多くのサイトが被害を受けている[1].

この改ざん攻撃に対し，サイト管理側での対策はもちろんのこと，標的となっている一般ユーザ側でも対策が求められている．しかし，下記のような理由からユーザ側での対策が困難になっている．

- マルウェア感染源が正規サイトである
- サイトを閲覧しただけでマルウェアに感染する可能性がある
- 視覚的な情報では改ざんの認識が難しい

このため著者らは，実際の改ざんサイト及び挿入された不正スクリプトを調査・分析し，そこから得られた特徴を用いて改ざんサイト・不正スクリプトを判別するための検知手法を提案した[2].

本論文では，実データをさらに入手し分析することにより方式の詳細化を図るとともに，その機能を持つシステム DICE(Detection of Injected Site using Cyber search Engine)をプロキシサーバ上に試作開発するとともに，精度や速度を，実験を通して評価したので報告する．

2. ユーザ標的型 Web 改ざんについて

2.1 改ざん方法

ユーザ標的型 Web 改ざん攻撃における Web ページの改ざんは，SQL インジェクションと呼ばれる手法を用いて，無差別かつ広範囲に行われている．

SQL インジェクションとは，正規サイト上でデータベースと連携して運用されている Web アプリケーションの脆弱性を突き，データベースを不正に操作することで，データベース内の情報の不正取得や改ざんなどを行う行為である．ユーザ標的型改ざん攻撃では，この手法を用いて JavaScript や iframe などのスクリプトが不正に Web ページ内に挿入されている．

2.2 攻撃の流れ

Web サイトの改ざんから一般ユーザにマルウェアが感染するまでの流れを図 1 に示す．

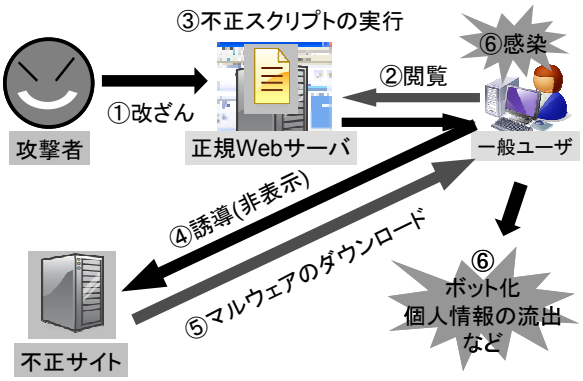

図 1：ユーザ標的型 Web 改ざんの流れ

まず, 2.1 で述べた手法で正規サイトが改ざんされる(図 1,①)．このとき挿入されるのは，以下の図 2 のようなスクリプトである

```
<script src=http://誘導先 URL><script>
<iframe src=http://誘導先 URL><iframe>
```

図 2：挿入される不正スクリプトの例

一般ユーザが改ざんされた正規サイトを閲覧した場合(図 1,②)，不正スクリプトが実行され(図 1,③)，スクリプト中に記述された不正サイトの URL 先に誘導・接続される(図 1,④)．不正スクリプトには誘導先のサイトが表示されないよう細工がされているため，一般ユーザは不正サイトに接続されていることを認識できない．そして，不正サイトからマルウェアがダウ

ンロードされ(図 1,⑤), 一般ユーザの PC に脆弱性があれば感染してしまう(図 1,⑥). このとき InternetExplorer, FlashPlayer など多種多様なアプリケーションの脆弱性が利用されるので，一部のアプリケーションのアップデートだけではこの攻撃に対処しきれない.

2.3 ユーザ側における対策の必要性

Web 改ざん攻撃が発生する根本的原因は，Web サイト側の脆弱性にあるといえる．すなわち, Web サーバ管理者側で Web サイトの脆弱性をなくしてしまえば，改ざん自体を防ぐことが可能である．しかし，全ての Web サーバ管理者がこうした対策を行っているとは限らないため，ユーザ側でも対策をとる必要がある．Web における簡便なマルウェア対策として「怪しいページにアクセスしない」,「怪しいファイルをダウンロードしない」という人の手による対策方法が挙げられる．しかし，ユーザ標的型 Web 改ざん攻撃に対しては，下記のような理由から全く効果がなくなってしまう．

- 改ざんされた正規ページを閲覧しただけで感染の恐れがある
- 改ざんを直感的に認識できない
- 不正サイトへの誘導が不可視である

このようなことから, ユーザ側においては「機械的に改ざんサイト及び不正サイトへのアクセスを防止する」という対策が必要となってくる.

3. ユーザ側における既存対策手法

3.1 既存手法の概要

「改ざんサイト及び不正サイトへのアクセスの防止」という対策の具体的方法として，危険なサイトに対して事前に警告を出す Google のセーフブラウジング機能の活用や，不正サイトや不正スクリプトのブラックリストによるアクセス制限などが挙げられる.

3.2 問題点

セーフブラウジング機能の警告は，Google のクローラが巡回した際にサイトの危険性を識別して出される．このことから，改ざんが行われてからクローラが巡回するまでの間は警告を出すことができないという問題点がある．ブラックリストによるアクセス制限は，この問題に対処することができるが，リストに掲載されていない未知の不正スクリプトに対しては対処できないという新たな問題が挙がる.

そこで我々は，この「未知の不正スクリプト」に対処するための方法として，不正スクリプト共通の特徴を用いて判定を行うことを考えた．次章では，その特徴を抽出するために行った調査・分析について述べる.

4. 調査

4.1 調査概要

先の調査では 1 次調査，2 次調査を通して次の下記の 5 つの特徴に注目すべきであるといった.

①タイトルの改ざん
②スクリプトの多重挿入
③スクリプト名の偏り
④不正誘導先 URL のトップレベルドメインの偏り
⑤スクリプト内の属性の設定・未設定

本章では新たにデータを取ることで，先の調査で発見した統計との比較を行い，見つかった特徴が現在でも有効であるかどうか，また新しい特徴の発見のために調査，分析を行った.

調査には，表 1 に示す環境で文献[2]のプログラムに改良を加えて行った．また今までは検索エンジンに Google の”Google AJAX Search API”を採用していたが，今回は Yahoo! JAPAN の”ウェブ検索 API”も使用した.

表 1：調査環境および開発環境

OS	Windows XP Professional
CPU	Intel® Pentium® M processor 1.20GHz
メモリ	760MB
開発環境	Visual Studio 2008 Professional Edition
	Google AJAX Search API
	Yahoo!ウェブ検索API
検索エンジン	Google
	Yahoo! JAPAN

4.2　検索エンジンによる特徴の調査

4.2.1　特徴①，②の再調査

調査にはインターネット上で公開されている不正誘導先ドメイン[3]を検索キーワードとして Google を使用しサイト検索を行い，先の調査との比較を行った．（表 2）

表 2：ブラックリストからの調査

	タイトル改ざん		多重挿入		合計
	有り	無し	有り	無し	
文献[2]の調査結果	10564 (52%)	9875 (48%)	1494 (7%)	18945 (93%)	20,439
8月30日	8500 (98%)	135 (2%)	519 (6%)	8116 (94%)	8635

表 2 よりタイトル改ざんは先の調査より現在の方が増加しており，ほぼ全てのサイトがタイトル改ざんされている．また多重挿入の割合についてはあまり変わった所が見られなかった．

また先の調査の方が多く収集出来ており,現在より約 3 倍弱あったが，これは現在 SQL インジェクションが減少傾向にあることと,データベース側で対策が取られているためと言える．

4.2.2　特徴③，④，⑤の再調査

この調査には「タイトルの改ざん」と「多重挿入」という特徴を有したサイトを Google と Yahoo!を使用して検索した. 11 月 4 日は文献[2]での調査日となっている．

特徴③の検索結果は表 3 のようになった．

表 3：特徴③の調査

日付	検索エンジン	単語1文字	単語2文字	ngg.js	script.js	その他
11月4日	Google	29%	10%	19%	8%	33%
4月21日	Google	68%	8%	8%	6%	10%
5月26日	Google	65%	9%	9%	5%	13%
6月21日	Google	69%	8%	8%	4%	11%
7月14日	Google	75%	4%	6%	2%	13%
8月25日	Google	79%	4%	6%	1%	9%
8月13日	Yahoo	65%	14%	4%	3%	14%

表 3 より先の調査で行った時よりも単語 1 文字のスクリプト名が多くなっている事が分かる．また最近では”ngg.js”が減少してきている．

先の調査でスクリプト名が単語 2 文字となっていたのは，”js.js”，”ri.js”がほとんどであった．今回は単語 2 文字に対しての統計を取った結果，新たに”kr.js”，”cn.js”が見つかった．（図 3）

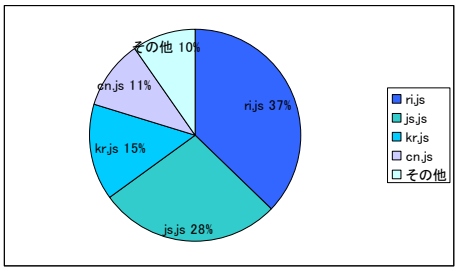

図 3：スクリプト名が単語 2 文字の内訳

また特徴④の検索結果は以下のようになった．

表 4：トップレベルドメインの偏り

日付	検索エンジン	.com	.ru	.cn	その他
11月4日	Google	42%	34%	16%	8%
4月21日	Google	40%	11%	33%	16%
5月26日	Google	35%	12%	35%	18%
6月21日	Google	35%	9%	36%	20%
7月14日	Google	27%	6%	33%	34%
8月25日	Google	22%	5%	31%	42%
8月13日	Yahoo	30%	5%	30%	35%

表 4 より”.ru”は先の調査よりも減少しており，”.cn”は増加している．また最近ではその他が増え，そのほとんどが”.in”，”.org”，”.net”となっていた．

特徴⑤は全ての不正スクリプトで”type”や”language”と言った属性が未設定であった．

以上の再調査から特徴③，④に対して新しい

特徴を見つけることが出来たが，先の調査で発見できた5つの特徴以外の物を発見することは出来なかった．よって現在でも文献[2]の判定アルゴリズムで改ざんサイトを判定できることが判明した．

5. 提案システム

5.1 提案システムの概要

文献[2]で一部組織内ネットワークのプロキシにおいて改ざんサイトの判定・検知を行い，通信の遮断もしくは警告を行うシステムを提案した．図4にそのシステム図を示す．

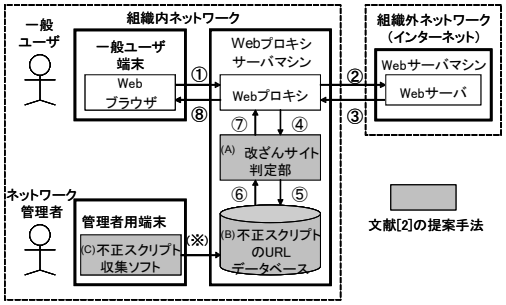

図4：提案手法のシステム図

図4の(A)，(B)，(C)部分の総称をDICEと名づけた．改ざんサイト判定部(図 4,α)では，発見された改ざんサイトの 5 つの特徴を用いた「特徴分析による判定」と，既存の「ブラックリストによる判定」によって改ざんサイトの判定を行う．図5にその判定アルゴリズムを示す．

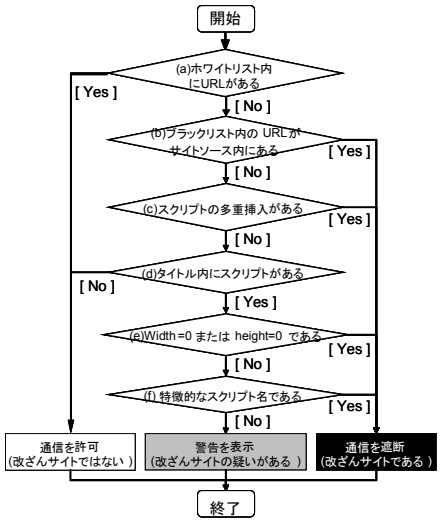

図5：改ざんサイト判定アルゴリズム

今回はこの判定アルゴリズムを実装したプロキシサーバを作成した．

5.2 プロキシサーバの実装

プロキシサーバの開発には表1で示した開発環境で行った．言語はC#を用いており，約600ステップである．このプロキシサーバの処理の流れを以下に示す

- 一般ユーザによるWebページアクセス時

1：Webブラウザからの Get 命令で，プロキシは対象 URL のソースファイルを取得する(図4,①~③).

2：取得したソースファイルについて，図 5 の判定アルゴリズムを用いて改ざんの有無をチェックする(図4,④~⑦). また，データベースに登録されていない不正スクリプトの URL を発見した場合は，データベースに追加する．

3：問題がなければブラウザで表示する(図4，⑧). 改ざんサイトの疑いがある場合は，警告ページを表示し，ユーザが閲覧したい場合は警告ページ内のリンクをクリックしてもらい，自己責任でページを表示させる．改ざんサイトである場合，通信を遮断し，危険なページであることを表示する．

5.3 実験と考察

5.3.1 機能実験

判定アルゴリズム自体の性能を実験するために一時的にブラックリストの機能を止めて動作させた．実験に使用する URL は 4.2.1 で収集したサイトの URL を無作為に 100 件選び調査した．

その結果，判定アルゴリズムで判定できたページは 13 件であった．この 13 件の内 12 件がタイトル挿入のみで特徴的なスクリプト名で判断できていた．残りの 1 件は多重挿入が行われていたため判断することが出来た．またこの 13 件の内 Google の警告が表示されていなかったのは 8 件存在した．この 8 件について実際にアクセスしたところ,不正なプログラムをダウ

ンロードすることは無かった．これは改ざんする際に何らかの問題があったため Web ページがスクリプトタグを HTML のタグとして認識しなかったからである．しかしスクリプトタグの誘導先はブラックリストに載っているドメインであったため，ブラックリストの機能を有効にしていた場合でも危険なページとして検知する．よってブラックリストの機能を抜いた判定アルゴリズムだけで危険なページを判定できることが分かった．

以上より作成したプロキシサーバは Google で警告の出ていない改ざんサイトに対して通信を遮断することが出来た．このことから DICE を搭載したプロキシサーバを使用することでより安全に Web ページを閲覧出来ることが判明した．

5.3.2 速度実験

プロキシサーバが改ざんされていないページや,改ざんされているページを判断し警告ページや危険なページであることを表示するまでにどれくらい掛かるか測定してみた．

改ざんされていないページとして Yahoo! JAPAN を表示したところ表示するまでに33秒かかった．プロキシサーバを使用しない場合として，回線速度 100Mbps の有線で測定したところページを表示するまで 3 秒であった．

改ざんされた疑いのあるサイトを表示したところ約 3 秒掛かり,改ざんされたサイトでも同じく 3 秒掛かった．

改ざんサイトや改ざんされた疑いのあるサイトの場合，ユーザが気にならない時間で警告ページを表示することが出来た．しかし改ざんされていないページを表示する際はプロキシサーバ無しの場合より 30 秒ほど余分にかっており，今後さらに時間の短縮を図っていきたい．

６．おわりに

著者らは先に，実際に改ざんサイトや不正スクリプトの調査を通して得たサイトタイトルやスクリプトの記述パターンの分析データを元に，ユーザがサイトを閲覧する際に改ざんサイトであるか自動で検出する方式を提案した．本論文では，実データをさらに入手し分析することにより方式の詳細化を図るとともに，その機能を持つシステム DICE(Detection of Injected Site using Cyber search Engine)をプロキシサーバ上に試作開発するとともに，精度や速度を，実験を通して評価し，実用化の見通しが得られた．

今後の課題はプロキシサーバの速度向上と性能の向上を行っていく必要がある．また現在は改ざんされたサイトにしか有効でないため，フィッシングサイトなどにも対応していくことが目標になってくる．調査についても定期的に実施し,新たな特徴を発見しだい順次追加していくことで新しい不正サイトに対応していく．

参考文献

[1] LAC, 侵入傾向分析レポート Vol.11, 2008.09.17

[2] 田村 佑輔, 甲斐 俊文, 佐々木 良一, “ユーザ標的型 Web サイト改ざんに対する検索エンジンを用いた検知手法の提案”情報処理学会 CSEC 研究会（2009 年 3 月）

[3] RO アカハック対策スレの hosts ファイルまとめ臨時
http://sky.geocities.jp/ro_hp_add/