リダイレクトの構造的類似性に基づく悪性 Web ページ検知手法

芝原 俊樹　八木 毅　秋山 満昭　高田 雄太　矢田 健

NTT セキュアプラットフォーム研究所
180-8585 東京都武蔵野市緑町 3-9-11
{shibahara.toshiki, yagi.takeshi, akiyama.mitsuaki,
takata.yuta, yada.takeshi}@lab.ntt.co.jp

あらまし Drive-by download 攻撃に利用される悪性ページを特定するために，Web ページのコンテンツの悪性判定に教師あり機械学習を適用する手法が提案されている．しかし，攻撃に使用されるコンテンツは変化しやすいため，頻繁に識別器の再学習が必要であり，その度に大量の悪性な教師データに攻撃コードが含まれているか手動での確認が必要であった．そこで，本稿では，Drive-by download 攻撃で発生するリダイレクトの情報を自動的に収集することで再学習可能な手法を提案する．提案手法では，リダイレクトの構造的類似性に基づき悪性判定を実施する．2 年半のクローリング結果を用いた評価では，提案手法は，従来手法の正答率を 11%改善できた．

Detecting Malicious Web Pages based on Structural Similarity of Redirection Chains

Toshiki Shibahara　Takeshi Yagi　Mitsuaki Akiyama　Yuta Takata
Takeshi Yada

NTT Secure Platform Laboratories
3-9-11, Midori-cho, Musashino-shi, Tokyo 180-8585, Japan
{shibahara.toshiki, yagi.takeshi, akiyama.mitsuaki,
takata.yuta, yada.takeshi}@lab.ntt.co.jp

Abstract In order to detect malicious pages used in drive-by download attacks, the methods which apply supervised machine learning to evaluate maliciousness of content on web pages are proposed. However, these methods need manual inspections for confirming that malicious training data include exploit codes when classifiers require retraining because content on malicious pages is easy to change. In this paper, we propose a method that can retrain the classifier by using the information of redirection chains arising from drive-by download attacks, which can be identified automatically. This method evaluates maliciousness on the basis of structural similarity of redirection chains. Our results of experiments using 2.5 years data showed that the accuracy of our method was 11% higher than the accuracy of the previous methods.

1　はじめに

近年，攻撃者によって改ざんされた Web ページにアクセスしたユーザを，リダイレクトにより攻撃者の Web ページに誘導することでマルウェアに感染させる，drive-by download 攻撃が脅威となっている．この攻撃への対策として，

攻撃に使用される一連の悪性ページをブラックリスト化し，通信を遮断する対策が有効であると考えられていた．しかし，ブラックリストを用いた対策への回避策として，悪性ページで用いられるドメインが頻繁に変更されるようになったことが確認されている [1]．このような悪性ページをもブラックリスト化するためには，大量の Web ページを定常的に解析し，見逃しと誤検知を防ぎつつ悪性ページを発見する必要がある．

大量の Web ページを解析するために，Web ページのコンテンツから特徴量を抽出し，教師あり機械学習を適用する手法が提案されている [2, 3, 4]．しかし，悪性ページのコンテンツの特徴は，変化することが知られている [5]．このため，これらの手法では，識別器の性能が低下しやすいと考えられる．したがって，識別器の性能を維持するために，識別器の構築に利用する教師データとして大量の悪性・良性コンテンツを用意し，頻繁に再学習を実施しなくてはならない．しかし，悪性なコンテンツは，難読化などの処理で隠蔽されており，自動的に特定することが困難である．このため，手動で大量のコンテンツの悪性判定を実施する必要がある．つまり，従来手法では，実運用において教師データを頻繁に用意することが困難であり，その結果，識別器の性能を維持できない可能性がある．

そこで，本稿では，自動的に収集可能かつ攻撃者によって変更されにくい情報をもとに，再学習を実施できる悪性ページ検知手法を提案する．さらに，提案手法における識別器の性能を維持するために必要な，教師データの収集期間と，再学習頻度を明らかにする．Drive-by download 攻撃では自動的に発生する多段のリダイレクトが必須であり，この特徴は攻撃者によって変更されにくいと考えれる．このため，提案手法では，リダイレクトの構造的類似性に基づいて Web ページの悪性判定を実施する．リダイレクトの情報と Web ページごとの悪性・良性ラベルは，実ブラウザ環境を利用して悪性判定を実施する高対話型ハニークライアント [6] によって自動的に収集できるため，大量のデータに基づく定期的な再学習が実施可能となり，識別器の性能を維持できると考えられる．

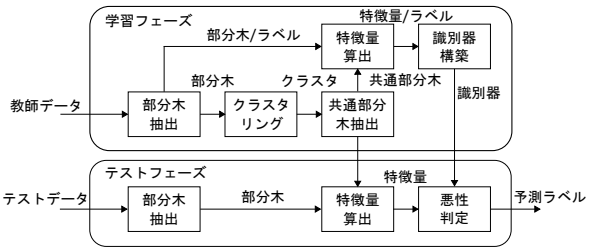

図 1: Web ページ悪性判定の処理手順

本稿の主な貢献は以下のとおりである．

- 教師データを自動的に収集可能であり，定期的に再学習を実施することで識別器の性能を維持可能な，リダイレクト構造の類似性に基づく悪性ページ検知手法を提案した．
- 提案手法における識別器の性能を維持するために必要な，教師データの収集期間と，再学習頻度を明らかにした．
- 2 年半の観測結果を用いた評価では，従来手法に対し識別器の性能を 82%から 93%に改善できることが明らかとなった．

2 提案の悪性ページ検知手法

従来手法の Prophiler [4] では，HTML，JavaScript，URL それぞれに対し悪性判定を実施する．HTML に関してはタグの出現回数や小さいエレメントの数，JavaScript に関しては関数の出現回数やスクリプトのエントロピー，URL に関してはドメイン名の長さやドメインに紐づく IP アドレス数などの特徴量を算出する．

これに対し，提案手法では，Web ページのリダイレクト構造に対し悪性判定を実施する．悪性・良性ページに出現する典型的なリダイレクト構造との類似度を特徴量として算出する．特徴量の算出に必要となるリダイレクトの情報として，リダイレクト元 URL，リダイレクト先 URL，リダイレクトの方法を収集する．リダイレクトの方法としては，HTTP のステータスコード 300 番台に加え，iframe タグによるリンクや script タグによるリンクなどを用いる．

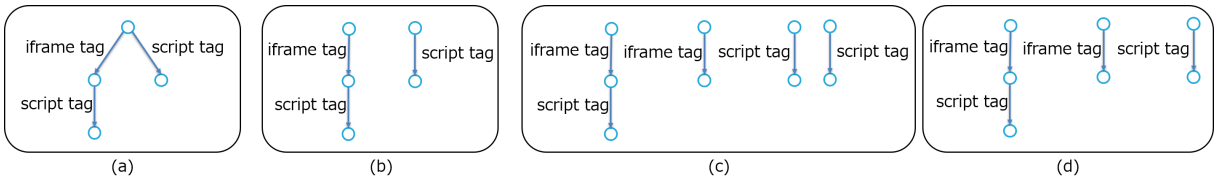

図 2: 部分木抽出手法．(a) リダイレクトの木構造．(b) 最初にアクセスした URL から終端の URL までの系列の抽出．(c) 抽出された系列から部分系列を抽出．(d) 重複を削除．

Web ページのリダイレクトは，URL をノード，リダイレクトをエッジとすると，木構造を持つデータとして表現できる．提案手法では，木構造から部分木を抽出し，部分木の一致度合いに基づいてリダイレクトの構造的類似性を算出することで悪性判定に用いる．具体的な処理手順を図 1 に示す．まず，学習フェーズでは，教師データを類似度に基づいてクラスタリングし，各クラスタから共通する部分木を抽出する．ここで抽出された部分木と各 Web ページのリダイレクトから抽出された部分木との類似度を並べ，特徴ベクトルとする．特徴ベクトルと教師ラベルに対し教師あり機械学習を適用し，識別器を作成する．次に，テストフェーズでは，学習フェーズと同様に特徴ベクトルを算出し，構築した識別器を用いて分類を実施する．以下，各ステップの詳細について述べる．

部分木抽出　リダイレクトの木構造（図 2-a）から部分木を抽出する．Drive-by download 攻撃では，改ざんされたページからマルウェアを配布するページまでの一連のリダイレクトが重要であると考えられる．そこで，リダイレクトの分岐ではなく，最初にアクセスした URL から各終端の URL までの一連のリダイレクトに着目する．

最初にアクセスした URL からすべての終端の URL までのリダイレクトの方法の系列を抽出し（図 2-b），それぞれの系列から部分系列を抽出する（図 2-c）．抽出された部分系列 t から重複するものを削除し，リダイレクトの部分木 $R = \{t_i\}$ とする（図 2-d）．この手法により，部分木抽出の計算量を削減しつつ，悪性判定に有効なリダイレクトを抽出できる．

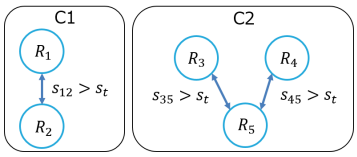

図 3: 部分木の類似度に基づくクラスタリング

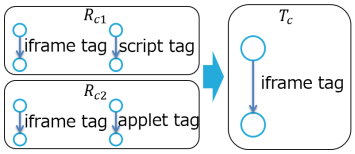

図 4: 共通部分木の抽出

クラスタリング　クラスタリングでは類似度として，Jaccard 係数を用いる．リダイレクト R_i と R_j の類似度を求める関数 $S(R_i, R_j)$ は，それぞれから抽出した部分木に共通する部分木数と，部分木数の種類数との割り合いで算出する．

$$S(R_i, R_j) = \frac{|R_i \cap R_j|}{|R_i \cup R_j|} \tag{1}$$

ここで，$|R|$ は部分木の数，$R_i \cap R_j$ は R_i と R_j の積集合，$R_i \cup R_j$ は R_i と R_j の和集合である．

式 1 で定義した類似度を用いてクラスタリングを行う．リダイレクト R とクラスタ $C = \{R_{ci}\}$ に属するリダイレクトとの類似度の最大値が，閾値 s_t より大きい場合クラスタ C に R を加える（図 3）．これを，各リダイレクトが 1 つのクラスタを構成している状態から，結合させるクラスタが存在しなくなるまで実施する．本稿では，経験的に $s_t = 0.9$ を用いた．

共通部分木抽出　複数のリダイレクトが属するクラスタ C において，すべてのリダイレクトに共通な部分木の集合 T_c を抽出する（図 4）．

$$T_c = \cap_{R_i \in C} R_i \tag{2}$$

特徴量算出　リダイレクト R から抽出された部分木と，各クラスタの共通する部分木 T_{ci} との類似度を並べ特徴ベクトル r とする．

$$r = [S(R, T_{c1}), S(R, T_{c2}), ..., S(R, T_{cn})] \quad (3)$$

識別器構築と悪性判定　学習フェーズでは，教師データから特徴ベクトルを算出し，教師あり機械学習を適用することで識別器を構築する．テストフェーズでは，テストデータから特徴ベクトルを算出し，構築した識別器を用いて悪性判定を行う．本稿では，機械学習手法として，非線形の識別が可能な SVM を適用し，カーネルには RBF カーネルを用いる．

3　実験

2 年半のクローリングにより収集したデータを用いて，悪性ページ検知における識別器の性能を評価した．まず，適切な教師データの収集期間を調査するために，収集期間と正答率の関係を評価した．次に，必要な再学習の頻度を調査するために，学習を実施してから経過した時間と正答率の関係を評価した．最後に，上記実験で明らかになった適切な教師データの収集期間と再学習頻度を想定した際の識別器の性能を評価した．

悪性ページのコンテンツの特徴が変化しないのであれば，Prophiler で採用している特徴量を用いた手法（以下，Prophiler 手法）で生成した識別器を継続的に使用することで Web ページの悪性判定を実施できると考えられる．そこで，Prophiler 手法と提案手法を実装し，識別器の性能を評価した．ただし，ブラックリストでの検知率を高めるという観点から，大量の Web ページを対象とした教師データを前提とする．このとき，全ての Web ページに対し手動で攻撃コードの確認を行い，Prophiler 手法の教師データを作成することは困難であると考えられる．そこで，今回は高対話型ハニークライアント [6] で自動的に検知した悪性ページのすべてのコンテンツを悪性とラベル付けを行い，Prophiler 手法の教師データを作成した．

3.1　データ収集方法

高対話型ハニークライアント [6] で公開ブラックリストや Alexa 等に掲載されている Web ページを巡回することでデータを収集した．この際，高対話型ハニークライアントで攻撃が検知された Web ページを悪性ページとし，公開ブラックリストに掲載されておらず攻撃を検知しなかった Web ページを良性ページとした．公開ブラックリストに掲載されており，高対話型ハニークライアントで攻撃を検知しなかった Web ページは，高対話型ハニークライアントのブラウザやプラグインのバージョンが原因で，攻撃が発動しなかった可能性があるため，評価データとして用いなかった．さらに，各手法の特徴量算出が不可能である HTTP リクエストとレスポンスの数が異なる Web ページ，リダイレクト情報の取得に失敗したページは評価データから除外した．悪性・良性ページの収集した期間とページ数を図 5 に示す．各期間に悪性･良性ページが少なくとも 1 つ以上存在するようにするため，本実験では 3ヶ月間ごとに期間を区切り評価を行った．なお，本稿では，2012 年 7 月から 2012 年 9 月までに収集したデータを 2012/7-9 のデータと記述する．

3.2　学習に必要な教師データの収集期間

学習に必要な教師データの収集期間を調査するため，評価に用いるテストデータの直近 3ヶ月間から 27ヶ月間まで 3ヶ月間ずつ収集期間を増やして正答率を評価した．テストデータは 2014/10-12 のデータを用いた．正答率は，テストデータ数に対する悪性・良性を正しく判定したページ数の割合である．

提案手法では，教師データの収集期間が 6ヶ月間から 9ヶ月間のときに正答率が 0.998 で最高となり，Prophiler 手法では，6ヶ月間から 12ヶ月間のときに正答率が 0.938 で最高となった（図 6）．提案手法，Prophiler 手法ともに教師データとして，6ヶ月間以上のデータが必要と考えられる．

教師データに長期間のデータを使用した場合，正答率の低下が観察された．テストデータより

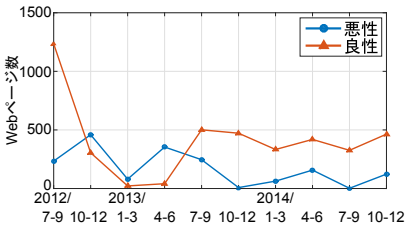

図 5: 悪性・良性ページの収集期間とページ数

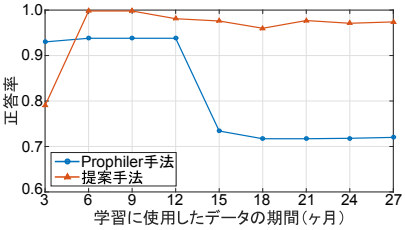

図 6: 教師データの使用期間の正答率への影響

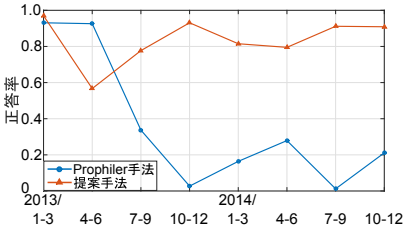

図 7: 時間経過に伴う正答率の変化

12ヶ月以上過去のデータにおけるリダイレクトやコンテンツの傾向がテストデータのそれらと傾向が異なっていたため，学習に悪影響を与えたと考えられる．

本評価の結果から，今後の実験では，本評価で最高の正答率を実現できた，6ヶ月間を教師データを収集する期間として採用する．

3.3 時間経過に伴う識別器の性能の変化

構築した識別器が有効に機能する期間を調査するために，2012/7-12 の期間に収集されたデータを用いて識別器を構築し，その後の 2013/1-2014/12 のデータに対する正答率を 3ヶ月間ごとに評価した．

正答率は，提案手法，Prophiler 手法ともに教師データを収集した期間の直後の 2013/1-3 が最高となった（図 7）．提案手法は，大きく正答率を落とす期間もあるが，2013/7 以降正答率が 0.5 を下回る Prophiler 手法よりは学習からの時間経過に強いことが確認できた．しかし，3ヶ月経過以降は正答率が 0.9 を下回ることが多いため，3ヶ月ごとに再学習を実施する必要がある．

提案手法において，正答率が低下している 2013/4-6 において誤識別をしていたページは，最初にアクセスした URL に攻撃コードが書かれているページであった．本実験では，悪性ページの収集に公開ブラックリストに掲載されている URL を利用したため，最初にアクセスした URL に攻撃コードやマルウェアを含むものが多かったと考えられる．このようなページでは，script タグによるリンクなど多くのページに共通して出現するリダイレクトのみ発生していたため，良性と判定されたと考えられる．

また，正答率の変化がリダイレクト構造やコンテンツの変化による影響であることを確認するために，識別に用いているリダイレクト構造やコンテンツが時間経過に伴って変化するかを調査した．リダイレクト構造の変化に関する評価では，リダイレクトの方法ごとの部分木中に出現する回数の平均を 3ヶ月間ごとに算出し，変化量を調査した．コンテンツの変化に関する調査では，Prophiler 手法で利用されている各特徴量の z スコアの 3ヶ月間ごとの平均を算出し，変化量を調査した．z スコアとは全データの平均が 0，分散が 1 となるように標準化したときの各データの値である．

提案手法における重要な特徴量である悪性ページのリダイレクトは，2013/10 前後で異なる傾向が観察された（図 8）．2013/10 以前では，ステータスコード 302 によるリダイレクトが多く出現し，2013/10 以降では iframe タグが多く出現していた．リダイレクトの傾向は変化したが，iframe タグは 2013/10 以前から継続的に悪性ページで用いられており，良性ページではあまり用いられていないため，提案手法では，傾向の変化に対して大きな正答率の低下が発生しなかったと考えられる．

Prophiler 手法における重要な特徴量であるコンテンツの傾向を解析するために，Prophiler 手法で使用されている HTML と JavaScript の特徴量の中で，悪性ページで z スコアが高かったものを選定し，図 9 と図 10 に悪性ページと良性ページにおける選定した特徴量の z スコアの変化を示す．ここで，HTML の特徴量のうち，シェルコードの可能性が高いコードとは，タグ間のデータ長が 128 より大きくスペースの割合が 5%未満のものであり，exploit 可能なオブジェ

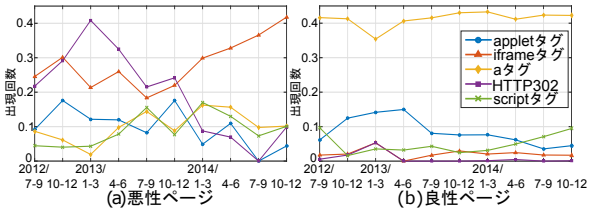

図 8: リダイレクト構造の変化

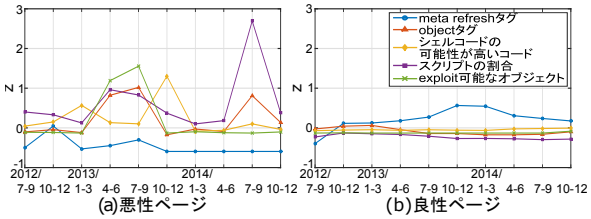

図 9: HTML の変化

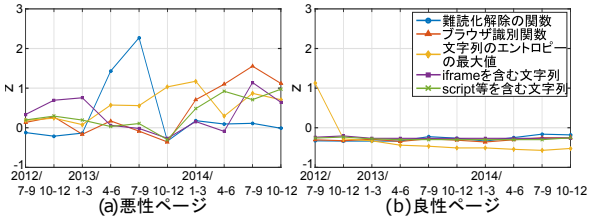

図 10: JavaScript の変化

クトとは，classid が exploit 可能な ActiveX コントロールとして知られているものである．図 9 と図 10 においてｚスコアが高い特徴量の組み合わせから，悪性ページにおけるコンテンツの傾向は 3ヶ月間から 6ヶ月間で変化していることが分かった．一方，良性ページにおけるコンテンツの傾向は，ほぼ変化しなかった．JavaScript における難読化解除の関数など，悪性ページで顕著に出現する特徴量は良性ページではほとんど出現せず，これらの特徴量に着目すれば悪性ページと良性ページを識別可能であると考えられる．ただし，悪性ページで顕著に出現する特徴量は時間経過とともに大きく変化するため，学習から時間が経過すると Prophiler 手法の正答率が大きく低下したと考えられる．例えば，JavaScript の特徴量のうち，文字列のエントロピーの最大値が，2012/7-9 では良性ページで高いのに対し，2013/4 以降悪性ページで高くなっているため，誤識別の原因になったと考えられる．

表 1: 2013/1-2014/12 の識別器の性能

手法	正答率	検出率	精度
提案手法	0.931	0.728	0.733
Prophiler 手法	0.820	0.942	0.630

3.4 再学習を実施した際の識別器の性能

3.2，3.3 節の評価で得られた，最適な教師データ収集期間と再学習頻度を運用に適用することを想定し，3ヶ月ごとに直近 6ヶ月間のデータを利用して再学習を実施した際の正答率，検出率，精度を評価した．ここで，検出率は悪性ページのうち悪性と予測したページの割合，精度は悪性と予測したページのうち悪性なページの割合である．

2013/1-2014/12 の平均では，提案手法は Prophiler 手法に比べ正答率が 11%，精度が 10% 高い結果となった（表 1）．一方，検出率においては，Prophiler 手法が提案手法に比べ 22%高い結果となった．ここで，提案手法における検出率の低下を調査するために，各評価項目の時系列変動を調査した（図 11）．この調査の結果，提案手法で検知できなかった悪性ページは，3.3 節の実験で誤識別していた悪性ページと同様に最初にアクセスした URL に攻撃コードが書かれているページであった．一方，悪性と誤検知された良性ページでは，iframe タグにより複数の URL にリダイレクトが発生していた．このようなページは，iframe タグにより複数の攻撃コードを取得する悪性ページとの類似度が高くなり，悪性と予測され，精度低下の原因となったと考えられる．

4 考察

4.1 大量の Web ページに対する悪性判定

本稿では，高対話型ハニークライアントで収集されたリダイレクト情報に対し提案手法を適用したが，リダイレクト情報はブラウザエミュレータを利用したおとりのクライアントシステムである低対話型ハニークライアントでも収集が可能である．低対話型ハニークライアントは，

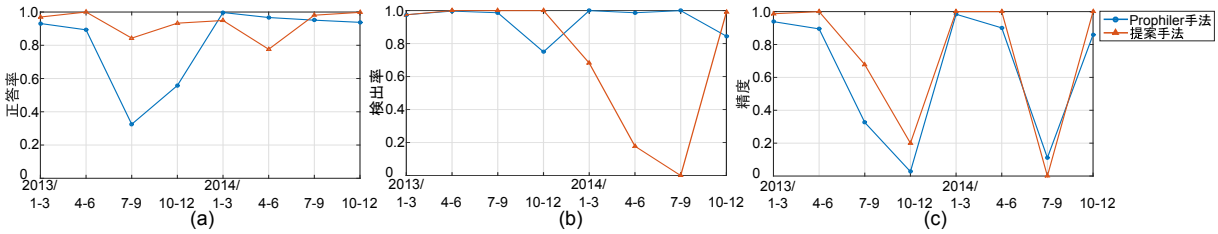

図 11: 再学習を定期的に実施した際の識別器の性能

高対話型ハニークライアントより高速に Web ページの解析が可能である．このため，低対話型ハニークライアントを用いることで，大量の Web ページを評価して悪性ページを検知することが可能となる．

4.2 リダイレクト情報の有効性

評価実験において，提案手法の正答率が Prophiler 手法を上回ったことから，リダイレクトの情報が悪性ページの検知に有効なことが示された．しかし，提案手法の正答率が低下した期間も存在し，そのとき誤識別した Web ページでは悪性・良性ページ共通に使用されるリダイレクトが発生していた．このような Web ページに対しては，Prophiler 手法の正答率の方が高かったため，悪性ページと良性ページではコンテンツが異なると考えられる．そこで，今後自動化が可能な範囲でコンテンツの情報も併用した手法への拡張を検討する．

4.3 提案手法の回避

提案手法を回避するための手法として，悪性ページにおいて良性ページで頻繁に利用されるリダイレクトを発生させる手法が考えられる．このような悪性ページは，良性ページが多く属するクラスタの共通部分木との類似度が増加し，悪性ページが多く属するクラスタの共通部分木との類似度が低下する．その結果，良性と判定されてしまう可能性がある．

この手法への対策として，特徴量を算出する際の類似度を $S(R,T) = \frac{|R \cap T|}{|T|}$ とすることで，悪性ページが多く属するクラスタの共通部分木との類似度低下を防ぐことが可能と考えられる．しかし，誤検知が増加する可能性があるため，検証が必要である．

4.4 評価用教師データのラベル付け

Prophiler 手法の評価では，手動での教師データに悪性コードが含まれているかの確認が困難であったため，高対話型ハニークライアントの識別結果をすべてのコンテンツに反映した．しかし，悪性と判定された Web ページすべてのコンテンツに悪性なコードが含まれているとは限らない．このため，本稿の実験では，誤ったラベル付けがされたコンテンツが学習に利用されることになる．この影響により，再学習を実施した際の正答率が，提案論文 [4] で報告されている 0.89 から 0.82 に低下したと考えられる．

5 関連研究

5.1 悪性ページ検知手法

さまざまな観点から特徴量を算出し，教師あり機械学習を適用して Web ページの悪性判定を実施する手法が提案されている．ZOZZLE [2] は，JavaScript の抽象構文木を構築し，for 文や while 文などの繰り返し処理や，文字列のノードを抽出する．抽出したノードの悪性な JavaScript と良性な JavaScript における出現率に有意差があるか χ^2 検定を用いて判定し，有意差があるノードを選定する．選定されたノードが，評価対象の JavaScript から構築された抽象構文木に出現するかを特徴量として用いる．

Revolver [3] は，JavaScript の抽象構文木におけるノードの種類の系列を抽出する．このノード系列の類似度を，一致するノードの割合で算

出し，類似した JavaScript のラベルに基づいて分類を行う．

Prophiler [4] は，複数のコンテンツから特徴量を抽出するため，他の手法に比べ見逃しが少ない手法である．ブラックリストを用いた対策では，見逃しと誤検知を削減する必要があるが，両者を比較するとマルウェア感染の被害を防ぐために，見逃しの削減の方が重要である．この観点から，Prophiler は他の手法と比較し優れていると考えられるため，提案手法の比較対象として選定した．

これらの手法は，Web ページごとに悪性判定を実施する高対話型ハニークライアントの悪性判定結果を教師データの悪性･良性ラベルとして適用することができない．一方，Web ページごとに悪性判定を実施するため，高対話型ハニークライアントの悪性判定結果を利用できる手法も提案されている．SpiderWeb [7] は，まず最終的なリダイレクト先 URL が同一の Web ページごとの集合を作成する．その後，各集合ごとにリダイレクト長などを特徴量として抽出し，悪性判定を実施する．この手法は，悪性判定において，各 Web ページのリダイレクト情報同士の比較が必要なため，大量の Web ページの悪性判定を実施する際に処理負荷が高く，適用が困難である．

5.2 木構造データ分類手法

木構造をもつデータの扱いは，グラフマイニングと呼ばれる分野で研究されており，木カーネルを用いてデータの分類を実施する手法が提案されている [8]．木カーネルとは，部分木の一致度合いをカーネル関数の値として用いる手法である．この手法により，構造に基づいた分類が可能となるが，木カーネルの計算量が大きく，大量のデータに対しては類似度を算出する回数が膨大になるため，適用が困難な手法である．

6 まとめ

本稿では，リダイレクトの構造的類似性に着目し，経年劣化しにくい Web ページの悪性判定手法を提案した．さらに，提案手法における識別器の性能を維持するために必要な，教師データの収集期間と，再学習頻度を明らかにした．本手法は，高対話型ハニークライアントによって自動的に生成された教師データを用いて再学習が可能であり，定期的に再学習することで継続的に大量の Web ページを高精度に評価し，悪性 Web ページを検知することができる．

参考文献

[1] C. Grier *et al.*, “Manufacturing compromise: the emergence of exploit-as-a-service,” Proceedings of the 2012 ACM conference on Computer and Communications Security, pp.821–832, 2012.

[2] C. Curtsinger *et al.*, “Zozzle: Fast and precise in-browser javascript malware detection.,” USENIX Security Symposium, pp.33–48, 2011.

[3] A. Kapravelos *et al.*, “Revolver: An automated approach to the detection of evasive web-based malware.,” USENIX Security Symposium, pp.637–652, 2013.

[4] D. Canali *et al.*, “Prophiler: a fast filter for the large-scale detection of malicious web pages,” Proceedings of the 20th international conference on World Wide Web, pp.197–206, 2011.

[5] 高田ら, “ドライブバイダウンロード攻撃に使用される悪性な javascript の実態調査,” 信学技報, vol.113, no.502, pp.59–64, 2014.

[6] M. Akiyama *et al.*, “Design and implementation of high interaction client honeypot for drive-by-download attacks,” IEICE Transactions on Communications, vol.93, no.5, pp.1131–1139, 2010.

[7] G. Stringhini *et al.*, “Shady paths: Leveraging surfing crowds to detect malicious web pages,” Proceedings of the 2013 ACM SIGSAC conference on Computer and Communications Security, pp.133–144, 2013.

[8] T. Gärtner, “A survey of kernels for structured data,” ACM SIGKDD Explorations Newsletter, vol.5, no.1, pp.49–58, 2003.