PTRレコードの連続的設定を加味した動的IPアドレスブロックおよびクラウド領域検出

中森 朋郁[1,a)]　千葉 大紀[2,b)]　秋山 満昭[2]　後藤 滋樹[1]

概要：サイバー攻撃ではマルウェアに感染しボット化したマルウェア感染端末からの攻撃が多い．このようなマルウェアに感染したホストの多くは主に一般のユーザのものである．それらの IP アドレスの多くは動的 IP アドレスである．このような攻撃や異常トラヒックに対して IP アドレスで対処することがある．しかし，動的 IP アドレスは設定や状況によって必ずしも同一マシンに同一の IP アドレスが振られるわけではない．したがって，IP アドレスを基に正確なサイバー攻撃の対策を実施するためには，対象の IP アドレスが動的かどうか把握する必要がある．従来の動的 IP アドレスブロックの特定手法には，対象 IP アドレスの PTR レコード（DNS 定義ファイルに設定される逆引きのレコードで IP アドレスに対するホスト名を示す）に含まれる文字列を利用したキーワードマッチングや，対象 IP アドレスから送出されるトラヒック観測がある．しかし，事前に指定するキーワードでは網羅できない領域がある．また，実トラヒックを広く取得可能な事業者は限定される．本稿では動的 IP アドレスブロックの PTR レコードは連続的に設定されることが多いという特徴に着目して，動的 IP アドレスブロックを判別する手法を提案する．大規模かつ多様な範囲の実データを利用した評価の結果，提案手法は従来手法と比較して高精度の正解率を保持しながら PTR レコード登録があるすべての IP アドレスに対して判別が可能であることを確認できた．また，この手法で得られる情報とクラウド領域（クラウドサービスで利用される IP アドレスブロック）の特徴を用いてクラウド領域を識別可能であることを確認した．

キーワード：動的 IP アドレス，PTR レコード，アドレス割り当て，ネットワークマネジメント，クラウド

1. 序章

現在のインターネットにおいて動的 IP アドレスの数は膨大なものとなっている．悪意のあるスパムやマルウェア感染端末などの送信元として検出される IP アドレスは動的 IP アドレスであることが多い．また IP アドレスをユニークな識別子として，疑わしいホストをブラックリストに登録したり，異常トラヒックに対処したりする手法がある [2]．悪性行動が観測された送信元 IP アドレスが登録されたブラックリストが好例である [3], [4], [5]．ただし，動的 IP アドレスを振り直すとき ISP のポリシーやタイミング，領域によって必ずしも同じホストが同じ IP アドレスに割り振られる保証はない．動的 IP アドレスを悪意のあるホストとみなして上記の処置を施すことが有効とは限らない．そこで，IP アドレスを基に正確なサイバー攻撃の対策を実施するためには，対象の IP アドレスが動的かどうかを事前に把握する必要がある．このような把握をせずに，動的 IP アドレスが悪性ホストに使用されているからといってブラックリストでこの IP アドレスを制限すると，その IP アドレスが悪性でないホストに割り当てられたときに誤検知してしまう．このように変化する動的な領域を特定するために対象 IP アドレスの PTR レコード（DNS 定義ファイルに設定される逆引きのレコードで IP アドレスに対するホスト名を示す）に含まれる文字列を利用したキーワードマッチングや対象 IP アドレスから送出されるトラヒックを観測して特定する研究が行われてきた [6], [7], [8], [9]．具体的には「`dhcp`，`dynamic`」のような動的キーワードを用いる検出，取得したトラヒックデータをクラスタリングして解析する等の研究がある．しかし，PTR レコードは事業者がわかりやすいように設定し，事業者ごとに設定するキーワード，規則は異なる．これらをすべてリスト化するのは困難である．そのため，キーワードマッチングでは検出できない例があり，すべての IP アドレスを網羅するこ

1 早稲田大学
2 NTT セキュアプラットフォーム研究所
a) tomoshelfumiphy@suou.waseda.jp
b) daiki.chiba@ieee.org
本稿は APAN Research Workshop 2018 で発表された論文 [1] の拡張版である．

© 2018 Information Processing Society of Japan

とはできないという課題がある．この問題を解決するために我々はPTRレコードの連続性を用いた動的IPアドレスの検出手法を提案する．我々の手法を用いれば，キーワードで検出できない部分を網羅できる．提案手法は，DNSの逆引き結果を用いるだけなので解析に負担がかからず正確に動的IPアドレスを検出できる．また，クラウド領域（クラウドサービスで利用されるIPアドレスブロック）上の複数のホストを用いた攻撃も実際に観測されている [10]．クラウド領域では，動的IPアドレスブロックと同様にIPアドレスに割り当てられるホストが変更される可能性があり，クラウド領域検出もサイバー攻撃対策に大きな役割を持つ．クラウド領域では，PTRレコードをユーザが任意に登録できるという特徴を用いてクラウド領域検出手法を提案する．大規模な実データを利用した評価の結果は，提案手法が高い正解率を出しながらも，カバー率が従来手法より高いことを示した．カバー率とは入力としたIPアドレス全体に対して動的か静的か判定することができたIPアドレスの割合である．クラウド領域についても正確に判定できることを示した．本論文の貢献は次の通りである．

- 任意のIPアドレスとその前後に存在するIPアドレスに対応する逆引きレコードの連続的な特徴を利用して，動的IPアドレスブロック，クラウド領域を特定する手法を提案した．
- 実際に割り当てられているIPアドレスを利用して，動的/静的の各々のIPアドレスの特性を検証した．
- 大規模かつ多様な範囲の実際のIPアドレスブロックのデータを利用して検証した結果，提案手法は従来手法では検出できない動的IPアドレスブロックを正確に特定できることがわかった．
- 検出したIPアドレスブロックのデータを用いてクラウド領域の検出ができることを示した．

本論文の以降の構成は次の通りである．まず，背景として第2章でIPアドレスのタイプの定義を説明する．次に第3章で提案手法に関して具体的なステップと類似度の判定を説明する．第4章では今回行った評価実験について述べる．次に第5章で関連研究について紹介する．最後に本研究の結論を述べる．

2. 背景

2.1 IPアドレスの割り振り方

2.1.1 静的IPアドレスブロック

静的IPアドレスは一般にWebサーバ，DNSサーバ，FTPサーバに代表されるようなサービスを提供するサーバに用いられる．登録されるPTRレコードとしてはIPアドレスが紐づいているサーバのサービスの内容から名前を付けられることが多い．例えば，Webサーバなら「`www`」，DNSサーバなら「`ns`，`dns`」，FTPサーバなら「`ftp`」である．また，ルータやゲートウェイのようなネットワーク機器もこれらの名前からPTRレコードが登録される．例えば，ルータなら「`router`，`rt`，`rtr`」，ゲートウェイなら「`gateway`，`gw`」である．これらは基本的には機器ごとに手動で設定される場合が多い．これらのIPアドレスやそのPTRレコードには，人間にわかりやすい意味のある番号，あるいはキリの良い番号が使われる場合がある．また，組織に割り振られたIPアドレスブロックに比べて少数のIPアドレスしか使わないこともある．

2.1.2 動的IPアドレスブロック

インターネットの使用者は膨大な数である．これら大半のエンドユーザはインターネットを長期間継続して使用することは少ない．この場合に静的IPアドレスを割り振ると，使用されていない時間帯が多いのに一つのIPアドレスを占有してしまうので効率が悪い．この問題を解決する方法の一つに使用者が必要なときだけIPアドレスを割り振るように動的割り当てが行われる．例えばISPはPPPoEセッションごとにユーザに割り振りを行う．この際に使用されるIPアドレスブロックが動的IPアドレスブロックである．動的IPアドレスはタイミングによって同一ホストに同一IPアドレスを割り当てるとは限らない．そのため，同一IPアドレスを解析してもマシン，ユーザは異なることがある．このような動的IPアドレスブロックは可能な限り判別できることが好ましい．

2.2 動的IPアドレスブロック検出による利点

サイバー攻撃の送信元の多くは動的IPアドレスと考えられる．なぜならば攻撃者は自身が特定されることを避けるためにマルウェア感染端末（ボット）で構成されるボットネットを使用することが主流だからである．攻撃者は何らかの手法を用いて一般ユーザのマシンをマルウェアに感染させる．マルウェアに感染した大量の一般のマシンと予め用意しておいたCommand & Control (C&C) サーバを用いて攻撃者はボットネットを作成する．例えば，Miraiボットネットは脆弱なデフォルトパスワードのままのパソコンやモバイルデバイス，IoT機器などに侵入し，マルウェアをダウンロードさせる．その後，C&Cサーバからの攻撃命令を待ちながら，感染拡大のために脆弱なデバイスがないかを調べるためにネットワークをスキャンする．このようなマルウェア感染している一般のマシンからDenial of Service (DoS) 攻撃，スパムの送信などが行われる．Miraiボットネットは脆弱なユーザ名やパスワードを使用している組み込み機器やIoT機器に侵入して，大規模なDoS攻撃を行った [11], [12]．このようなサイバー攻撃を受けた際に送信元を解析してもタイミングによってはIPアドレスに紐づいているマシンが変更している場合がありうまくいかない．また，これらの攻撃に対して適切な処置を施すために動的IPアドレスブロックを判別することは重要な役割を持つ．

© 2018 Information Processing Society of Japan

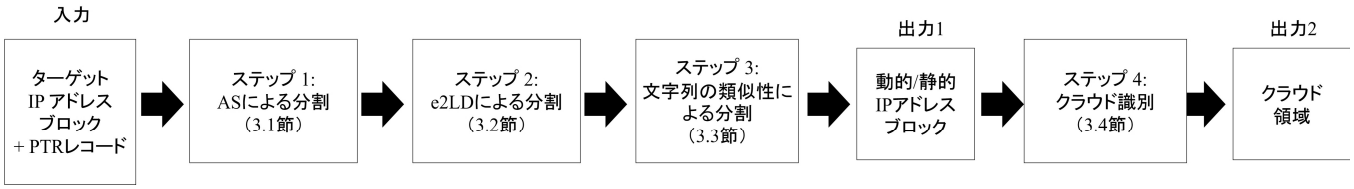

図 1　提案手法概略図

2.3　クラウド領域

マルウェアに感染するようなモバイルデバイスや IoT 機器が明示的な PTR レコード登録をすることは考えにくい．また，このような機器に割り当てられることが多い動的 IP アドレスは通常は ISP の管理下であると考えられ，エンドユーザが自由に PTR レコードを登録することはできない．一方で，クラウド上のホストの使い方は様々であり，メールサーバ，Web サーバとして利用する場合 PTR レコードを設定する必要がある．例えば，Amazon の AWS では PTR レコードを設定することができる [13]．悪性な行動がクラウド上のホストから送信されている場合には ISP による対策よりもクラウド業者が対策する方が望ましい．よって，クラウド領域を検出することもサイバー攻撃対策の重要な判断材料になる．

3.　提案手法

本稿では IPv4 アドレスの DNS の逆引き結果（以下 RDNS 結果という）から動的な IP アドレスブロックを判別する手法を提案する．本手法は RDNS 結果に記載されているすべての IP アドレスについて動的な IP アドレスブロックを判別する．本手法が従来手法と異なる点は，一つの IP アドレスの RDNS 結果に対して個々に判別を行うのではなく，前後の IP アドレスの RDNS 結果も参照して，連続性を加味して判別している点である．これは，一般的なネットワーク運用者が IP アドレスを管理する際の特徴を考慮している．動的な IP アドレスを大量に割り振る際には，機械的に番号などをつけて連続的な名前を設定することが多い．このような文字列は類似性が高く，この類似性が高いと判別できるブロックを動的な IP アドレスブロックと判別する．従来手法ではキーワードおよび正規表現により検出できない場合には動的か静的か判別できない IP アドレスが多く存在するが，本手法では動的か静的どちらかに判別することができる．**図 1** は提案手法の手順の概略を示す．提案手法は 4 つのステップから構成される．対象の IP アドレスブロックと PTR レコードの組を入力し，ステップ 3 までの各ステップが連続した IP アドレスの区切り（以後セグメントと呼ぶ）を認識して分割し，特定した動的/静的 IP アドレスブロックの情報を出力する．最終ステップで，この特定した IP アドレスブロックの情報を元にクラウド領域情報を出力する．なお，PTR レコードがデータとして入力されない場合は，IP アドレスブロック判定時に PTR レコード情報の取得を試みる．それでも情報を得られなかった場合には，その IP アドレスについては無視する．よって PTR レコードが抜けている IP アドレスを範囲内に含むセグメントもある．各々のステップを以降で順番に説明する．また**図 2** は提案手法による分割のイメージ図である．ステップ番号と線はそのステップが分割する部分を示している．また，最終的な動的か静的かの分類はステップ 3 が終わった段階で判断される．出力 1 は提案手法で動的か静的を判別した結果を示す．出力 2 は提案手法によるクラウド領域かどうかの判別結果を示す．

3.1　ステップ 1: AS による分割

入力の対象 IP アドレスブロック (例. `192.0.2.0/24`) の情報から，各 IP アドレスが属する AS 番号を調査する．AS 番号の情報は MaxMind 社のデータベース [14] を参照する．対象 IP アドレスブロックの中に異なる AS 番号に属するものが含まれる場合，その前後でセグメントを分割する．この理由は，AS が異なる場合にはその IP アドレスの管理主体や管理方法が大きく変わるからである．図 2 では `192.0.2.128` と `192.0.2.129` は AS 番号が異なるので分割される．`192.0.2.0` から `192.0.2.128` がこのステップで分割されてできる一つの大きなセグメントである．

3.2　ステップ 2: e2LD による分割

分割したセグメント内を effective second level domain (e2LD) が異なるかどうかでさらに細かく分割する．e2LD とは一般にユーザに登録されうるドメイン名の最小単位のことを指す [15]．e2LD の抽出には Mozilla Foundation が提供している Public Suffix List [16] を利用した．図 2 では `192.0.2.20` と `192.0.2.21` は e2LD が異なるので分割される．`192.0.2.0` から `192.0.2.20` までと `192.0.2.21` から `192.0.2.128` までの 2 つがこのステップで分割されてできるセグメントである．

3.3　ステップ 3: 文字列の類似性による分割

前節で細かく分割したセグメントに属する IP アドレスの PTR レコード (例. `host1.example.net`) から e2LD (例. `example.net`) を除いた文字列 (例. `host1`) を対象に，その文字列に類似性がある IP アドレスが 5 つ以上続く場合その IP アドレスブロックを動的ブロックと判別し，そうでない場合は静的ブロックと判別する．なお事前検証の結果，静的 IP アドレスと思われるものでも `www1.example.net`, `www2.example.net` など 4 つの静的 IP アドレスが続いていることが多く確認され，これらを動的と判定しないために閾値を 5 とした．図 2 において，例えば，ステップ 2

© 2018 Information Processing Society of Japan

	IPアドレス	ネットワーク	ASN	PTRレコード	e2LD	出力1	出力2
	192.0.2.0	192.0.2.0/25	AS64496	host1.example.net	example.net	Dynamic	非クラウド
	192.0.2.1	192.0.2.0/25	AS64496	host2.example.net	example.net	Dynamic	非クラウド
	…	…	…	…	…	…	…
ステップ 2	192.0.2.20	192.0.2.0/25	AS64496	host20.example.net	example.net	Dynamic	非クラウド
	192.0.2.21	192.0.2.0/25	AS64496	test21.example.com	example.com	Dynamic	クラウド
	192.0.2.22	192.0.2.0/25	AS64496	test22.example.com	example.com	Dynamic	クラウド
	192.0.2.23	192.0.2.0/25	AS64496	test23.example.com	example.com	Dynamic	クラウド
	192.0.2.24	192.0.2.0/25	AS64496	test24.example.com	example.com	Dynamic	クラウド
ステップ 3	192.0.2.25	192.0.2.0/25	AS64496	test25.example.com	example.com	Dynamic	クラウド
	192.0.2.26	192.0.2.0/25	AS64496	www1.example.com	example.com	Static	クラウド
	192.0.2.27	192.0.2.0/25	AS64496	mail.example.com	example.com	Static	クラウド
ステップ 3	192.0.2.28	192.0.2.0/25	AS64496	ftp.example.com	example.com	Static	クラウド
	192.0.2.29	192.0.2.0/25	AS64496	test29.example.com	example.com	Dynamic	クラウド
	…	…	…	…	…	…	クラウド
ステップ 1	192.0.2.128	192.0.2.128/25	AS64496	test128.example.com	example.com	Dynamic	クラウド
	192.0.2.129	192.0.2.128/25	AS64497	test129.example.org	example.org	Dynamic	非クラウド
	…	…	…	…	…	…	…

図 2 提案手法の分割の例

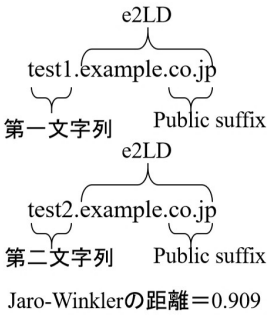

図 3 類似文字列の例

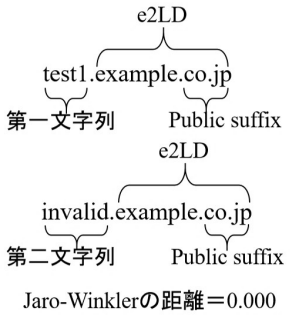

図 4 非類似文字列の例

までで分割された 192.0.2.21 から 192.0.2.128 までのセグメントをさらに分割することを考える．類似性がある PTR レコードは 192.0.2.21 から 192.0.2.25 までと 192.0.2.29 から 192.0.2.128 までであり，これらのセグメントが動的と判別される．192.0.2.26 から 192.0.2.28 までは類似性がないので静的と判別される．文字列の類似性は以下に示す 4 つの基準を順番に考慮する．ステップ 3-1 は IP アドレスが含まれる場合に連続していると判定する．IP アドレスを含まない場合に次のステップ 3-2 に進む．ステップ 3-2 で連続していると判定した場合には次のステップ 3-3 に進み，連続していないと判定した場合にはステップ 3 の処理を終了する．ステップ 3-3 も同様に，連続していると判定した場合には次のステップ 3-4 に進み，連続していないと判定した場合にはステップ 3 の処理を終了する．最後にステップ 3-4 でも連続していると判定した場合，2 つの文字列を連続していると判定する．そうでない場合連続していないと判定する．

3.3.1 ステップ 3-1: IP アドレスに沿った名前付け

ネットワークの管理上，IP アドレスを PTR レコードの一部として設定する場合も多い．これらは，先頭が第 1 オクテットで始まっているもの (例. 192-0-2-0.example.net) なら連続しているので正確に判定できるが，逆順に設定されているもの (例. 0-2-0-192.example.net) は後述する Jaro-Winkler の距離では判定できない．そこで IP アドレスが含まれる場合にはその数値を考慮して連続性を判定する．具体的には「-，_」を使用して区切っている IP アドレスの数値を正規表現を使って検出する．検出された場合は連続と判定する．ただし，ただの数字の誤検知を防ぐため，検査している IP アドレスと一致している場合にのみ連続していると判別する．

3.3.2 ステップ 3-2: Jaro-Winkler の距離

動的な IP アドレスを一定の範囲で設定する場合，ネットワーク運用者は機械的に“プレフィックス＋番号”（例. example0.net， example1.net）と設定することが多い．我々はこの性質に着目した．

本稿では，プレフィックスを考慮した文字列の類似性を判定する手法として Jaro-Winkler の距離 [17] を利用する．今回我々は Jaro-Winkler の距離を計算するために Python のパッケージを利用した [18]．Jaro-Winkler の距離は 2 つの文字列が完全一致している場合は 1，まったく一致していない場合は 0 になる．この手法は Jaro の距離 [19] を基にしている．Jaro の距離の定義を式 1 に示す．

$$\Phi_J = W_1 \cdot c/d + W_2 \cdot c/r + W_\tau \cdot (c-\tau)/c \qquad (1)$$

W_1：最初の文字列の文字に掛かる重み
W_2：2 番目の文字列の文字に掛かる重み
W_τ：文字の置換に掛かる重み
d：最初の文字列の長さ
r：2 番目の文字列の長さ
τ：置換した文字数
c：共通文字数

$c = 0$ のとき $\Phi_J = 0$ である．c は二つの文字の位置が $(max(d,r)/2) - 1$ 以内のときに一致した文字の数である．実際には $W_1, W_2, W\tau$ には 1/3 が使用される．これを用いて Jaro-Winkler の距離は式 2 のように定義される．

$$\Phi_{JW} = \Phi_J + p \cdot 0.08 \cdot (1 - \Phi_J) \qquad (2)$$

© 2018 Information Processing Society of Japan

p は共通のプレフィックスの文字数で，最大で 4 である．しかし，今回使用したパッケージではこの p は制限されていない．我々は事前検証により p への重みを 0.08 とした．Jaro-Winkler の距離は，スペルミスやタイプミスを検出する際に用いられる．また，Web での検索の際に一部の文字列だけで予測の候補を提案してくれる予測検索にも使用される．この手法であれば人間から見て類似していないと判断できる文字列同士を類似していると誤ることが少なく，また短い名前であっても適切に判定できる．これを用いて実際のデータから閾値を設定して類似している PTR レコードを判別した．今回閾値として 0.900 以上を類似性があると判定した．**図 3** に類似性がある文字列の具体例，**図 4** に類似性のない文字列の具体例を挙げる．図 3 は第一文字列と第二文字列の「`test`」の部分がプレフィックスとなっていて Jaro-Winkler の距離は 0.909 となり，類似していると判別される．図 4 では共通しているプレフィックスを持たないため Jaro-Winkler の距離が 0 となる．したがって類似していないと判別する．

3.3.3 ステップ 3-3: 静的キーワードを含む場合

上記の処理をしても一定数は静的 IP アドレスを誤判定してしまうこともある．動的と判別した IP アドレスのうち，明らかに静的な利用用途を示すキーワードを含む場合は静的と判断できるものとして処理を行った．今回設定した静的キーワードは従来手法 [6], [7], [8], [9] で使用されているキーワードを参考に設定した．具体的なキーワードリストを第 4.2 節に示す．

3.3.4 ステップ 3-4: 名前の長さ

IP アドレスブロックの中には，前後の PTR レコードとは特性が異なるものが静的 IP アドレスとして設定される場合がある．このような IP アドレスを動的と誤判定しないため，PTR レコードの文字列の長さが前後で大きく変化するところは，連続していないと判定する．具体的には，前後の文字列の長さが 3 文字以上異なる場合には連続していないと判定する．名前の付け方として番号に使うものは 3 桁までが多い．また，プレフィックスが長いと Jaro-Winkler の距離は 1 に近づき類似していると判定する．番号がついている名前同士なら長さの差はほぼ 2 文字以内だが，それ以外の長い文字列は 3 文字以上の差がでる．よって閾値を 3 文字とした．この処理をしない場合に誤判定してしまう文字列の例を**図 5** に示す．図 5 の 2 つの文字列は「`testlocalhost`」がプレフィックスとなっているが，その後に続く文字は片方は数字，片方は英字なので連続性が見られない．しかし，Jaro-Winkler の距離は 1 と完全一致の判定であり誤判定している．

3.4 ステップ 4: クラウド識別

クラウド上のホストの使い方は様々であり，メールサーバ，Web サーバなどとして利用する場合には PTR レコードを設定する必要がある．ブロックの途中に静的な用途を明示する PTR レコードが現れ，連続ではなくなることがある．この特徴と提案手法のステップ 3 までで得られた IP アドレスブロックの情報からクラウドの IP アドレス領域を検出する．上述した特徴は判定したアドレスブロックが動的，静的，動的と続いている状態である．また，同一の管理領域にある動的 IP アドレスブロックは同一の共通プレフィックスを持っているはずである．このような動的 IP アドレスブロックの間に静的 IP アドレスブロックがある場合，対象となる動的，静的 IP アドレスブロック両者をクラウド領域と判定する．図 2 では，`192.0.2.21` から `192.0.2.25` までと `192.0.2.29` から `192.0.2.128` までのセグメントが同一の共通プレフィックス `test` を持っている．そして `192.0.2.26` から `192.0.2.28` までの静的 IP アドレスブロックがこの二つの動的 IP アドレスブロックの間にある．よって，`192.0.2.21` から `192.0.2.128` までの IP アドレスブロックをクラウド領域と判定する．

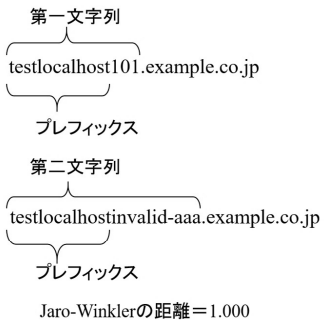

図 5 誤判定文字列の例

4. 評価

本章では実際の IPv4 アドレスとそれに対応する PTR レコードの情報を利用して，提案手法の妥当性の検証と，提案手法の動的 IP アドレス識別の精度の比較，クラウド領域検出の評価を行う．

4.1 データセット

4.1.1 IPv4 アドレスの PTR レコード

本稿の評価では，IPv4 アドレスとそれに対応する PTR レコードを以下に示す 2 種類の方法で収集した．まず，Project Sonar [20] によって公開されている Reverse DNS (RDNS) 結果を利用した．これは，IPv4 アドレスに対して DNS 逆引きを行った結果を公開しているデータである．Project Sonar は，セキュリティの研究のためにインターネット上の様々なサービス，プロトコルの脆弱性を明らかにするプロジェクトである．RDNS 結果は IPv4 アドレスの DNS 逆引きに応えたデータセットであり，ブラックリストに登録されている IP アドレスやプライベート IP アドレスの結果は含まれていない．なお，このデータには，

© 2018 Information Processing Society of Japan

表 1　ラベル付きデータセット

データセット	範囲	国	ISP	動的 IP アドレス数	静的 IP アドレス数
データセット 1	67.160.0.1/24	USA	Comcast	510	0
データセット 2	100.17.0.0/22	USA	Verison	1,011	5
データセット 3	104.0.0.0/22, 104.0.4.0/24	USA	AT&T	1,278	0
データセット 4	106.72.0.0/15, 106.128.0.1 - 106.185.45.254	Japan	KDDI	3,596,918	1,856
データセット 5	110.79.255.6 - 110.79.255.254, 110.158.0.1 - 110.158.2.20, 110.158.64.0/18, 110.158.128.0/17	Japan	OCN	44,668	39
データセット 6	143.0.84.0/22	Brazil	Sim Telecom	992	9

表 2　クラウド評価用ラベル付きデータセット

データセット	範囲	国	ISP	クラウド IP アドレス数	クラウド以外
データセット 7	175.41.129.0/24	USA	Amazon	256	0
データセット 8	69.104.56.0/24	USA	AT&T	0	256

スキャン時のエラー，PTR レコードが元々存在しないもの，利用されていない IP アドレスの場合には，対応する結果が存在しないことに注意したい．本稿では 2017 年 10 月 18 日時点にスキャンされたデータを利用した．次に，PTR レコードが抜けている場合には，IP アドレスブロック判定時での逆引き DNS 結果も合わせて取得して利用した．これは上記の Project Sonar の結果を補填するために行われた．提案手法の判定時点で最新の結果を参照するため，我々の実験環境から直接 DNS 逆引きのリクエストを送出した．この DNS 逆引きリクエストについては，過度にネットワークに負荷をかけないように，Project Sonar に結果が存在しないものに限定して，常識的な Rate でクエリを送出した．

4.1.2　正解データセット

提案手法により検出した動的 IP アドレス，静的 IP アドレスの検出結果が妥当であることを検証するために正解データを用いて実験により正解率を算出した．正解データは手動で動的，静的を目視で確認して作成し，複数の領域，地域で用意した．**表 1**，**表 2** はデータセットの詳細である．正解データセット 1，3，6 はすべて動的 IP アドレスで連続しているもの，データセット 2，4，5 は動的 IP アドレス，静的 IP アドレスが混在しているものである．データセット 7 はクラウド領域のみのもの，データセット 8 はクラウド以外の領域しか含まないものである．

4.2　動的 IP アドレスの識別精度比較

提案手法で判定した動的 IP アドレスブロック，静的 IP アドレスブロックが正解データセットに対して正しく判定できているかどうか実験を行った．また，比較のために，キーワードマッチングによる従来手法 [6], [7], [8], [9], S25R [21] の正解データセットに対する正解率を実験により求めた．

従来手法 [7], [8] は，PTR レコードの文字列から静的 IP アドレスと動的 IP アドレスをそれぞれ事前に指定したキーワードで特定している．さらに，メールサーバのログを解析したり，IP アドレスの実態調査をした結果動的 IP アドレスの特徴を捉える必要があり，その一環として事前に指定したキーワードで動的 IP アドレスを特定している従来手法がある [6], [9]．従来手法の検証として使用したキーワードは以下の通りである．

- 静的キーワード
  - `server, srv svr, mx, mail, smtp, www, ns, ftp, router, rtr, rt, gateway, gw, dns, sw`
- 動的キーワード
  - `dialup, modem, cable, hsb, dyn, dynamic, wireless, pool, access, dhcp, dialup, ppp, adsl`

S25R [21] は，スパムを送信するエンドユーザを識別するための正規表現やルールの組合せであり，IP アドレスの PTR レコードに対してルールにマッチするかどうかを調べる．また，一般規則，ブラックリスト，ホワイトリストと規則が 3 つあるが，一般規則以外はメールサーバを検出する場合がある．今回の比較評価では一般規則のみを使用した．正規表現によるルールを以下に示す．また，S25R は静的なものを検知できない仕組みであるから，常に True Negative，False Negative は 0 である．

- `^[^.]*[0-9][^0-9.]+[0-9].*\.`
- `^[^.]\*[0-9]\{5\}`
- `^([^.]+\.)?[0-9][^.]*\.[^.]+\..+\.[a-z]`
- `^[^.]*[0-9]\.[^.]*[0-9]-[0-9]`
- `^[^.]*[0-9]\.[^.]*[0-9]\.[^.]+\..+\.`
- `^(dhcp|dialup|ppp|[achrsvx]?dsl)[^.]*[0-9]`

提案手法は動的 IP アドレスを判別することを目的とする．そのため，動的 IP アドレスを Positive，静的 IP アドレスを Negative とみなして以後の評価を進める．**表 4** はこれらの関係を示したものである．True Positive (TP) と

© 2018 Information Processing Society of Japan

表 3 識別精度比較結果

データセット 1

	#TPs	#FNs	動的 IP アドレス検出数	#FPs	#TNs	静的 IP アドレス検出数	TPR	FNR	FPR	TNR	カバー率
提案手法	510	0	510	0	0	0	1.000	0.000	0.000	0.000	1.000
キーワード	0	0	0	0	0	0	0.000	0.000	0.000	0.000	0.000
S25R	510	0	510	0	0	0	1.000	0.000	0.000	0.000	1.000

データセット 2

	#TPs	#FNs	動的 IP アドレス検出数	#FPs	#TNs	静的 IP アドレス検出数	TPR	FNR	FPR	TNR	カバー率
提案手法	1,011	0	1,011	0	5	5	1.000	0.000	0.000	1.000	1.000
キーワード	759	0	759	0	0	0	1.000	0.000	0.000	0.000	0.747
S25R	1,011	0	1,011	4	0	4	1.000	0.000	1.000	0.000	0.999

データセット 3

	#TPs	#FNs	動的 IP アドレス検出数	#FPs	#TNs	静的 IP アドレス検出数	TPR	FNR	FPR	TNR	カバー率
提案手法	1,278	0	1,278	0	0	0	1.000	0.000	0.000	0.000	1.000
キーワード	0	0	0	0	0	0	0.000	0.000	0.000	0.000	0.000
S25R	1,278	0	1,278	0	0	0	1.000	0.000	0.000	0.000	1.000

データセット 4

	#TPs	#FNs	動的 IP アドレス検出数	#FPs	#TNs	静的 IP アドレス検出数	TPR	FNR	FPR	TNR	カバー率
提案手法	3,593,334	3,584	3,596,918	364	1,492	1,856	0.999	0.001	0.196	0.804	1.000
キーワード	1,041,779	640	1,042,419	0	405	405	0.999	0.001	0.000	1.000	0.290
S25R	3,596,841	0	3,596,841	489	0	489	1.000	0.000	1.000	0.000	0.999

データセット 5

	#TPs	#FNs	動的 IP アドレス検出数	#FPs	#TNs	静的 IP アドレス検出数	TPR	FNR	FPR	TNR	カバー率
提案手法	44,668	0	44,668	0	39	39	1.000	0.000	0.000	1.000	1.000
キーワード	0	0	0	2	29	31	0.000	0.000	0.065	0.936	0.001
S25R	44,668	0	44,668	2	0	2	1.000	0.000	1.000	0.000	0.999

データセット 6

	#TPs	#FNs	動的 IP アドレス検出数	#FPs	#TNs	静的 IP アドレス検出数	TPR	FNR	FPR	TNR	カバー率
提案手法	991	1	992	0	9	9	0.999	0.001	0.000	1.000	1.000
キーワード	0	0	0	0	7	7	0.000	0.000	0.000	1.000	0.007
S25R	0	0	0	0	0	0	0.000	0.000	0.000	0.000	0.000

表 4 各値の関係

		真の値	
		動的	静的
判定	動的	True Positive (TP)	False Positive(FP)
結果	静的	False Negative (FN)	True Negative (TN)

は動的 IP アドレスを正しく動的 IP アドレスと判定できたことを示す．False Negative (FN) とは動的 IP アドレスを誤って静的 IP アドレスと判定してしまったことを示す．False Positive (FP) とは静的 IP アドレスを誤って動的 IP アドレスと判定してしまったことを示す．True Negative (TN) とは静的 IP アドレスを正しく静的 IP アドレスと判定できたことを示す．TPR とは動的 IP アドレスを正しく動的 IP アドレスと判定できた割合を指し，$TP/(TP+FN)$ で定義される．FNR とは動的 IP アドレスを動的 IP アドレスと判定できなかった割合を指し，$FN/(TP+FN)$ で定義される．FPR とは静的 IP アドレスを静的 IP アドレスと判定できなかった割合を指し，$FP/(TN+FP)$ で定義される．TNR とは静的 IP アドレスを正しく静的 IP アドレスと判定できた割合を指し，$TN/(TN+FP)$ で定義される．また，**表 3** に実験結果を示す．カバー率はデータセット中の全 IP アドレスに対して動的，静的を検出した IP アドレスの割合である．

表 3 を見ると，TPR は提案手法，S25R ともに高い精度を出していることがわかる．従来手法のキーワードマッチング（表 3 中では紙面の都合上キーワードマッチングをキーワードと表記してある）も TPR は高い精度であるが，カバー率はデータセット 2 が 0.747 で最も高く，次にデータセット 4 が 0.290，他のデータセットはほぼ 0 であり，キーワードマッチングでは検出できる数が少ないことがわかる．また，S25R ではデータセット 6 でのカバー率が 0 であり，検出できない領域があることがわかる．データセット 2，4，5 では S25R は TPR と FPR が共に 1 であり，正しく判別できていないデータセットも存在する．数の多いデータセット 4 以外では提案手法は誤った判定をしていない．データセット 4 についても S25R よりも FP 数は少ない．また，動的だけでなく静的も判別できている．提案手法はカバー率はすべて 1 でありながらも従来手法と

© 2018 Information Processing Society of Japan

表 5　クラウド識別結果

	データセット 7	データセット 8
クラウド IP アドレス数	256	0
非クラウド IP アドレス数	0	256

同等以上の精度で動的 IP アドレスブロックを検出していることがわかる．

4.3　クラウド識別

本手法によりクラウドの IP アドレスブロックが正解データセットに対して正しく判定できているかどうか実験を行った．**表 5** を見ると，クラウド領域であるデータセット 7 ではクラウドをすべて識別できている．また，データセット 8 ではクラウドではない領域をクラウドではないと正しく識別できており，誤検知していないことがわかる．

5.　結論

マルウェア感染端末などへの対処や異常トラヒックへの対処に IP アドレスをユニークな識別子として用いるとき，タイミングによってはマシンと IP アドレスの対応が一意ではないことが対処を難しくしていた．これに対し，動的 IP アドレス検出の手法にはキーワードマッチングやトラヒック解析などの手法があるがキーワードマッチングは網羅的でなく，トラヒック解析は限定された事業者しかできないという課題があった．それを解決するため，我々は PTR レコードの連続性を用いた動的 IP アドレスの検出手法を提案した．実験により，従来手法ではなしえない動的か静的かどちらかに必ず判別できることを示すカバー率 1 を達成しつつ，従来手法と同等以上の結果を得ることができた．キーワードマッチングでは判別できる IP アドレスが極端に少ないことがわかった．また，スパムメールを検出するための S25R は良い結果を出したが，カバー率が悪い部分もあり，すべてを検出できるわけではないことがわかった．また，S25R は静的 IP アドレスを誤検知する可能性があることもわかった．クラウド領域では PTR レコードをユーザが任意に設定できるという特徴と本手法のブロック情報を元にクラウド領域を判定することができた．

提案手法による動的 IP アドレスブロック，クラウド領域特定結果は，マルウェア感染端末の検知や今後サイバー攻撃を受けた場合に送信元から判断した適切な対策検討に役立つと期待できる．

謝辞　本研究の一部は JSPS 科研費 16H02832 の助成を受けたものです．

参考文献

[1] Nakamori, T., Chiba, D., Akiyama, M. and Goto, S.: Detecting Dynamic IP Addresses Using the Sequential Characteristics of PTR Records, *Proc. Asia-Pacific Advanced Network (APAN)* (2018).

[2] Padmanabhan, R., Dhamdhere, A., Aben, E., kc claffy and Spring, N.: Reasons Dynamic Addresses Change, *Proc. ACM Internet Measurement Conference (IMC)*, pp. 183–198 (2016).

[3] Spamhaus: The Spamhaus Project Ltd., The Spamhaus Project., `https://www.spamhaus.org/`.

[4] Spamhaus: The Spamhaus Project Ltd., Compoist Blocking List., `http://www.abuseat.org/`.

[5] SORBS: Spam and Open Relay Blocking System (SORBS), `http://www.sorbs.net/`.

[6] Xie, Y., Yu, F., Achan, K., Gillum, E., Goldszmidt, M. and Wobber, T.: How dynamic are IP addresses?, *Proc. ACM SIGCOMM 2007 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pp. 301–312 (2007).

[7] Richter, P., Smaragdakis, G., Plonka, D. and Berger, A. W.: Beyond Counting: New Perspectives on the Active IPv4 Address Space, *Proc. ACM Internet Measurement Conference (IMC)*, pp. 135–149 (2016).

[8] Jin, Y., Sharafuddin, E. and Zhang, Z.: Identifying dynamic IP address blocks serendipitously through background scanning traffic, *Proc. ACM Conf. on Emerging Network Experiment and Technology (CoNEXT)*, p. 4 (2007).

[9] Cai, X. and Heidemann, J. S.: Understanding block-level address usage in the visible internet, *Proc. ACM SIGCOMM 2010 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pp. 99–110 (2010).

[10] Miao, R., Potharaju, R., Yu, M. and Jain, N.: The Dark Menace: Characterizing Network-based Attacks in the Cloud, *Proc. ACM Internet Measurement Conference (IMC)*.

[11] Akamai: Attack Spotlight: An Internet of Things Botnet, `https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/attack-spotlight-internet-of-things-botnet-threat-advisory.pdf`.

[12] Manos, A., Tim, A., Michael, B. et al.: Understanding the Mirai Botnet, *Proc. USENIX Security Symp.*, pp. 1093–1110 (2017).

[13] Amazon: Route 53 逆引き DNS, `https://aws.amazon.com/jp/premiumsupport/knowledge-center/route-53-reverse-dns/`.

[14] MaxMind: GeoLite Legacy Downloadable Databases, `https://dev.maxmind.com/geoip/legacy/geolite/`.

[15] Chiba, D., Akiyama, M., Yagi, T., Hato, K., Mori, T. and Goto, S.: DomainChroma: Building actionable threat intelligence from malicious domain names, *Computers & Security*, Vol. 77, pp. 138 – 161 (2018).

[16] Mozilla Foundation: Public Suffix List, `https://publicsuffix.org/`.

[17] Winkler, W. E.: Overview of Record Linkage and Current Research Directions, Technical report (2006).

[18] Haapala, A.: python-Levenshtein, `https://rawgit.com/ztane/python-Levenshtein/master/docs/Levenshtein.html`.

[19] Jaro, M. A.: Probabilistic linkage of large public health data files, *Statistics in Medicine*, Vol. 14, No. 5–7, pp. 491–498 (1995).

[20] Rapid7: Reverse DNS (RDNS), `https://opendata.rapid7.com/sonar.rdns_v2/`.

[21] Asami, H.: Study Report of an Anti-spam System with a 99% Block Rate, `http://www.gabacho-net.jp/en/anti-spam/anti-spam-system.html`.

© 2018 Information Processing Society of Japan