悪性 Web サイト探索のための優先巡回順序の選定法

千葉 大紀 †　　森 達哉 ‡　　後藤 滋樹 †

† 早稲田大学 基幹理工学研究科 情報理工学専攻
169-8555 東京都新宿区大久保 3-4-1
{chiba,goto}@goto.info.waseda.ac.jp

‡NTT ネットワーク基盤技術研究所
180-8585 東京都武蔵野市緑町 3-9-11
mori.tatsuya@lab.ntt.co.jp

あらまし Web ブラウザを攻撃対象とする悪性サイトが増加している．この脅威に対し，Web サイトを巡回し，悪性サイトを発見するためのクライアント型ハニーポット技術が研究開発され，日々攻撃情報が収集されている．しかし，クライアント型ハニーポットで網羅的に巡回するには多大なリソースを必要とするため，より効率的に Web サイトを巡回するべきである．そこで本研究では，悪性サイトの IP アドレス，WHOIS，FQDN 文字列の情報から統計的特徴ベクトルを作成し，教師あり機械学習を適用することで，最適な巡回リストを生成する技術を提案する．実データを用いた評価の結果，提案手法はより多くの悪性サイトを含む巡回リストを生成可能であることがわかった．

Deciding priority crawling in searching for malicious websites

Daiki Chiba†　　Tatsuya Mori‡　　Shigeki Goto†

†Graduate School of Fundamental Science and Engineering, Waseda University
3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555, JAPAN
{chiba,goto}@goto.info.waseda.ac.jp

‡NTT Network Technology Laboratories
3-9-11 Midori-cho, Musashino-shi, Tokyo 180-8585, JAPAN
mori.tatsuya@lab.ntt.co.jp

Abstract Malicious websites that attack web browsers have become one of the most serious threats. To collect attack information for protective use, client honeypots have been developed. However, the attack information that can be collected by a honeypot is limited. This paper proposes a new method for collecting attacks more effectively. Our method makes use of IP addresses, WHOIS, and FQDN strings and it constructs a statistical feature vector. We apply a supervised machine learning to generate a URL list with crawling priority. We validate that our new method can generate an effective URL list for client honeypots.

1　はじめに

Web 経由でのマルウェア感染事例が増加している．特に Web ブラウザやそのプラグインの脆弱性を攻撃対象とする Drive-by download 攻撃 [1] が深刻化している．この脅威に対し，Web クライアント型ハニーポットを利用した対策が研究開発されている [1, 2]．Web クライアント型ハニーポットとは，Web 空間を巡回し，悪性サイトを発見するためのシステムであり，エミュレータを用いて構成される低対話型と，実際の OS やブラウザによって構成される高対話型の 2 種類に大別できる [1]．高対話型ハニーポットは，実際のシステムを利用することから，より多くの攻撃情報を収集でき，最新の悪性サイトの調査に利用される．例えば文献 [2] では，高対話型ハニーポットを用いた，悪性サイトの収集・分析・対策手段の研究開発の成果が報告されている．

しかし，高対話型の Web クライアント型ハニーポットには，コストに起因する 2 つの課題が存在する．1 つは，巡回すべき悪性サイトの選定である．攻撃者は悪性サイトを多く展開しており，さらにその URL は短い期間で遷移している [1]．この状況の中で，すべての悪性サイトをハニーポットによって発見することは難しく，より効率的に調査すべき悪性サイトを選定する必要がある．もう 1 つの課題は，サイトの再巡回である．Web サイトはハニーポットによって一度巡回・検査されるだけでは不十分である [3]．なぜなら，巡回された時点では悪性サイトではなくとも，その後悪性サイトと変化する可能性があるためである．Web 空間は日々拡大する一方，

ハニーポットの単位時間あたりの巡回可能サイト数には限界がある．したがって，より悪性の可能性が高い Web サイトを抽出し，再巡回の効率を高める必要がある．

上記の 2 つの課題を解決するために，本研究では，悪性サイトの IP アドレス，WHOIS，FQDN 文字列の情報から統計的特徴ベクトルを作成し，教師あり機械学習を適用することで，Web クライアント型ハニーポットに最適な巡回 URL リストを提供する技術を提案する．中心となるアイデアは，良性サイトと悪性サイトを識別し得る特徴を抽出し，より悪性の可能性の高いサイトに高い優先巡回順序を付与することである．本研究の貢献は，既存の悪性サイト検知手法や，Web クライアント型ハニーポットを置き換えるものではなく，より効率的に悪性サイトを発見するための手段を提供することである．

本論文は以下の章により構成される．2 章で，関連研究を概観し，3 章で，本研究の提案手法を説明する．4 章で，提案手法で利用する各特徴量の有効性を評価し，5 章で，提案手法の性能評価と考察を行う．最後に 6 章で本研究をまとめる．

2　関連研究

悪性 URL・悪性ドメイン名を検知する代表的な研究に，[4, 5, 6] がある．文献 [4] は，既知の悪性と良性 URL に含まれる文字列，ドメイン登録情報やその他の特徴から，入力 URL を悪性か良性に分類する手法を提案している．文献 [5] では，既知の悪性ドメイン名の登録情報や DNS 情報を用いて，未知の悪性ドメインの予測を行なっている．文献 [6] は，ボットネットやスパム（迷惑メール）で利用されるランダム性の高い悪性ドメイン名の検知を試みている．上記 3 つの研究は，我々の手法とは以下の 3 点で異なる．すなわち，我々の手法は (1) 良性と悪性の二値分類ではなく，連続値で悪性度を算出する．(2) IP アドレス自体の構造的特徴を活かした特徴抽出 [7] を行う．(3) 良性と悪性の両方を分析することで得られる識別能力の高い特徴を利用している．

検索エンジンを活用して悪性サイト探索を行う研究が行われている [1, 8]．文献 [1] では，既知の悪性 URL の近隣に存在する URL を検索エンジンで抽出することで，未知の悪性 URL を発見する手法が提案されている．この研究は悪性 URL のみを対象としているのに対し，我々の手法は，(1) 良性と悪性の両方を分析する点，(2) 検索エンジンを使わない点，で異なる．文献 [8] は，既知の悪性サイトのコンテンツ類似度，ドメイン登録情報，その他の特徴を用いることで，悪性サイトが含まれる可能性の高い検索クエリを作成し，効率的に悪性サイトを発見する手法を提案している．この研究は我々の研究目的及びアプローチに最も近いが，以下の 3 点で我々の手法と異なる．すなわち，我々の手法は，(1) コンテンツを利用せず，IP アドレス・ドメイン登録情報・FQDN 文字列のみを用いる軽量な手法である点，(2) 出力が検索クエリではなく，優先巡回順序付きの URL リストである点，(3) 検索エンジンのクローラ向けの手法ではなく，の Web クライアント型ハニーポット向けの手法である点，で異なる．

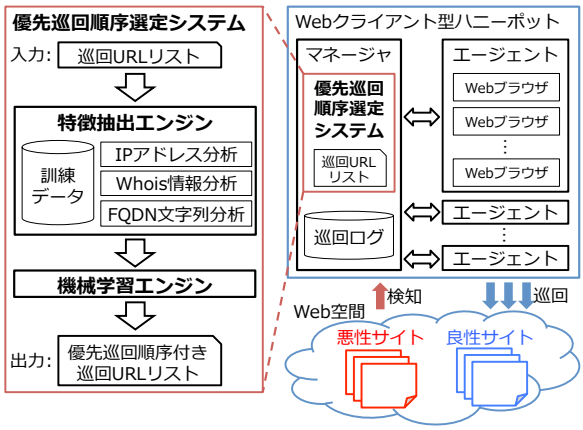

図 1: 優先巡回順序選定システム．

ハニーポット自体の性能向上を行う研究も行われている [9]．文献 [9] では，高対話型の Web クライアント型ハニーポットの巡回性能を向上させるために，マルチ OS 及びマルチプロセスによるマルチエージェントを実現する手法が提案されている．この手法によりハニーポットの巡回性能は大幅に向上するが，我々の研究成果を使えばさらに性能を高めることができる．このことを 5 章の評価実験で示す．

3　提案手法

3.1　概要

本章では，悪性 Web サイト探索のための優先巡回順序の選定法について説明する．本手法を実現する，優先巡回順序選定システムの概要図を図 1 に示す．本手法は，以下の 4 つの手順で優先巡回順序を決定する．

手順 1: 特徴抽出エンジンにて，既知の良性と悪性サイト（訓練データ）から IP アドレス，WHOIS，FQDN 文字列の情報を用いて統計的特徴ベクトルを作成する．なお，特徴抽出エンジンについては 3.2 章で詳細を説明する．

手順 2: 機械学習エンジンにて，上記で作成した特徴ベクトルをもとに教師あり機械学習を適用し，訓練モデルを作成する．なお，機械学習エンジンについては 3.3 章で解説する．

手順 3: 巡回 URL リスト（テストデータ）を入力として与え，特徴抽出エンジンにて，それぞれの URL から手順 1 と同様に特徴ベクトルを抽出する，

手順 4: 機械学習エンジンにて，手順 2 で作成した訓練モデルをもとに，手順 3 で作成した巡回 URL リストの特徴ベクトルから，それぞれの URL の悪性度を算出する．算出した悪性度に基づき，優先巡回順序を決定する．その結果を優先巡回順序付き巡回 URL リストとして出力する．

図 1 に示した本システムは，Web クライアント型ハニーポットの一部として動作し，本システムが出力する優先巡回順序付きの巡回 URL リストを利用して，Web 空間を巡回することを想定している．なお，今回はクライアント型ハニーポットのアーキテクチャとして，文献 [9] で提案されているものを採用する．すなわち，ハニーポットはマネージャと複数のエージェントによって構成され，マネージャは，巡回 URL リスト，巡回ログを管理し，各エージェントの動作を制御する．一方，各エージェントは，複数の Web ブラウザのプロセスを用いて巡回 URL リストに含まれる URL を巡回する．

3.2 特徴抽出エンジン

良性サイトと悪性サイトを識別し得る特徴を抽出し，統計的特徴ベクトルを生成する．本研究では，以下の 3 つの分析手法（IP アドレス分析，WHOIS 情報分析，FQDN 文字列分析）を用いた特徴抽出を提案する．

3.2.1 IP アドレス分析

Web サイトの IP アドレスからの特徴抽出手法を説明する．中心となるアイデアは，IP アドレス空間の偏りを利用することである．悪性な活動（ボットネット，迷惑メール送信元）に利用される IP アドレスは，あるネットワークブロックに空間的に偏ることが明らかになっている [10, 11]．我々は，以前の研究 [7] で，悪性 Web サイトに利用される IP アドレスにも上記の性質があることを確認し，この性質を利用した特徴抽出手法を提案した．本研究では，上記研究のうち，最も良い識別精度が得られる手法を利用する．具体的には，特徴抽出を行う Web サイトの IP アドレスの第 1〜3 オクテットの数値および，IP アドレスの第 1・2 オクテットの組合せ，第 1・2・3 オクテットの組合せから特徴ベクトルのビット列 $\{b_0, \cdots, b_{1279}\}$ を定義する．すなわち，それぞれの変数 b_k を，以下の式で定義する．

$$\begin{cases} b_k = 1 \quad (k \text{ in } \bigcup_{n=1}^{3} \{2^8 \cdot (n-1) + X_n\}) \\ b_k = 1 \quad (k \text{ in } \bigcup_{m=3}^{4} \{2^8 \cdot m + (\sum_{n=1}^{m-1} X_n) \bmod 2^8\}) \\ b_k = 0 \quad (\text{otherwise}). \end{cases}$$

ここで，X_n は特徴抽出を行う IP アドレスの第 n $(1 \leq n \leq 3)$ オクテットの 10 進表記である．

3.2.2 WHOIS 情報分析

Web サイトのドメイン名から得られる WHOIS 情報を利用した特徴抽出手法を説明する．本手法では，WHOIS 情報のうちドメイン登録日を利用して，登録日の新しさに着目した特徴抽出を行う．具体的には，ドメインが登録されてからの経過日数（登録期間）を特徴ベクトルの要素 W として以下の式で定義する．

$$W = d_n - d$$

ここで，d_n は現在の日時，d は特徴抽出を行う Web サイトのドメイン登録日とする．この手法は，新しいドメイン登録日を持つ Web サイトの方がより悪性度が高いという仮定に基づいており，W が小さいほど，悪性度が高くなるように特徴抽出を行う．なお，ドメイン登録日と良性・悪性サイトの関係は，以下の 4.3 章で検証する．

3.2.3 FQDN 文字列分析

Web サイトの FQDN 文字列の長さ，エントロピー，n-gram から得られる情報を利用した特徴抽出手法を順番に説明する．

FQDN 文字列の長さ FQDN 文字列の長さを特徴として利用する．以下の 4.4.1 章で，良性と悪性 FQDN 文字列の長さを分析したところ，その分布に差があることが確認できた．この特徴を用いて，特徴ベクトルの要素 L を以下の式で定義する．

$$L = (FQDN \text{ 文字列の長さ})$$

FQDN 文字列のエントロピー FQDN 文字列のエントロピーを特徴として利用する．基本的なアイデアは，悪性サイトに使われる FQDN 文字列はランダム性が高い文字列の組合せになることが多く，エントロピーが高くなるという性質を利用することである．以下の 4.4.2 章では，実データを用いて，良性と悪性 FQDN 文字列のエントロピーを比較し，この性質を示す．ここで，FQDN 文字列のエントロピーを利用した特徴ベクトルの要素 E を，n 文字の FQDN 文字列 $X = \{x_1, x_2, \cdots, x_n\}$ のエントロピーを用いて，以下の式で定義する．

$$E = -\sum_{i=1}^{n} p(x_i) \log p(x_i)$$

ここで，$p(x_i)$ は，FQDN 文字列 X においてある文字が出現した経験確率である．

FQDN 文字列の n-gram FQDN 文字列から n-gram $(n = 2)$ を抽出し，特徴として利用する．基本的なアイデアは上節のエントロピーと同様で，悪性 FQDN 文字列の高いランダム性を利用することである．具体的には，FQDN 文字列から 2 文字の連続した文字列を切り出す．ここで，FQDN に存在する文字として，アルファベット（26 文字），数字（10 文字），記号（ドット，ハイフン）がある．これらの 2 文字の組合せのうち，数字または記号が少なくとも 1 文字含まれるもの（例 '1a', 'e-'）のみを特徴として抽出する．下記 4.4.3 章では，数字または

記号を含む文字列の出現頻度を分析することで，本手法の有効性を実証する．ここで，各 FQDN 文字列の n-gram 文字列を利用した特徴ベクトルをビット列 $\{g_{'-0'}, \cdots, g_k, \cdots, g_{'z9'}\}$ として定義し，それぞれの変数 g_k を以下の式で定義する．

$$g_k = N_k$$

ここで N_k は，n-gram 文字列 k の各 FQDN 文字列における出現頻度である．

n-gram における例外処理　上記の n-gram を用いた特徴抽出手法における例外処理の必要性を述べる．悪性 FQDN にはランダム性が高い文字列が含まれる可能性が高い．一方で，良性 FQDN の中にも，商品名・会社名・コンテンツ識別子の都合により，ランダム性の高い文字列が使われることがある．この状況で，n-gram 文字列の特徴ベクトルを利用すると，適切に評価できない良性 FQDN が存在する．そこで，特徴抽出エンジンでは，n-gram の特徴を利用しないドメイン名（例外ドメイン名）をあらかじめ定義することでこの問題を回避する．この例外ドメイン名には，悪性サイトの可能性が極めて小さいドメイン名を登録する．今回の研究では，例外ドメイン名として 20 個のドメイン名（例 大手検索業者のサービスや大手ソーシャルアプリケーション）を選択した．以下，5 章では，例外ドメイン名を用いた例外処理を行う場合と行わない場合の性能を比較する．

3.3　機械学習エンジン

3.2 章の特徴抽出エンジンにて抽出された特徴ベクトルを統合し，教師あり機械学習手法を適用する．本研究では，機械学習手法としてサポートベクターマシン (SVM) [12] を選択する．SVM を選択した理由は，関連研究 [4, 7] において高い精度で悪性サイトを検知した報告があるだけでなく，これまで多種多様な課題に対して適用され，優れた識別能力があることが示されている [12] ためである．

まず，SVM を用いた訓練モデル生成手法（手順 2）について説明する．SVM は，訓練データの特徴ベクトルを入力し，そのデータを高次元に写像した上で，データを識別するための超平面を構築する．入力データを高次元空間に写像するためのカーネル関数としては，代表的なガウスカーネル $e^{-\gamma||\mathbf{x}-\mathbf{y}||}$ を採用する．ここで，γ はパラメータであり，経験的に決定する．また，超平面を構築する際には，超平面とそれぞれの訓練データとの距離（マージン）を最大化するアプローチ（マージン最大化）をとることで，最適な超平面を構築する．超平面構築の際にはパラメータ C を経験的に決定し，マージン最大化条件と境界面によって生じるエラーに関するトレードオフを適切に制御する必要がある．高次元への写像および超平面構築の際に必要なパラメータの組合せ (γ, C) は，グリッド探索により最適なものを選択する．

次に，SVM による巡回 URL（テストデータ）の悪性度算出および優先巡回順序の決定方法（手順 4）について説明する．それぞれのテストデータは手順 2 で得た訓練モデル（最適な超平面）を参照し，超平面によって定まる識別境界により識別される．ここで，一般的な SVM では，良性と悪性の二値分類の結果しか得られない．しかし，今回は文献 [13] で提案されている手法を応用し，超平面からの距離を近似することにより，悪性度の確率を連続値で算出する．算出した悪性度をもとに，巡回 URL を降順（悪性度が高い順序）に並び替えることで，優先巡回順序付きの巡回 URL リストを得る．すなわち，このリストは悪性度が高いと推定される URL から順番に並んだ URL リストとなり，このリストを用いることで，ハニーポットが効率的に巡回を行うことが可能となる．

4　各特徴量の有効性

本章では，提案手法で利用するそれぞれの特徴が良性と悪性を判別するために有効であることを示すために，実データを用いた検証を行う．

4.1　データセット

実証実験のために利用した各データセットについて説明する．提案手法では，事前に良性か悪性が判明している Web サイトの情報（訓練データ）を用いた教師あり機械学習を行う．本研究で利用した訓練データセットの内訳を表 1 に示す．良性訓練データとして，Alexa 社が提供している Top sites [14] に掲載されている Web サイトのうち上位 10,000 件の FQDN（重複なし）を利用した．Alexa Top sites は，Web サイトの平均日別訪問者数および過去 1ヶ月間のページビュー数によって算出されている Web サイトランキングであり，上位に掲載されているサイトは良性である可能性が高い Web サイトとみなすことができる．一方，悪性訓練データとして，公開の悪性サイトブラックリストである Malware Domain List (MDL) [15] のうち，35,438 件の FQDN（重複なし）を利用する．なお，良性と悪性の両訓練データは，2011 年 4 月 30 日現在までに収集したものを利用している．

次に，提案手法を評価するために作成したテストデータセットについて説明する．本手法を用いることで，良性と悪性サイトが混在する Web 空間の中から，未知の悪性サイトに高い優先巡回順序を付与できることが望ましい．そこで今回は，実際に閲覧されている良性サイトと，評価時点で未知であった悪性サイトが混在する巡回リストをテストデータとして作成する．このテストデータは，提案手法の有効性を示すための妥当なデータとして，実際の Web 空間に近い状況を再現することを意図している．テストデータセットの内訳を表 2 に示す．良性テストデータは，あるネットワークの 2 週間のトラヒックデータを用いて収集された 96,597 件の FQDN（重複なし）である．この良性テストデータに悪性サイト

表 1: 訓練データセット.

データ	収集期間	FQDN 数
良性訓練	2011/4/30	10,000
悪性訓練	2009/1/1〜2011/4/30	35,438
合計		45,438

表 2: テストデータセット.

データ	収集期間	FQDN 数
良性テスト	2011/5/1〜2011/5/14	96,567
悪性テスト	2011/5/1〜2012/4/18	10,561
合計		107,128

が混入している可能性を排除するため，Google Safe Browsing API [16] を用いたチェックを行い，悪性サイトの可能性がある 2,515 件の FQDN を除外した．一方，悪性テストデータとしては，訓練データにも利用した Malware Domain List (MDL) [15] から 10,561 件の FQDN（重複なし）を利用する．ただし，悪性テストデータは，2011 年 5 月 1 日〜2012 月 4 月 18 日の 353 日間に新たに登場した悪性サイトを利用している．なお，悪性テストデータからは，既存のブラックリストで防御可能な悪性 FQDN および IP アドレスをすべて除去している．したがって，このテストデータセットを用いることで未知の悪性サイトに対する提案手法の有効性を適切に評価可能である．

4.2 IP アドレスの局所性

訓練データセット（表 1）に含まれる Web サイトの IP アドレスの特徴を調査した．今回は良性と悪性サイトそれぞれの IP アドレスをヒルベルト曲線に基づく 2 次元グラフ上に配置した．ヒルベルト曲線とは，再帰的に定義される空間充填曲線のうちの 1 つであり，この曲線を用いることで，IP アドレスの近接性を維持したまま IPv4 アドレス空間を 2 次元グラフとして視覚化できる [7, 10, 17, 18]．図 2 は，あるネットワークアドレスブロック x.0.0.0/14 の IP アドレスの配置をヒルベルト曲線によって視覚化したものである．図 2 の点線部から，悪性サイトに使われている IP アドレスが，あるネットワークアドレスブロックに偏っていることが視覚的に確認できる．3.2.1 章で示した特徴抽出エンジン（IP アドレス分析）では，この特徴を活用した特徴ベクトル抽出を行い，良性と悪性サイトの識別を行う．

4.3 ドメイン登録日の比較

訓練データセット（表 1）に含まれる良性と悪性サイトのドメインから WHOIS 情報を取得し，そのドメイン登録日を調査した．図 3 は良性と悪性ドメインそれぞれの登録期間の累積分布 (CDF: Cummulative Distribution Function) である．ここで，図 3 の横軸は，3.2.2 章で定義した各ドメインが登録されてからの登録期間 W であり，短いほど新しいドメインである．関連研究 [4, 5, 8] でも指摘されている通り，悪性サイトのドメイン登録日は，良性サイトに比べて新しいものが多いことが確認できた．この特徴を用いて，3.2.2 章で示した特徴抽出エンジン（WHOIS 情報分析）では，ドメイン登録日が新しいほど高い悪性度が付与されるよう特徴抽出を行なっている．ただし，古いドメイン登録日のものがすべて良性であるとは限らない．例えば，co.cc ドメインは 1997 年 10 月に取得されたドメインであるが，無料のサブドメインや URL 転送のサービスに利用されており，悪性サイトの温床となっている [19].

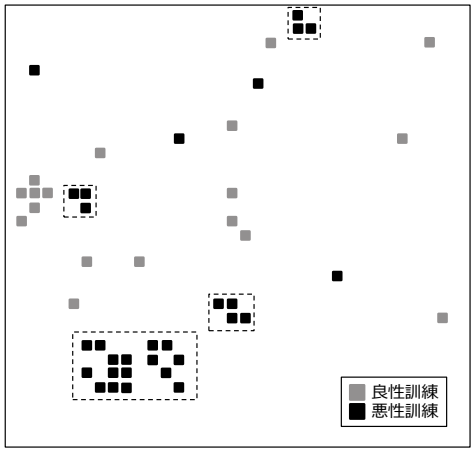

図 2: IP アドレス分布の可視化.

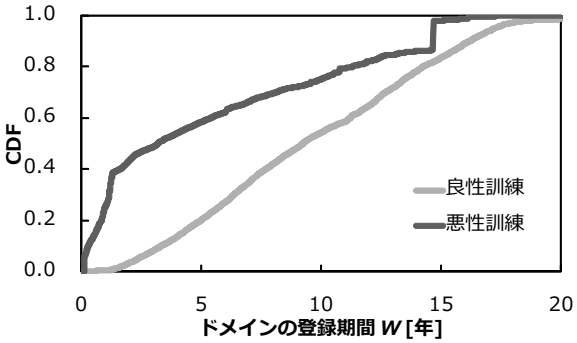

図 3: 訓練データのドメイン登録日の累積分布.

4.4 FQDN 文字列の特徴

4.4.1 文字列の長さの比較

訓練データセット（表 1）に含まれる良性と悪性サイトの FQDN 文字列の長さを比較し，図 4 にその累積補分布 (CCDF: Complementary Cumulative Distribution Function) を示す．図 4 より，悪性 FQDN はより長い文字列で構成されることがわかる．3.2.3 章に示した特徴抽出エンジン（FQDN 文字列分析）では，FQDN 文字列の長さに着目した特徴抽出を行なっている．

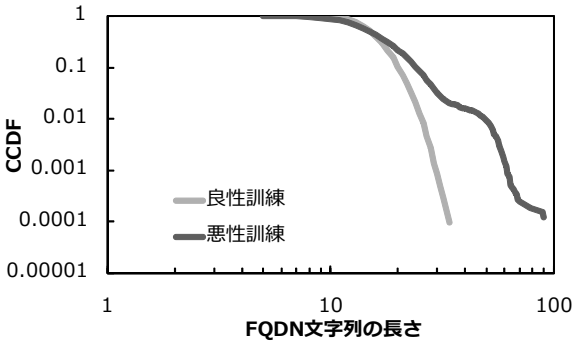

図 4: FQDN 文字列の長さの累積補分布.

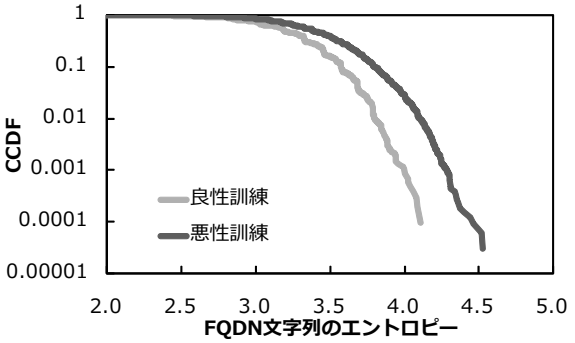

図 5: FQDN 文字列のエントロピーの累積補分布.

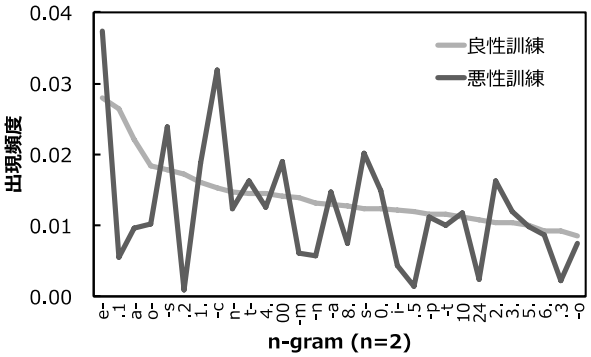

図 6: FQDN 文字列の n-gram の頻度分布.

表 3: 特徴抽出エンジンの組合せ.

特徴抽出エンジン	優先巡回 1	優先巡回 2	優先巡回 3	優先巡回 4
IP アドレス	✓	✓	✓	-
WHOIS 登録日	-	✓	✓	✓
FQDN 長さ	-	✓	✓	✓
FQDN エントロピー	-	✓	✓	✓
FQDN n-gram	-	✓	✓	✓
n-gram 例外処理	-	-	✓	✓

4.4.2 文字列のエントロピーの比較

訓練データセット（表 1）に含まれる良性と悪性サイトの FQDN 文字列のエントロピーを比較し，図 5 にその累積補分布を示す．なお，本研究における FQDN 文字列のエントロピーは 3.2.3 章にて定義している．図 5 より，悪性 FQDN のエントロピーは，良性よりも高いものが多いことがわかる．すなわち，悪性 FQDN は，ランダム性の高いホスト名やドメイン名で構成される場合が多い．3.2.3 章に示した特徴抽出エンジン（FQDN 文字列分析）では，FQDN 文字列のエントロピーを考慮した特徴抽出を行なっている．

4.4.3 文字列の n-gram の比較

訓練データセット（表 1）に含まれる良性と悪性サイトの FQDN 文字列の n-gram ($n = 2$) を調査した．その結果，n-gram 文字列のうち，少なくとも 1 文字が数字あるいは記号で構成されているものに，良性と悪性を識別し得る特徴があることを確認した．図 6 に出現頻度が上位 30 位までの n-gram の頻度分布を示す．図 6 より，良性と悪性のそれぞれで特徴的な n-gram 文字列が存在し，それらの出現頻度に差があることがわかる．これは，良性と悪性では FQDN に使用される文字列が異なるということを示しており，3.2.3 章に示した特徴抽出エンジン（FQDN 文字列分析）では，この特徴を利用した特徴ベクトル抽出を行なっている．

5 提案手法の評価

5.1 悪性サイトのヒット率

提案手法によって優先巡回順序を決定する際の，悪性サイトへのヒット率を計測する．ここで，優先巡回順序を決定するそれぞれの手法（優先巡回 1〜4）で利用する特徴抽出エンジンの組合せを表 3 に示す．優先巡回 1 では，特徴抽出エンジンから得られる特徴のうち IP アドレスのみを利用する．優先巡回 2 では，特徴抽出エンジンで得られるすべての特徴（IP アドレス，WHOIS 登録日，FQDN 文字列の長さ，FQDN 文字列のエントロピー，FQDN 文字列の n-gram）を利用する．優先巡回 3 では，3.2.3 章で解説した n-gram における例外処理を適用する．すなわち，例外ドメイン名に一致した FQDN からは，n-gram の特徴を抽出しない．優先巡回 4 では，IP アドレス以外のすべての特徴を利用する．

優先巡回 1〜4 を用いて優先巡回順序を決定したあとに巡回する場合と，ランダムに巡回する場合の，悪性サイトのヒット率の比較を行い，その結果を表 4 に示す．ここで，ある長さの巡回リストを選択した際に，その中に実際に含まれていた悪性サイト数の割合を悪性サイトのヒット率と定義する．ランダム巡回を行った際に，ヒット率が約 10%となるのは，テストデータセット（表 2）に含まれる，良性と悪性の比率が約 9:1 となっているからである．優先巡回 1〜4 を用いる場合は，優先巡回順序を決定した後に，上位順序から巡回リスト長分の URL を選択する．例えば，巡回リスト長が 1,000 の時は，優先

表 4: 悪性サイトのヒット率.

巡回リスト長	ランダム巡回	優先巡回 1	優先巡回 2	優先巡回 3	優先巡回 4
1,000	10%	100%	69%	94%	54%
5,000	10%	83%	71%	82%	32%
10,000	10%	56%	58%	63%	33%
20,000	10%	40%	41%	43%	32%
30,000	10%	30%	31%	31%	26%
40,000	10%	24%	24%	24%	21%
50,000	10%	20%	20%	20%	18%
60,000	10%	17%	17%	17%	16%
70,000	10%	15%	15%	15%	14%
80,000	10%	13%	13%	13%	13%
90,000	10%	12%	12%	12%	11%
100,000	10%	10%	11%	11%	10%

表 5: 悪性サイトの巡回速度.

悪性サイト発見数	ランダム巡回	優先巡回 1	優先巡回 2	優先巡回 3	優先巡回 4
100	1.00	1.33	0.77	0.94	1.41
500	1.00	4.62	3.04	3.60	4.81
1,000	1.00	6.22	4.35	5.06	2.71
2,000	1.00	7.72	5.63	6.42	2.99
3,000	1.00	8.36	6.15	7.09	3.11
4,000	1.00	7.56	6.14	7.10	3.20
5,000	1.00	6.37	5.77	6.65	3.20
6,000	1.00	5.11	5.27	6.10	3.13
7,000	1.00	4.94	4.62	5.40	2.96
8,000	1.00	3.61	4.14	4.76	2.57
9,000	1.00	3.10	3.30	3.55	1.80
10,000	1.00	1.81	2.02	2.09	1.34

巡回順序 1〜1,000 となった URL を巡回リストとして選択する．表 4 より，巡回リスト長が 5,000 までは，優先巡回 1 のヒット率が最も大きい．巡回リスト長が 10,000 を超えたあとは，優先巡回 2・3 のヒット率が優先巡回 1 よりも大きくなる．これは，巡回リスト長が小さいときには，優先巡回 2・3 の手法で，誤って良性サイトに高い優先巡回順序を付けてしまうエラーの影響が無視できないためである．このエラーについては，5.3 章で詳しく解説する．優先巡回 4 の場合は，優先巡回 1〜3 よりヒット率が低い．優先巡回 4 は，唯一 IP アドレスの特徴を利用していない手法であり，この結果より，IP アドレスが他の特徴に比べ，有効な特徴量であることがわかる．上記の実験結果より，提案手法によって優先巡回順序を決定することで，悪性サイトのヒット率が高い巡回 URL リストを得られることが確認できた．

5.2 総巡回時間・巡回速度

Web クライアント型ハニーポットを用いて巡回リストをランダムに巡回する場合と，優先巡回 1〜4 を用いて，優先巡回順序を決定したあとに巡回する場合の総巡回時間と巡回速度を比較する．ここで，ハニーポットの巡回にかかる時間としては，文献 [9] で提案されているマルチエージェント・マルチプロセス化されたハニーポットの実験結果を参照する．具体的には，10 個のハニーポットエージェントを 20 プロセス並列に動作させる場合に，1,000 個の URL を 600 秒で巡回できると仮定する．また，優先巡回順序の決定にかかる時間は，機械学習エンジンでの訓練時間およびテストデータに優先巡回順序を付与する時間と定義する．ただし，今回は特徴抽出エンジンでの処理時間は考慮していない．なぜなら，今回の提案手法で利用する特徴抽出手法は軽量のため，機械学習エンジンでの処理時間に比べ非常に小さいからである．なお，優先巡回順序決定に使うマシンは，CPU: Intel Xeon X3430 (2.40GHz)，メモリ: 4GB の Linux マシンである．

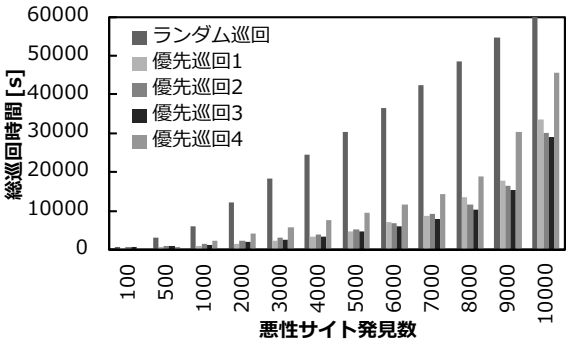

図 7: 悪性サイトの総巡回時間.

ある特定数の悪性サイトを発見するまでにかかる総巡回時間を図 7 に示す．図 7 より，同じ数の悪性サイトを発見するために，ランダム巡回と比べ，優先巡回 1〜4 の総巡回時間は大幅に短いことがわかる．この結果より，提案手法は軽量な手法であり，優先巡回順序決定までの時間はハニーポットの巡回時間に比べ，非常に小さいことがわかる．次に，総巡回時間から相対的な巡回速度を算出した結果を表 5 に示す．表 5 は，ランダム巡回の速度を 1.00 とした時の優先巡回 1〜4 の相対的な速度を示しており，優先巡回の数値が 1.00 以上の場合，ランダム巡回よりも高速であることを表す．実験結果より，悪性サイト発見数が 100 個の場合以外は，優先巡回順序を決定してから巡回することで，ランダムに巡回するよりも巡回速度を速くすることができ，最大で 8 倍以上速度を向上させることに成功した．一般に，悪性サイト発見数が増えるほど，ランダム巡回と優先巡回の差が小さくなる．なぜなら，テストデータに含まれる悪性サイトは表 2 に示す通り，合計 10,561 件であり，悪性サイト発見数が 10,000 に近づくほど，見つけるべき悪性サイトが減少するためである．

5.3 エラー分析

本来は良性サイトであるのにも関わらず，提案手法によって高い優先巡回順序が付与された場合の原因を巡回手法毎に分析する．

優先巡回 1 では，IP アドレスのみを特徴量として用いている．この優先巡回 1 では，良性と悪性サイトが混在するホスティングサイトにおいて，同一または近い IP アドレスが利用されている場合に，良性サイトに高い優先巡回順序を付与するエラーが発生した．

優先巡回 2 では，ランダム性の高い FQDN を持つ良性サイトの優先巡回順序を高く設定するエラーが発生した．エラーとなった良性 FQDN はいずれも悪性 FQDN に含まれるランダム性の高い文字列から得られる特徴と類似していた．具体的には，ソーシャルアプリケーションやブラウザの拡張機能におけるコンテンツの識別子として利用されるものや国際化ドメイン名 [20] に利用される Punnycode にランダム性の高い文字列が存在した．

優先巡回 3 では，例外ドメイン名による例外処理の追加により，優先巡回 2 で発生していた上記エラーが減少した．この結果，優先巡回 3 では，5.1 章と 5.2 章で示した通り，悪性ヒット率と巡回速度が優先巡回 2 に比べ向上した．

優先巡回 4 では，IP アドレスの特徴量を利用しないことで，ドメイン登録日が比較的新しく，長い FQDN 文字列を持つ良性サイトの優先巡回順序を高く設定するエラーが多く発生した．また，優先巡回 3 と 4 の優先巡回順序の比較により，IP アドレスの特徴の有無の影響が大きく，FQDN 文字列よりも IP アドレスの特徴がより有効なことがわかった．

6 まとめ

本研究では，限られたリソースしか持たない Web クライアント型ハニーポットが悪性サイトをより効率的に発見するための優先巡回順序の選定法を提案した．中心となるアイデアは，Web サイトの IP アドレス，WHOIS 情報，FQDN 文字列の情報から良性と悪性サイトを識別し得る特徴を抽出し，より悪性の可能性の高いサイトに高い優先巡回順序を付与することである．実データを用いた検証の結果，提案手法が悪性サイトのヒット率を高め，ハニーポットの総巡回時間の削減に大きく寄与することを示した．また，提案手法で用いる特徴量のうち，FQDN 文字列よりも IP アドレスから得られる特徴がより有効であることがわかった．しかし，今回の提案手法だけでは良性サイトの巡回順序を高く設定するエラーがあり得るため，今後の課題として URL のパス情報を含めた特徴抽出手法の拡張，巡回ポリシーに応じた優先巡回順序の調整手法の導入，実運用されている Web クライアント型ハニーポットを利用した評価を行い，より実用的なシステムにすることを検討している．

参考文献

[1] M. Akiyama, et al. “Searching structural neighborhood of malicious URLs to improve blacklisting,” Proc. of IEEE/IPSJ International Symposium on Applications and the Internet (SAINT 2011), pp.1–10, 2011.

[2] 八木 毅, “マルウェア感染を検知・制御するブラックリストシステムの設計,” 電子情報通信学会技術研究報告 信学技報, Vol.112, No.29, pp.49–54, 2012.

[3] 笠間 貴弘ほか “ドライブ・バイ・ダウンロード攻撃対策フレームワークの提案,” コンピュータセキュリティシンポジウム (CSS 2011), Vol.2011, No.3, pp.780–785, 2011.

[4] J. Ma, et al. “Beyond blacklists: learning to detect malicious web sites from suspicious URLs,” Proc. of ACM SIGKDD international conference on Knowledge discovery and data mining (KDD 2009), pp.1245–1254, 2009.

[5] M. Felegyhazi, et al. “On the potential of proactive domain blacklisting,” Proc. of USENIX conference on Large-scale exploits and emergent threats (LEET 2010), pp.6–6, 2010.

[6] S. Yadav, et al. “Detecting algorithmically generated malicious domain names,” Proc. of ACM SIGCOMM conference on Internet measurement (IMC 2010), pp.48–61, 2010.

[7] D. Chiba, et al. “Detecting malicious websites by learning IP address features,” Proc. of IEEE/IPSJ International Symposium on Applications and the Internet (SAINT 2012), pp.29–39, 2012.

[8] L. Invernizzi, et al. “EvilSeed: A Guided Approach to Finding Malicious Web Pages,” Proc. of IEEE Symposium on Security and Privacy, pp.428–442, 2012.

[9] M. Akiyama, et al. “Scalable and performance-efficient client honeypot on high interaction system,” Proc. of IEEE/IPSJ International Symposium on Applications and the Internet (SAINT 2012), pp.40–50, 2012.

[10] S. Hao, et al. “Detecting spammers with SNARE: spatio-temporal network-level automatic reputation engine,” Proc. of USENIX security symposium (SSYM 2009), pp.101–118, 2009.

[11] M. P. Collins, et al. “Using uncleanliness to predict future botnet addresses,” Proc. of ACM SIGCOMM conference on Internet measurement (IMC2007), pp.93–104, 2007.

[12] C. C. Chang, et al. “LIBSVM : a library for support vector machines,” ACM Transactions on Intelligent Systems and Technology, Vol.2, pp.27:1–27:27, 2011.

[13] J. Platt, “Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods,” Advances in Large Margin Classifiers, MIT Press, pp.61–74, 1999.

[14] “Alexa Top Sites,” http://www.alexa.com/topsites/

[15] “Malware Domain List,” http://malwaredomainlist.com/

[16] “Google Safe Browsing API,” http://code.google.com/intl/en/apis/safebrowsing/

[17] 千葉 大紀ほか “多種多様な攻撃に用いられる IP アドレス間の相関解析,” コンピュータセキュリティシンポジウム (CSS 2011), Vol.2011, No.3, pp.185–190, 2011.

[18] X. Cai, et al. “Understanding block-level address usage in the visible internet,” Proc. ACM SIGCOMM 2010, pp.99–110, 2010.

[19] Symantec Security Response Blog, “12 Million Exploit Attacks Originating from the CO.CC Domain,” http://www.symantec.com/connect/blogs/12-million-exploit-attacks-originating-cocc-domain

[20] “国際化ドメイン名,” JPNIC, http://www.nic.ad.jp/ja/dom/idn.html