リンク情報の時間変化に着目したWeb改ざん検知支援システムの提案

望月 翔太† 高田 哲司†

†電気通信大学
182-8585 東京都調布市調布ヶ丘 1-5-1
m-shouta@uec.ac.jp,zetaka @ computer.org

あらまし Drive-by Download 攻撃など，マルウェア配布の手段として Web 閲覧による受動的攻撃が問題となっている．この攻撃は，Web ページ改ざんが起点となっており，その改ざんに気づけることが重要となる．この問題への対策としてブラックリストが存在するが，即時性と網羅性の双方において限界がある．そこで本論文では，別の Web 改ざん検知手法としてクライアント側だけによる新たな Web 改ざん検知手法を提案する．提案手法は「Web 閲覧者は Web ページを繰り返し閲覧する」という仮定のもと前回訪問時の構成情報と今回閲覧時の構成情報を比較することで，改ざん可能性に関する情報を提示することにある．

Web Falsification Detection Support System by Using A Change of The Link Information

Shota Mochizuki† Tetsuji Takada†

†Graduate School of Informatics and Engineering,
The University of Electro-Communications
1-5-1 Chofugaoka, Chofu-si, Tokyo, 182-8585, JAPAN
m-shouta@uec.ac.jp,zetaka @ computer.org

Abstract Such as Drive-by Download attack, passive attack by Web browsing has become a problem as a means of malware distribution. In these attacks, a Web falsification becomes a starting point. Therefore, it is important to discover a Web falsification. Blacklist exists as a measures to this problem, however there is a limit in both completeness and immediacy. In this paper, we propose a new Web falsification detection method based on only the client side as a Web falsification detection another approach. By compares the time variation of the Web pages in the assumption that “Web visitors browses the again same Web page”, the proposed method presents shows the information about the potential falsification.

1 はじめに

昨年度の日本におけるサイバー攻撃の「3 大脅威」として,「正規 Web サイト改ざん」,「不正プログラムによるオンライン詐欺」,「アカウントリスト攻撃」が挙げられる [1]．この 3 大脅威のうち，正規 Web サイト改ざんと不正プログラムによるオンライン詐欺の 2 つが Web を介した攻撃に含まれる．

Web を介した攻撃は 2 種類に分類可能である．Web サーバを対象としたものと Web 閲覧者を対象としたものである．Web 閲覧者を対象としているものは，オンライン詐欺等の不正プログ

ラムへの感染を目的としたものや，フィッシング詐欺等があげられる．これらの攻撃は「Web 改ざん」が起点となる．したがって，Web 改ざんされていることを閲覧者が判断することができれば被害を防ぐことにつながる．頻繁に行われている Web を介した攻撃として，「Drive-by Download 攻撃」と「水飲み場攻撃」がある．Drive-by Download 攻撃とは，Web ページを閲覧した際に，マルウェアを感染させる攻撃である．改ざんされた Web ページを閲覧，他のサーバへと自動的に接続し，アプリ等の脆弱性を利用してマルウェアに感染させる．この攻撃は，自動的に感染するためユーザが気づきにくい性質を持つ．IBM が公開している Tokyo SOC Report[2] によると，Drive-by Download 攻撃の件数は 2012 年に比べ，2013 年は下半期比 2 倍となっているとの報告がある．水飲み場攻撃とは，標的型攻撃の一つである．標的とする特定の組織が改ざんされた Web ページをアクセスした際に IP アドレスを特定する．特定した IP アドレスからアクセスした時のみにマルウェアを感染させる攻撃である．感染条件が限定的であるため，発見することが困難である．

JPCERT/CC に報告された Web 改ざんの件数は，2014 年 1 月〜6 月の期間で 2,624 件であった．昨年の件数 (2013 年 7 月〜12 月で 4,378 件) より減っているが，400 件/月のペースで Web 改ざんの報告があり，発見されていないだけでさらに改ざんされた Web サイトが存在していると言われている [3]．

Web 改ざんに対し，サーバ側とサーバ+クライアント側の対策がある．サーバ側の対策として，Web サーバを構築しているコンテンツを常に監視し，変更があった際にアラートを行う仕組みがある．サーバ+クライアント側の対策として，悪性サイトと既に判明しているサイトのリスト (ブラックリスト) を利用したアクセス制御がある．

エンドユーザが Web 閲覧中に Web 改ざんに気づくことは少ない．既存の主たる対策方法である Firewall やウィルス対策ソフトは，Web 改ざん検知に効果がない．また，ブラックリストは，誤検知や網羅性，そして正規サイトの改ざんが反映されにくい問題を抱えている．攻撃者は，Web ブラウザやアプリの脆弱性を突いた攻撃をしてくるため，エンドユーザが取れる行動は危険なサイトに訪れない，または既に発見された脆弱性を防いだ最新版に常に更新することしかなかった．これらの行動は，サーバ側が対処するまで行動に移すことができない．

サーバ側が Web 改ざんに対処する速度に影響せずに Web セキュリティを向上させるため，本研究ではクライアント側のみによる Web 改ざんの検知を目的とする．クライアント側のみを対象とする理由として，以下の 2 点を挙げる．

a 「サーバ側の対策待ち」以外の方法の提供
サーバ側が Web 改ざんに気づき，対策を行うまでユーザは被害を未然に防ぐ手段がない．クライアント自身が Web 改ざんに気づくことが可能となれば，Web 改ざんを起点とした攻撃から自衛することが可能となる．

b 悪性サイトを網羅する必要性の減少
個人でアクセスするサイトは限られている．ユーザ個人としては，自身が訪れるサイトのみ安全であれば良い．ユーザが訪れるサイトのみをクライアント側でチェックすることで，解析時間等の削減が可能となる．

我々は，Web ページに記載された URL を情報源とし，時間変化に基づく Web 改ざん指標提示システムを提案する．

2 既存対策・研究

Web 改ざんに対する対策・研究は，サーバ側，サーバ+クライアント側の 2 種類ある．それぞれについて以下に記述する．

サーバ側による Web 改ざんを検知するシステムとして，isAdmin[4] がある．Web サイトのコンテンツを定期的に取得し，hash 値，ファイルサイズ，更新日付を基に更新を検知する．更新を検知するたびに，管理者にアラームを通知する仕組みである．isAdmin は Web サーバのコンテンツをシステム内部で保持しており，Web 管理者は，isAdmin のコンテンツに対し更新を行う．isAdmin は，更新されたファイルを Web

サーバへと反映させる．isAdmin を介さないでコンテンツが更新された場合，その更新が正規の管理者による更新ではないと判断する．

竹森ら [5] は，Web サーバリモート監視による Web 改ざん検知のためにコンテンツの中身を精査することで Web 改ざんの検知を行った．Web 改ざんを判定するために，１）末尾挿入，２）キーワード，３）HTML ファイルの不完全構造，４）タイトルの変化，５）エンコードの変化，６）背景色または文字色の変化　の 6 つの基準を使用した．6 つの特徴のうち，1 つ以上該当していた場合，Web 改ざんとして検知し，アラームを出す仕組みである．正規コンテンツを 89 件，改ざんコンテンツを 73 件としてシステムを評価した結果，正判定率 100%，誤判定率 1.1%となり，小さな誤判定率で検知することが可能となった．

サーバ+クライアント側における対策として，ブラックリストの利用したアクセス制御がある．有名なブラックリストとして，Google の Safe Browsing[6] がある．Google が持つクローラにより探索し，危険と判断したサイトをブラックリストに登録･更新する．ユーザが訪れたサイトがブラックリストに登録されていた場合，警告を表示する．Google Safe Browsing は Google Chrome 等のブラウザで利用されている．

田村ら [7] は，組織内ネットワークのプロキシによって改ざんサイトを判定，怪しいサイトに対し警告または通信の遮断を行うシステムを提案した．改ざんサイトの判定に，１）ブラックリストに含む URL があるか，２）スクリプト多重挿入の有無，３）タイトル内のスクリプトの有無，４）タグの属性が「width=0 または height=0」であるか，５）特徴のあるスクリプト名　の 5 つの基準を使用している．1) から順にチェックしていき，改ざんサイト，改ざんの疑いのあるサイト，正常なサイトを判定する．ネットワーク管理者が不正スクリプトの URL をプロキシ内のデータベースに追加していくことで，新たな改ざんサイトを発見，検知することが可能となる．

3　提案システム

3.1　概要

我々は，クライアント側のみで Web 改ざんの検知を行う．クライアント側のみで検知を行うため，サーバ内部の情報は取得することができない．そこで，ユーザが以前閲覧した際の Web コンテンツ情報と現在閲覧している Web コンテンツ情報を比較することにより，Web 改ざんの検知を行う．ユーザは，Web サイトを閲覧する際，定期的に同じ Web サイトを訪れる傾向がある．ブログやニュースサイト等日々更新があるサイトなどは毎日のように訪れると思われる．このユーザの傾向を利用することで，以前の Web コンテンツ情報と現在の Web コンテンツ情報を比較することが可能となる．

比較に使用する Web コンテンツ情報について説明する．Web コンテンツ情報には，HTML 内に記述されている URL を情報源として使用する．URL を情報源とする理由として，攻撃者が Web サイトを改ざんする際にマルウェア配布サイトへ誘導するために URL を追加または変更する必要があるためである．このマルウェア配布サイトなどの誘導先の URL を捕捉し，特徴点とすることで，Web 改ざんを検知するための手がかりとなる．

3.2　実装

我々は提案するシステムの構成図を図 1 に示す．本システムは，Google Chrome の拡張機能を利用して実装した．システムは「情報取得部」,「情報比較部」,「視覚化処理部」の 3 つの処理部から構成される．

情報取得部では，Web サイトからコンテンツ情報を抽出する．ユーザが閲覧中の URL を入力値とし，HTML ファイル内に記載されているタグを抽出する．抽出するタグは，属性情報に URL を含むことが可能であるタグである (例：A タグ，IFRAME タグ等)．抽出したタグから URL を取り出し，タグ名と関連付ける．また，抽出した URL をブラックリスト (Google Safe Browsing API) と照合する．以降「タグ名」,

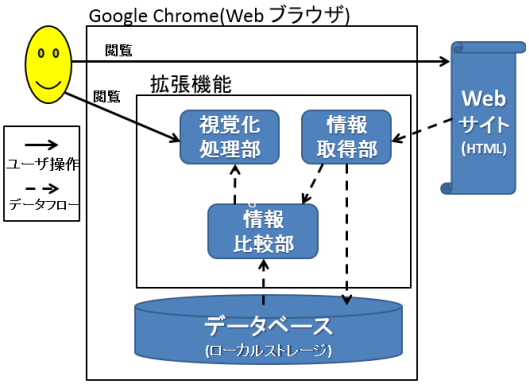

図 1: システム構成図

「URL」,「ブラックリストの照合結果」を構成情報と呼称する．訪れているサイトがはじめての場合，情報取得部で取得したデータは，データベースに保存され，視覚化処理部に渡される．2 回目以降の場合，取得したデータは情報比較部に渡される．

情報比較部では，データベース内に保存されている構成情報 (過去に訪れた際の Web サイトの構成情報) と情報取得部から入力された構成情報の比較を行う．2 回目以降にサイトに訪れた際，情報処理部では，データベース内の構成情報 (過去情報) と情報取得部で取得した構成情報 (新規情報) 内の URL を比較し，以下の 3 つの種類に分類される．

- 消失：過去情報に存在するが新規情報に存在しない URL
- 出現：過去情報に存在しないが新規情報に存在する URL
- 現存:過去情報，新規情報共に存在する URL

また，新規情報と過去情報のデータの差を「変更率」として計算する．

変更率は，消失に分類されるデータの割合を「旧 URL 率」，出現に分類されるデータの割合を「新 URL 率」として表示する．過去情報に含まれる URL の集合を S_a，新規情報に含まれる URL の集合を S_b とすると，消失に分類される URL の集合は $S_a - (S_a \cap S_b)$, 出現に分類される URL の集合は $S_b - (S_a \cap S_b)$ と表される．旧 URL 率 P_{new}，新 URL 率 P_{old} は以下の式 1 で計算する．

$$\begin{aligned} \text{旧 } URL \text{ 率 } P_{old} &= \frac{n(S_a-(S_a \cap S_b))}{n(S_a)} \\ \text{新 } URL \text{ 率 } P_{new} &= \frac{n(S_b-(S_a \cap S_b))}{n(S_b)} \end{aligned} \quad (1)$$

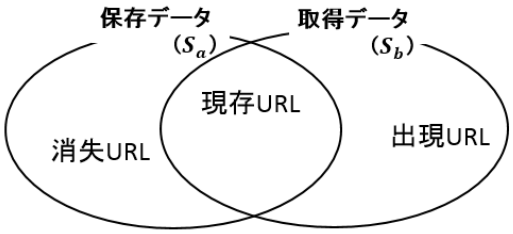

図 2: URL の分布

視覚化処理部では，情報比較部からの入力値を基に構成情報を視覚化する．視覚化画面を図 3 に示す．視覚化画面では，中心に各タグの割合が円グラフで表現されている．この円グラフを描くために使用するタグはシステム上部に設置しているラジオボタンを使用することで，取捨選択可能である．中心の円グラフと線で結ばれている円 (ノード) は一つ一つ，URL のホストを表現している．ノードには，Web サイトを識別しうるアイコンを表示させている．各ノードは，各タグごとに等間隔で表示している．

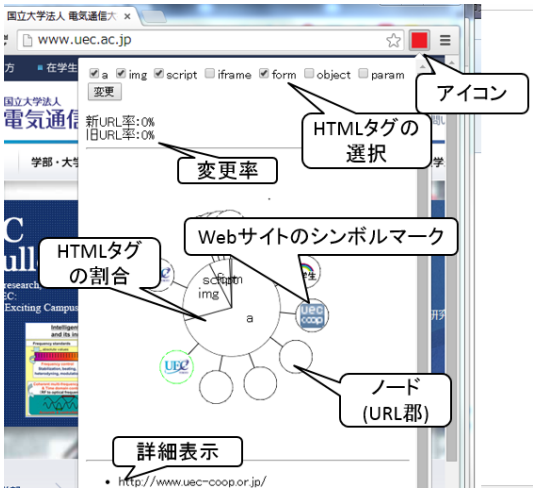

図 3: 視覚化方法

表 1: ノード内部データ

ノード	内部データ (URL)	比較結果
ノード 1 (example.com)	http://example.com/a.html	現存
	http://example.com/b.html	現存
	…	…
ノード 2 (example2.com)	http://example2.com/a.html	現存
	http://example2.com/b.png	出現
	…	…

各ノードは，情報比較部で行った比較結果を基に色によって表現する．各ノードは同じホストを所持する URL の集合体である．ノードに属する URL は情報比較部により「消失」,「出現」,「現存」にそれぞれ分類されている．各ノードは色によって表現されており (図 4)，色づけ規則は表 2 に順ずる．ノードが一色で塗りつぶされているとき，Web ページで利用したことのないホスト先へのリンクが張られたことを意味する．これは，攻撃者がマルウェア配布サイト等に誘導する際に用いる URL が元のサイトで利用している URL と異なることがあることから，この表現方法を使用した．

表 2: ノードの色付け規則

ノードの種類 (色)	URL の種類
現存 (黒枠線)	現存のみ
出現 1(赤枠線)	出現+現存
出現 2(赤塗りつぶし)	出現のみ
消失 1(青枠線)	消失+現存
消失 2(青塗りつぶし)	消失のみ
複合 (緑枠線)	出現+消失 (+現存)

グラフの下部にノードが表すホストを含む URL が一覧表示される．一覧表示される URL は青色，赤色，黒色の何れかで表示され，それぞれ「消失」,「出現」,「現存」を意味している．

3.3 利用方法

Web ブラウザ (Google Chrome) で Web ページを閲覧しているときに，Google Chrome の

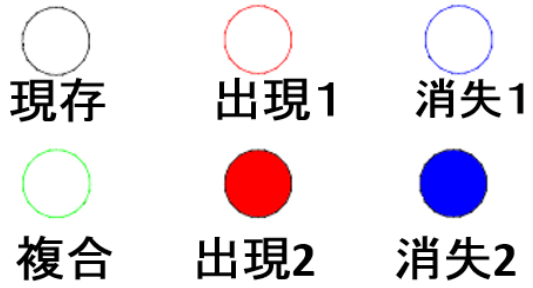

図 4: 各ノードの色

右上に表示されるアイコンをクリックすることで本システムの視覚化画面が表示される (図 5)．図 5 より，A タグから 7 つ，IMG タグから 1 つ，SCRIPT タグから 3 つ，FORM タグから 1 つのノードが伸びている．各ノードには，アイコンが付与されているため，アイコンからどのノードが何のホストを示しているかがわかる．視覚化画像から見て取れる情報として，ノード上に表示されているアイコンから学生生協のページや google のページが Web ページ内で参照されていることがわかる．アイコンが見づらい場合は，上部に設置されているラジオボタンを使用して表示するタグを選択することで調節可能である．

4 システム運用例

本章では，システムの視覚化事例を紹介する．

まず，正規サイトを訪れた際の視覚化事例を紹介する．図 6 のような表示になる．視覚化画面を確認すると，全てのノードが黒枠線で描かれているため，前回に訪れたときと比べ，Web ページが更新されていないことを意味する．図

図 5: 視覚化画面 (電気通信大学 HP)

7は，Yahoo!JAPANのトップページにアクセスしたときの視覚化画像である．Yahoo!JAPANのサイトは，ニュースを配信しているため，頻繁にURLが変わるため，取得する構成情報は毎回異なる．ニュースサイトは図7のように，緑色のノードがAタグに現れるという特徴がある．これは，ニュースの変更によって生じるAタグのURLの変更が，パスの部分での変更に限られるためである．

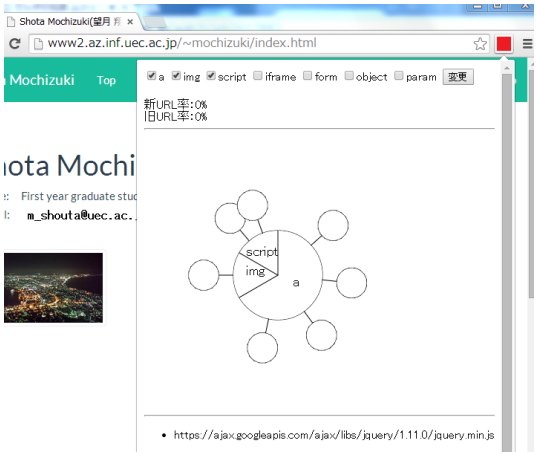

図 6: 更新頻度の少ない Web サイト

次に，Web改ざんの種類は大きく分けて2種類あり，新たなファイルをサーバに蔵置するファイル蔵置型とJavaScript文等のスクリプトをHTMLファイルの一部に挿入するスクリプト挿入型である．

図8は，ファイル蔵置型のWeb改ざんが行われたWebサイトを訪れた際の表示である．各

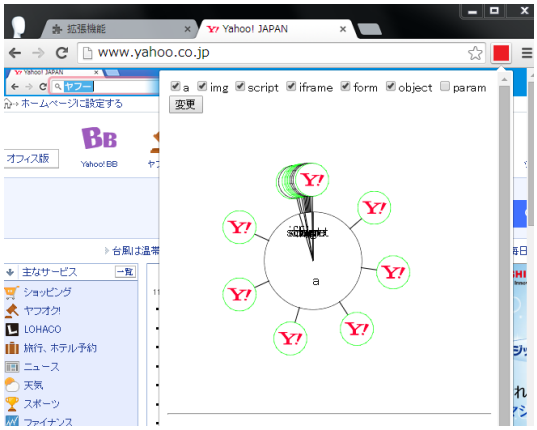

図 7: 更新頻度の多い Web サイト

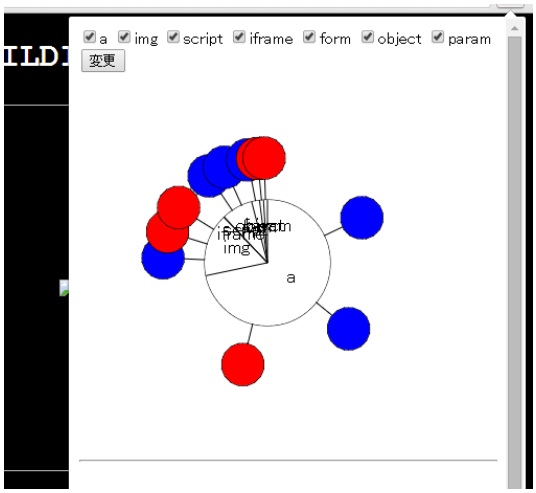

図 8: ファイル蔵置型

ノードに注目すると，各ノードが全て赤一色または青一色のどちらかで描かれている．また，図上部に記載されている変更率を確認すると，旧URL率，新URL率が共に100%になっている．これは，前回訪れた際と比較し，全てが書き換わってしまっていることを示す．このような表示が見られるとき，閲覧中のページはWeb改ざんの被害に遭っていると判断することが可能である．

図9，10はスクリプト挿入型のWeb改ざんが行われた際の表示である．それぞれ，外部サーバのスクリプトを指定したURLに変更されたケースと新たにタグを挿入されたケースとなる．図9より，IFRAMEタグに赤色で塗りつぶされ

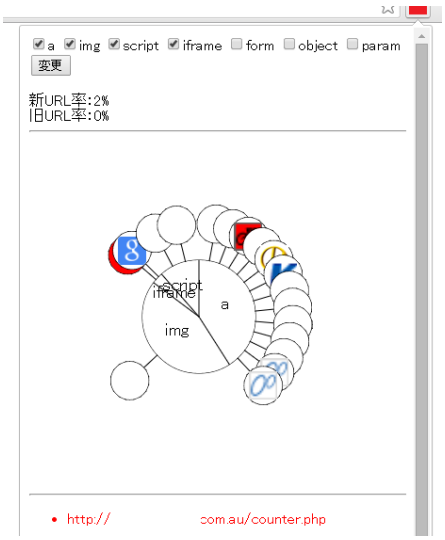

図 9: スクリプト挿入型 (iframe)

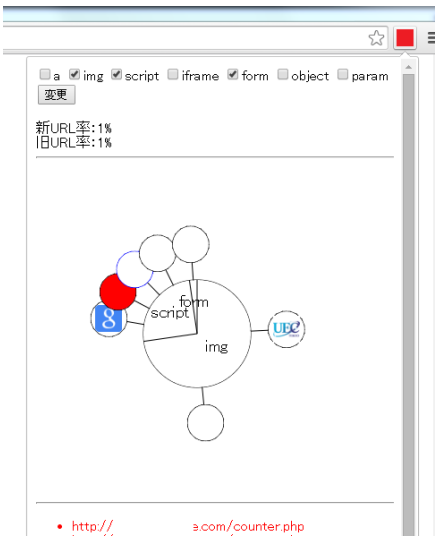

図 10: スクリプト挿入型 (script)

たノードがあることがわかる．このノードの詳細表示をすると,「counter.php」というファイルであることがわかる．counter.php で検索すると，外部のサーバに誘導し，マルウェアを感染させるためのファイルであることが判明した．このファイルは，style 属性で「visibility:hidden」が指定されている．これは，要素を非表示にする設定で，Web ページを見るだけでは発見することはできない．図 10 より，SCRIPT タグで赤色で塗りつぶされたのノードと青枠線のノードが確認できる．これは，SCRIPT タグが書き換えられて過去には通信していなかったホスト先へのリンクが張られたことを意味する．赤色で塗りつぶされたノードに属する URL を表示すると,「gifimg.php」というファイルへのアクセスであった．gifimg.php で検索すると，マルウェアに感染させるためのファイルであることが判明した．

5 考察

5.1 利点と制約

本システムにより，Web ページの正規の変更とは思われない怪しい更新を発見できる．これにより，ユーザはブラックリスト等を運営しているサーバが未発見の Web 改ざんに対し気づくことが可能となった．ユーザがサーバ側の対処を待たずして，危険性を認識し次なる対策 (例：クライアント計算機が感染していないかのチェック，Web サーバ管理者への報告等) をいち早くとることのできる機会が与えられる．

本システムは，クライアント側に提供されたファイルを基に作成している．そのため，解析の対象となるファイルは HTML ファイルである．もし，Web ページが PHP 等のサーバ側言語で構成されており，水飲み場攻撃といった IP アドレスによって送信するファイルが異なる場合，全ての Web 改ざんを見つけることは不可能である．また，Web 改ざんを視覚化手法で発見しても，Drive-by Download 攻撃等の被害に遭っている可能性が高い．未知なる Web 改ざんによる被害を未然に防ぐための技術が今後求められる．

5.2 今後の課題

今後の課題として 4 つほど挙げる．一つ目は初めて訪れたサイトへの対応である．本システムは，過去に一度以上訪れているサイトであることが前提である．そのため，初めて訪れたサイトが Web 改ざんの被害に遭っていた場合，検知することはできない．

二つ目は JavaScript への対応である．JavaScript によってクライアント側で動的に URL が

生成されている場合，本システムではHTMLのみからURLを抽出しているため検知することができない．ユーザが検知するためにはWebサイトを構成する全てのURLを精査する必要がある．

三つ目は短縮URLへの対応である．サイト内で短縮URLが利用されていた場合，ホストが全て同じになってしまい元のURLが隠蔽されてしまう．短縮URLを元のURLへと変換する工程が必要である．

最後に，システムの日常的利用の問題である．ユーザが閲覧したサイトがWeb改ざんの被害に遭っているかを判断するために，本システムのアイコンをクリックする必要が生じる．訪れる全てのサイトをチェックするために，毎回拡張機能を動作させる必要があり，ユーザが利用に手間を感じ利用しなくなってしまう可能性がある．

6 おわりに

Webサイトを通じてWeb閲覧者を攻撃する手法が問題になっている．これらは，Web改ざんが攻撃の起点となっている場合が多い．しかし，一般的な対策手法となっているブラックリストでは，全てのWeb改ざんサイトを網羅できない．また，Web改ざんの被害が発生後，ブラックリストに更新されるまでユーザは対処することができない．ユーザがWeb改ざんに気づくことが可能となれば，被害を最小限にとどめることが可能となる．そこで，本研究ではWeb閲覧者が閲覧中のWebサイトの改ざんに気づくために必要な情報を提供可能とするシステムを提案した．

ファイル蔵置型やスクリプト挿入型のWeb改ざんに対して指標を提示することで，異常な状況を判断することが可能となった．本システムを利用することで，Web改ざんの早期発見を補助することが可能となる．

参考文献

[1] TREND MICRO : サイバー攻撃の傾向と実態，入手先<http://www.trendmicro.co.jp/jp/sp/asr-2013/#threat2014>(参照 2014-08-23)

[2] IBM Security Service : 2013 年下半期 Tokyo SOC 情報分析レポート，入手先<http://www-935.ibm.com/services/multimedia/tokyo-soc-report2013-h2-jp.pdf>(参照 2014-08-20)

[3] JPCERT/CC : 注意喚起「ウェブサイトの改ざん回避のために早急な対応を」，入手先<https://www.jpcert.or.jp/pr/2014/pr140003.html>(参照 2014-08-20)

[4] 株式会社 JNS:Web 改ざん検知/自動復旧システム isAdmin，入手先<http://www.jnsjp.com/isadmin.html>(参照 2013-09-25)

[5] 竹森敬祐，三宅優，中尾康二 : Web サーバリモート監視におけるホームページ改竄判定“ ，情報処理学会研究報告．CSEC，vol.2002，No.68，pp.27-32(オンライン)，入手先<http://ci.nii.ac.jp/naid/110002675589>(2002)

[6] Google : Safe Browsing，入手先<https://developers.google.com/safe-browsing/developers_guide_v3?hl=ja>(参照 2014-09-25)

[7] 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，情報処理学会論文誌，vol.51，No.1，pp.191-198(オンライン)，入手先<http://ci.nii.ac.jp/naid/110007970627/>(2010)