RA-002

文字列圧縮を用いたネットワークセキュリティにおけるインシデント検出
Detecting Network Security Incidents Based on String Compression

衛藤 公希[*] **小野 廣隆**[†‡] **山下 雅史**[§‡] **竹内 純一**[§‡]
Kouki Eto[*] Hirotaka Ono[†‡] Masafumi Yamashita[§‡] Jun'ichi Takeuchi[§‡]

1. はじめに

1.1. ボットネットワークとインシデント

（コンピュータ）ウイルスやワームと呼ばれる, システムの非合法な破壊や利用を目的とする悪性プログラム（マルウェア）による被害が急速に拡大しており, 最近は, 中でも**ボット** (bot) と呼ばれるウイルスによる被害は深刻になっている．ボットはネットワークを介して他のコンピュータに侵入するが, 侵入だけが目的でなく, 感染したコンピュータを悪意のある第三者の遠隔制御下に置くことで, それを更なる感染源として悪用する．たとえば, ボットに感染したコンピュータは，攻撃者が用意した指令サーバなどに自動的に接続され，数十から数百万台のボットに感染したコンピュータを従えた**ボットネット** (botnet) と呼ばれるネットワークを形成し，命令者から命令を受けて一斉に, 迷惑メールの送信，サーバーへの DDoS 攻撃，情報の漏えいといった反社会的活動 – これを**ネットワークセキュリティ上のインシデント**, 以下単にインシデント (incident) と呼ぶ –を行う．したがって, ボットネットの形成段階で, ボットが発するパケット集合からインシデントの予兆をできるだけ早く捕らえて, 攻撃に晒されようとしているコンピュータ群に警告を発することが重要である．

しかし, ボットの基本的特徴として，1) 感染したことに気付きにくい，2) 自分自身を自動的にアップデートできる，3) 亜種作成が容易であり種類が多い，ことが知られており, さらに, ボットネットからの攻撃の多くは DDoS 攻撃であり，したがって, パケットのそれぞれは正規のアクセスと区別がつかない場合が多い [4]. 要するに, 残念ではあるが, ボットネットによるインシデントは, 個々のコンピュータ・小規模ネットワークでは, それ自体の検知が非常に困難となる特徴を備えている．

1.2.nicter システム

独立行政法人情報通信研究機構（National Institute of Information and Communications Technology – NICT) では, **nicter** と呼ばれるネットワークインシデント対策センターを運営している [2]. nicter の二つの柱の一つは, **ダークネット** (dark Internet, dark address) と呼ばれる, 未使用 IP アドレス群に設置したセンサーへの到着パケット情報を用いたインシデント検知である.¶ ダークネットに到着する TCP パケットは, 本来, 正常な通信のためであるとは考えにくいので, そのほとんどがウイルスや迷惑メールの類であると考えられるが, この中から特にボットのインシデントを検知するのが主要な目的の一つである．(ダークネットを用いるマクロ解析を目的として)nicter が構築しているシステムを, 混乱を招かない限り, 本論文では nicter と呼ぶことにする．

インシデントの検出が非常に困難である理由を上で述べた．そこで, nicter では ChangeFinder[10] などのいくつかの統計分析ツールを用いて, 一部の検知機能の自動化を試みてきた．しかし, これらの分析ツールを用いた解析には十分な時間と十分なデータの蓄積が必要であり, したがって, 特に, 攻撃の初期挙動検知は専らオペレータの持つノウハウに委ねられている．そこで, NICT では自身らによる「インシデント分析の広域化・高速化（nicter システムの拡張・高度化)」研究プロジェクト, これらを実用化へ向けて発展させた, 総務省による「国際連携によるサイバー攻撃の予知技術の研究開発」研究プロジェクトなどを通して, 実回線モニターによる攻撃初期挙動検知手法の開発とそのシステムの実用技術化を行っている．

著者達は, これらの研究プロジェクトの一端を担い, ノンパラメトリックな統計手法である圧縮度‖に基づく分類手法を, インターネットパケット情報 (ログ) に対して適用してインシデントを検知することを提案し, その適用可能性を探るための基礎的実験を行った．そして, 満足できる結果を得たので, 「国際連携によるサイバー攻撃の予知技術の研究開発」プロジェクトの一環として, 実用化へ向けて実証実験システムへの実装を進めている．

本論文では, 行った基礎的実験の詳細を報告する．データ圧縮に基づく分類手法のアイデアを次に説明する．

1.3. データ圧縮による分類

ボットの特徴 2),3) から予想できることは, ある想定した特徴や事前知識に依存したインシデント検知手法は常に出し抜かれ, イタチごっこに陥るということである．したがって, あらかじめ想定した特徴や事前知識を一切用いないインシデント検知手法, すなわちノンパラメトリックな手法が望まれる．我々が適用するノンパラメトリックな手法は**データ圧縮に基づく分類**である [8, 13].

二つの文字列を x と y とし, x と y の類似性を定量化することを考える．相互情報量の概念を用いるまでもなく, x と y が似ているほど, x を用いることで y を

*福岡銀行
†九州大学大学院経済学研究院・経済工学部門, hirotaka@en.kyushu-u.ac.jp
‡九州先端科学技術研究所
§九州大学大学院システム情報科学研究院・情報学部門, {mak,tak}@inf.kyushu-u.ac.jp

¶nicter のもう一つの柱はハニーポットを用いたミクロ解析であり, ダークネットを用いるマクロ解析と組み合わせることで, 検知能力を向上させようとしている．ハニーポットを用いたミクロ解析は本論文とは直接の関係がないので, これ以上の説明は省略する．

‖本論文では, 手法の説明に**圧縮度**という用語を定義せずに使用するが, 直感的にはユニバーサル符号による圧縮率として理解いただきたい．詳しくは [13] 等を参照のこと．実際に実験で利用する**圧縮率**については 2.3 節で定義する．

大きく圧縮できる．すなわち, 任意のまともな圧縮プログラム C を固定し, 文字列 w を C で圧縮したときの w の記述長を $C(w)$ とするとき, x の記述を用いた y の記述長は $C(y|x) = C(xy) - C(x)$ で表現され, x と y が似ているほど, $C(y|x)$ は小さいと考えられる．ここで, xy は x と y の連接である．Li 達の貢献 [8] は, 非類似度を表わす距離である正規圧縮距離 (normalized compression distance　– NCD)

$$NCD(y,x) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}$$

を定義し, その妥当性をコルモゴルフ記述量の立場から検証し, それを用いたクラスタリングの可能性を提示したことにある． $C(x) \geq C(y)$ を仮定すると, $NCD(y,x) = C(y|x)/C(y)$ であり, y の圧縮における x の役割を $NCD(y,x)$ が測っていることが直観的に理解できる．

1.4. 研究目的

個人情報管理の観点から解析に利用可能なパケット情報は, 1) 通信が行われた時刻, 2) 送信元 IP アドレス, 3) 通信に用いられたプロトコル名, 4) 受信元ポート番号, に限られる．したがって, パケットの持つ重要な情報のほとんどが利用**不可能**である．nicter では全世界から nicter の管理するダークネットに流れ込むパケットを上述の利用可能な情報に基づき可視化した上で, 基本的にはオペレータのノウハウを総動員して, ボット由来のパケットを判別（推測）している．残念なことに, 通常 nicter にはボットに由来するパケットが多数到着する．しかし, オペレータにとっても正確な判断が不可能であることにボット攻撃検知の困難さの一端がある.**

本研究の大きな目標は, 上述の利用可能なパケット情報 (ログ) だけを利用した攻撃初期挙動検知手法を提案し, 実システム上で運用可能とすることにある．このための基礎的考察として, 与えられた利用可能なパケット情報 (ログ) の集合 (以下, 利用可能なパケット情報・ログを単にパケットとも呼ぶ) を, その中に含まれるボット由来のパケット量によって分類する手法を提案し, 実データを用いてその手法の有効性を検討した．

1.5. 手法とその有効性

我々に与えられるデータは時間区間 $T_i = [t_i, t_i + \delta)(i = 0, 1, \ldots)$ に到達したパケットの集合 P_i である．ここで, $t_{i+1} \geq t_i + \delta$ であり, 各パケット $p \in P$ は, 上述の情報 1)–4) の 4 項組である．我々は, これらのパケット集合 P_i をいわばオンライン的に処理して, 各 P_i をそこに含まれるボット由来のパケット数の割合によって分類したい．ここで, オンライン的と断っているのは, 初期挙動検知に目的を絞っているので, データを多量に蓄積した上での解析は目的に沿わないためである.†† 手法は素朴である．与えられたパケット集合 $P = \{p_1, p_2, \ldots, p_m\}$ から文字列 $x = p_1 p_2 \ldots p_m$ を構築し, 適当なユニバーサル圧縮アルゴリズム C(本論文での実験では LZW の亜種を採用している) を用いて圧縮する．あるボットネット由来のパケットが P に集まっていれば, アクセスパターンなどの**類似性から** $C(x)$ **は小さくなり**, その圧縮率から目的とする情報が抽出できるのではないか? これが著者達が描くシナリオであった．

結論を述べると, nicter が提供するパケット集合に対する実験により, $C(x)$ によって (オペレータの示唆する) ボット由来のパケットの多寡が特徴付けられることが明らかになった．さらには, その分類ではオペレータが見落としていたボット由来のパケットの発見にも成功している．すなわち, パケット集合に対する圧縮を用いれば, インシデントは高精度で検知可能と示唆する結果を得た．興味深いことに, 本実験の分類では, ボット由来のパケットを多く含むパケット集合 P は平常時のパケット集合と比べ, **より大きい** $C(x)$ により特徴づけられる, というものであった．すなわち圧縮度による分類は, 著者達の当初の予想とは異なる形で実現されたこととなる．本論文では, この実験の経緯ならびにその分析結果を詳細に述べる．なお, 本手法の延長上にある NCD などを用いた自動クラスタリングの研究も現在進行中であり, 予備実験の成功が確認されるとともに, 実証システム上における検証も進んでいる．こちらに関しても, 充分な結果が集まり次第の公表を予定している．

1.6. 関連研究と本研究の新規性

ボットを含むコンピュータウイルスやスパムの検知はネットワークセキュリティの観点から現在最重要の課題であると言ってよく, 様々な研究がある (マルウェアの現状は [11] を参照されたい). 小松 [6] はマルウェア検出技術を, パターンマッチング方式, ヒューリスティック方式, ジェネリック方式,　振舞い検知に分類し, 防御方式として, Web レピュテーション, E-mail レピュテーション, ファイルレピュテーション, スマートフィードバックと相関分析を挙げているが, それらの実現にはいずれもマルウェアの動作に関する十分な事前情報が必要であり, とても我々の目的に適用できるとは思えない．データ圧縮を用いたボットネットワークの検知に限って上の説明を補強すると, [1] は振舞い, [14] はバイナリ形式実行可能ファイル, [3] はコード構造情報をそれぞれを十分に知った上でのデータ圧縮を用いた解析である．やや古いが, ネットワーク上の DDoS 攻撃の検知をコルモゴルフ記述量の観点から試みるものとしては, [7] の研究がある．

本研究のように, 非常に限られたパケット情報だけを利用し, しかもデータの十分な蓄積がなくても行えるボットのインシデント解析は, 著者達の知る限りでは行われてこなかった．

本論文の構成は以下の通りである．第 2 節では基礎実験を行った環境と枠組みについて説明する．第 3 節では基礎実験結果と考察を行う．第 4 節では残された問題を整理し, 本論文をまとめる．

** これらの観察可能なパケット情報だけから, 個々のパケットがあるボットに由来するものか否かを判定することは事実上不可能である．ダークネットにアクセスするパケットの目的は, ほとんどの場合正常なものとは考えにくいことにも注意せよ．

†† 我々の手法は, 正確な意味でのオンラインアルゴリズムを導かない．本来の意味でのオンラインアルゴリズムでは, P_t が与えられると即座に解答する必要がある．

2. 提案手法の枠組みと対象データ

前節に述べたように, 本論文で提案するインシデント検出手法は, パケット情報（通信時刻, 送信/受信 IP アドレス, プロトコル, ポート番号) からなるデータを圧縮し, その振る舞いの特徴を元にインシデントの検出を試みるものである. 以上を踏まえ, 本節では実験で利用する圧縮アルゴリズムと解析対象となるログデータについて説明する.

2.1. 圧縮アルゴリズム

元来, 圧縮とは，データのもつ情報を保持したままデータサイズを削減する処理・符号化のことであり, データ通信の高速化や，データを保存する際に必要な記憶容量の削減のために用いられる. 本インシデント検出手法では, 最もよく知られたユニバーサルデータ圧縮法の一つである LZ 法に基づく圧縮アルゴリズムをエンジンとして採用する. LZ 法は情報源の知識を仮定しないユニバーサルな圧縮アルゴリズムである.

LZ 法では一般に “辞書” と呼ばれる文字列のテーブルを作成し，データ圧縮を行う. LZ 法は J. Ziv と A. Lempel が考案したものであり, 1977 年に発表されたスライド辞書法と呼ばれる手法を用いた LZ77 アルゴリズム [16] と，翌年に発表された動的辞書法と呼ばれる手法を用いた LZ78[17] アルゴリズムの二系統が知られるが, 本提案手法では LZ78 の後継である LZW アルゴリズム [15] の亜種 [9] を用いる.

2.1.1.LZW アルゴリズム

LZW アルゴリズムは，LZ78 を元に 1984 年に T.A. Welch が発表した圧縮アルゴリズムである. 動的辞書法と呼ばれる手法を用いており，辞書とこれに対応する辞書木を構築しながら符号化を行う.

LZW アルゴリズムの符号化 入力文字列のアルファベットサイズを σ としたとき, まず辞書に 1 文字のフレーズ σ 個を全て登録する. その際，フレーズ番号を 0 から $\sigma-1$ とし，根ノードとフレーズ番号に対応する葉ノードだけをもつ辞書木を構築する. 入力文字列を先頭から順に読み込みながら，既に辞書に登録されているフレーズと最長一致する文字列を探し, その文字列の末尾に 1 文字加えた文字列を新たなフレーズとして辞書に登録する操作を繰り返す. 最長一致した文字列のフレーズ番号を出力し, これを圧縮文字列とする. 本インシデント検知システムで用いる LZW に基づくアルゴリズム [9] (本論文では区別のため Re-LZW と呼ぶ) について説明する. Re-LZW と LZW の相違点は, 新たなフレーズの辞書登録方法にある.

Re-LZW アルゴリズムの符号化 LZW アルゴリズムと同様, まず辞書に 1 文字からなるフレーズ σ 個を全て登録し, 辞書木を構築する. 最長一致したフレーズを探す点も同じだが, 符号化を行っている段階の一致文字列と, その直前の段階の一致文字列とで構成される全ての文字列を新たなフレーズとして辞書に登録する点が LZW と異なる.

例えば，今回の一致文字列が CD，前回の一致文字列が AB であった場合, 辞書に登録するフレーズは ABC,ABCD となる. 出力は LZW と同様に最長一致した文字列のフレーズ番号である.

例として T =ababaaababa, $\sigma = 256$ として, Re-LZW を用いて圧縮を行った結果を示す. 先の例と同様に 1 文字のフレーズ全てが辞書の 0 番から 255 番まで登録されている. 最初の a に対応する 97，次の b に対応する 98 を出力し, この段階で ab を辞書の 256 番目のフレーズとして登録する. 次の最長一致文字列は ab であるので, 256 を出力するとともに 1 つ前の一致文字列 b と合わせて，ba,bab をそれぞれ辞書の 257,258 番目のフレーズとして登録する. 以下同様の処理を行うと，図 1 のように “97 98 256 97 97 259 257” という出力符号が得られる. 上側の数字は新たに辞書に登録す

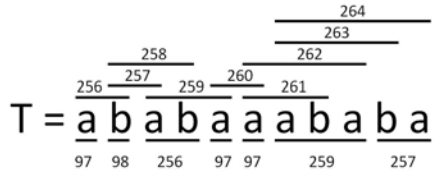

図 1: Re-LZW を用いた符号化の例

るフレーズのフレーズ番号を表している. このときの辞書木を図 2 に示す.

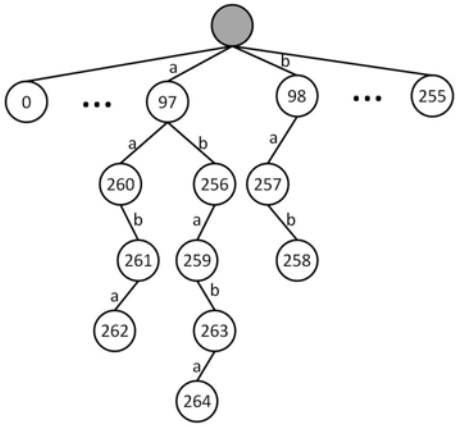

図 2: Re-LZW を用いたときの辞書木

この圧縮文字列を表すのに必要なビット数は $8+8+9+8+8+9+9=59$ ビットとなっており, LZW の 60 ビットとほとんど差はないが, 辞書に登録されているフレーズの数が異なる.

2.2. 対象データ

本インシデント検知システムが対象とする「パケット情報」はダークネットへの到着パケットのログ (データ) である. 通常到着パケットのログには様々な情報が含まれているが, 1 節で述べた通り, プライバシー保護などの理由により, 使用できる情報は以下の通りとなっている.

- Time：通信が行われた時刻
- Src Address：送信元 IP アドレス
- Protocol： 通信に用いられたプロトコル (TCP, UDP など)
- Dst Port：受信元ポート番号

すなわち, 各到着パケットごとにこれらの情報が付随し, これらの情報 (データ) 自体が解析の対象となる. そのデータは, Protocol と Dst Port を 1 つの情報として, 各情報はスペースで区切って表される. 例えば，12:00 に 111.222.333.444 アドレスから 123 番ポートに向けて TCP を用いた通信が行われた場合，”12:00 TCP123 111.222.333.444”という形の文字列が 1 パケットあたりのログデータとして得られる. すなわち対象データは, 1 パケットあたり 1 行の文字列データからなるテキストファイルとして与えられる.

2.3. インシデント検知手法の枠組み

提案するインシデント検知手法は, パケットを単位時間ごとにまとめて圧縮し, その圧縮度の特徴から異常（インシデント）検知を試みるものである. まず, このシステム構築に当たって, パケットログデータに対する前処理について述べる.

前述のように, 各パケット情報には Protocol の項目がある. 今回, 解析の対象とするプロトコルは, パケットに占める割合, P2P 通信などにおける利用を鑑みて TCP のみとする. 今後，Protocol と Dst Port を合わせた情報を単に Port と呼ぶことにする.

ログデータは 15 分ごとに分割して解析を行う. 一般に LZ 法はデータサイズにより圧縮率が大きく異なることが知られている（データが大きくなるほど圧縮率は良くなる）. このため, 単純にログを圧縮した際の圧縮度の違いだけでは, ログデータの異常・正常の判断が難しい. このため, サンプリングによりデータサイズの正規化を行う. 具体的には, 15 分間に到達したパケットのうち，ランダムに 3000 パケットを抽出したものを圧縮の対象とする. 以上がデータの前処理である.

こうして得られたデータは 1 日分で 96 個の 3000 行 (1 行が 1 パケットに対応する）からなるテキストファイルとなる. 本システムはそれぞれを圧縮し, その圧縮度を調べ, 異常を検知する.

圧縮には 2.1.1 節で述べた Re-LZW アルゴリズムにおいて，$\sigma = 256$ とし行う. このアルゴリズムの圧縮結果から, **圧縮率**を

$$\text{圧縮率} = \frac{\text{Re-LZW 圧縮後のファイルサイズ}}{\text{入力ファイルサイズ}} \quad (1)$$

として定義する (混乱を避けるため, 以下では圧縮率の値が小さいことを「良い」, 圧縮率の値が大きいことを「悪い」と表現する). ボットからの攻撃（インシデント）によるパケットが含まれている時間帯とその他の時間帯で, 圧縮率に顕著な違いが認められれば, 提案システムがインシデント検知システムとして機能することとなる.

ただし, ある時間帯にボットからの攻撃（インシデント）によるパケットが含まれているか否かを判定することは, それ自体が研究課題であり, 本手法だけではシステムの妥当性の判断は困難である. 以上から, 次節で扱う nicter の実データによる実験では, 予め経験ある nicter のオペレータによってインシデントパケットが含まれているか否かの判定がなされているログデータを用いて, 提案手法の妥当性について考察する.

3.nicter データに対する実験と考察

本節では NICT（情報通信機構）提供のダークネットへのパケット情報 (ログ) に前節で説明した提案手法を適用した結果について述べる. 使用するデータは nicter のオペレータによる解析により既知のボットからの攻撃によるパケットが特定されている. ただし未知のボットからの攻撃によるパケットが含まれている可能性は否定できないため分類は必ずしも正確ではない.

3.1. パケット数と圧縮率の関係

本解析では, ボットの活動が確認されているデータとそうでないデータにはどのような違いが表れるのかを調べる. 図 3 にボットの活動が確認されていないデータ（2008 年 9 月 9 日）のパケット数と圧縮率の関係を示す. 横軸は時間軸で，各点は 15 分間を表している. 青色の棒グラフが 15 分間に到達した総パケット数を, 赤色の折れ線グラフが圧縮率を表している. 図 4 はこれをパケット数昇順にソートしたものである. これを見ると，総パケット数が多い時間帯程，おおむね良い圧縮率を示していることがわかる.

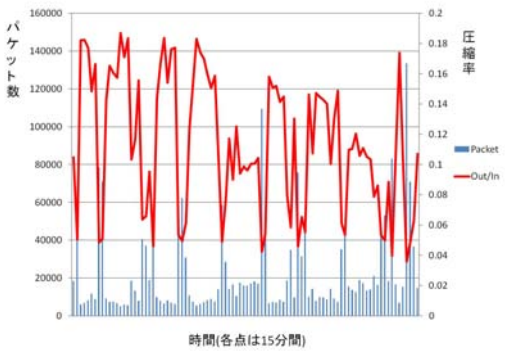

図 3: ボット活動が確認されていないデータのパケット数と圧縮率

次にボットの活動が確認されているデータ（2008 年 9 月 10 日）についての結果を図 5，6 にそれぞれ示す. 緑色の棒グラフは 15 分間に到達したボット活動によるパケットを表している. 比較すると，ボットが確認されていないデータとの違いがみられる. 図 4 では，総パケット数が多くなればなるほど, 圧縮率が良くなっているのに対し，図 6 では，ボットの活動が確認されている時間帯の圧縮率が悪くなっていることが見て取れる.

表 1 に各範囲のパケット数ごとの圧縮率の平均を示す. この集計でもボットの活動が確認されている時間帯でのログ圧縮率が，そうでない時間帯よりも悪くなっ

パケット数	2 万以下	2 万〜5 万	5 万〜10 万	10 万以上
2008/9/9(ボット活動未確認)	0.152	0.068	0.055	0.043
2008/9/10(同未確認)	0.139	0.065	0.057	0.043
2008/9/10(同確認済)	0.240	0.220	0.128	0.097

表 1: 圧縮率の平均（ボットの有無による比較）

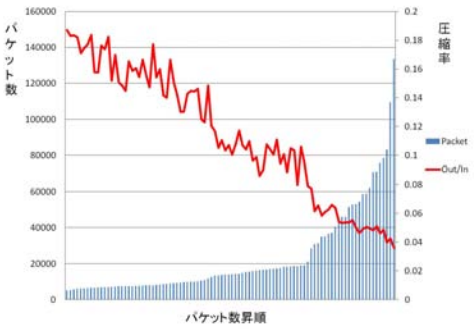

図 4: 図 3 をパケット数昇順でソートしたグラフ

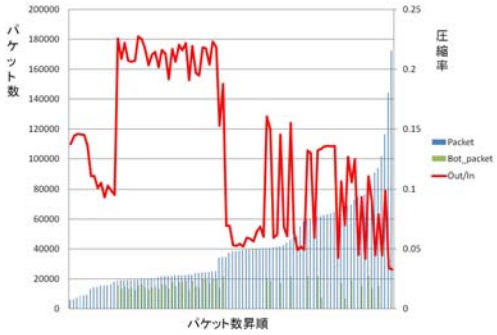

図 6: 図 5 をパケット数昇順でソートしたグラフ

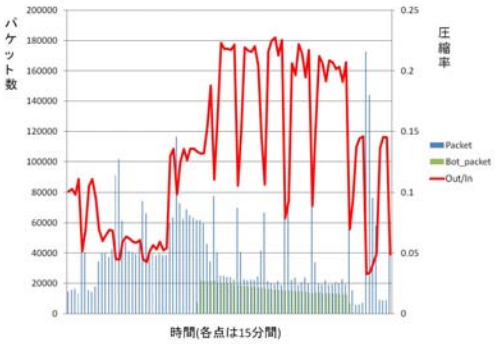

図 5: ボット活動が確認されているデータのパケット数と圧縮率

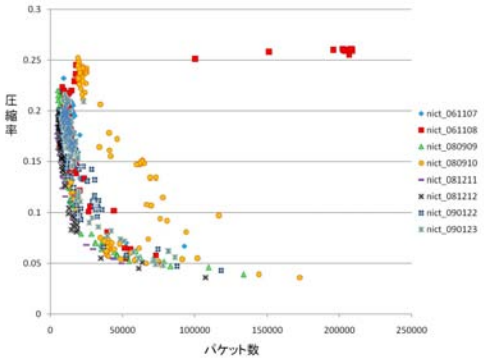

図 7: NICT1 についてのパケット数と圧縮率の関係

ていることが確認できる．

3.2. 複数データへの適用 (NICT1)

前項ではパケット数と圧縮率の関係を調べたが，これによりボット活動が確認されている時間帯にある種の特徴（圧縮率が悪い）が見て取れた．そこで本項では，更に多くのデータに対して同様の処理を行った結果を示す．その際, 横軸をパケット数, 縦軸を圧縮率として 1 つのグラフエリアに複数のデータに対しての結果を散布図として表す．

まず 2006 年から 2009 年 1 月までのデータ（これらのデータ群を NICT1 と呼ぶ）に対しての結果を図 7 に示す．先ほどと同様に各点は 15 分間を表しており，1 つの日付に対して 1 つの印を用いている．

これに対して，ボットが確認されている時間帯とそうでない時間帯に分類した結果を図 8 に示す．ボット活動が確認されている時間帯を赤色で, そうでない時間帯を青色で表している．ここで, 15 分間に到達した総パケット数のうち 10%以上がボットからの攻撃によるパケットであるときに，ボットが確認されていると定義している．また，緑色の曲線は青色の点の累乗近似曲線を表している．

青色の点 (ボット活動が確認された時間帯) は，黄色で囲んだ部分を除いておおむねこの曲線付近に出現しているが，この部分に関して改めて NICT に解析を依頼したところ, 当初の分類での確認漏れがあり, この黄色で囲んだ時間帯にもボット活動によるパケットが含まれていたとの回答を得た．したがって, 結果は図 9 のようになる．表 2 に各範囲のパケット数ごとの圧縮率の平均を示す．この場合も，ボットが確認されている時間帯では圧縮率が悪くなっていることがわかる．

NICT1 より新しい, 2009 年 4 月以降のデータ（これらのデータ群を NICT2 と呼ぶ）に関する実験に関しても同様の実験を行った．紙面の都合上，ここでの紹介を省略するが，NICT1 と同様にボットが確認されている時間帯とそうでない時間帯に分類した結果，NICT1 の結果と違い，日付による傾向の違いが大きいことがわかった．この原因については, 現在, 調査中である．

パケット数	2 万以下	2 万〜5 万	5 万〜10 万	10 万以上
ボット活動あり（総計）	0.169	0.088	0.057	0.042
ボット活動なし（総計）	0.232	0.220	0.128	0.246

表 2: 圧縮率の平均（NICT1 についてボット活動の有無による比較）

パケット数	2 万以下	2 万〜5 万	5 万〜10 万	10 万以上
2009/4/2(ボット未確認)	-	0.262	0.202	0.130
2009/4/2(同確認済)	-	-	0.266	0.217
2009/5/24(同未確認)	-	0.213	0.086	-
2009/5/24(同確認済)	-	0.199	0.220	0.205
2009/5/25(同未確認)	-	0.226	0.134	0.083
2009/6/21(同未確認)	0.207	0.167	0.083	-
2009/7/12(同未確認)	-	0.274	0.248	0.177
2009/7/12(同確認済)	-	-	0.271	-

表 3: 圧縮率の平均（NICT2 についてボットの有無による比較）

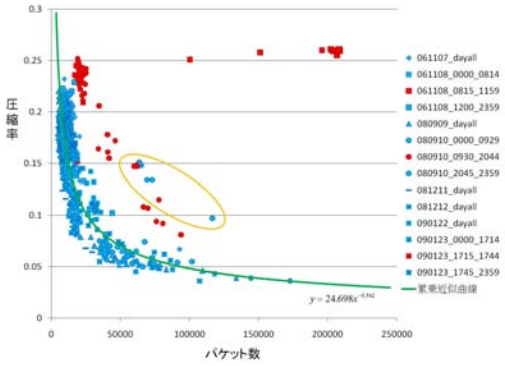

図 8: NICT1 についてボット活動の有無による分類

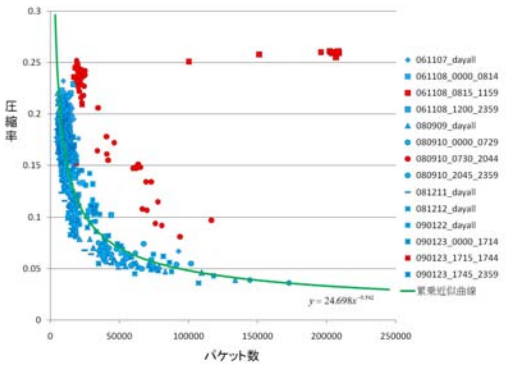

図 9: NICT1 についてボットの有無による分類（修正後）

3.3. 実験の考察

前節までの結果から，ボットが確認されている時間帯の方が，そうでない時間帯よりも圧縮率が悪くなっていることがわかった．これは仮定からの予想とは違う結果である．本節ではその原因を考察する．

使用した情報のうち，どれが圧縮率に影響を及ぼしているのかを調べる．上述のように使用した情報は, 時刻, Src Address, Port であるが, 時刻に関しては環境に対する敏感性から排除し, Src Address のみ，Port のみ，それら 2 つのみをそれぞれ抽出して圧縮したときの圧縮率を確認する．

図 3 で示したボットが確認されていないデータに対しての結果を図 10 に，図 5 で示したボットが確認されているデータに対しての結果を図 11 にそれぞれ示す．折れ線グラフについては，赤色が全ての情報の，黄緑色が Port のみの，紫色が Src Address のみの，水色が 2 つを合わせたものの圧縮率をそれぞれ表している．また，図 11 の灰色の棒グラフはボットの攻撃によるパケットを表している．黄色で囲んだ部分については図 8 で示した箇所と一致しており，この時間帯にも正確なパケット数はわからないものの，ボットによる攻撃が確認されている．

これらの図を見ると，ボットによる攻撃が確認されている時間帯では，そうでない部分と比較して明らかな違いが見て取れる．この時間帯では，Src Address のみの圧縮率は悪くなっており，Port のみの圧縮率は良くなっている．それぞれのデータについて，各時間帯の Src Address のみと Port のみの圧縮率の差の絶対値の平均を表 4 に示す．この表を見ると，ボットによる攻撃が確認されている時間帯では，両者の差が大きくなっていることがわかる．

Src Address は Port と比較して単純に文字数が多いため，その分圧縮率の差に及ぼす影響は大きくなるはずである．このことから Src Address の情報に引っ張られ，全体としての圧縮率は悪くなっていると考えられる．

予想とは違う結果が得られたが，ボットによる攻撃が確認されている時間帯とそうでない時間帯には違いが表れたので，今回の手法を用いてのインシデントの検出は可能であると考えられる．

2008/9/9(ボット未確認)	0.026
2008/9/10(ボット未確認)	0.016
2008/9/10(ボット確認済)	0.230

表 4: Src Address のみと Port のみの圧縮率の差の絶対値

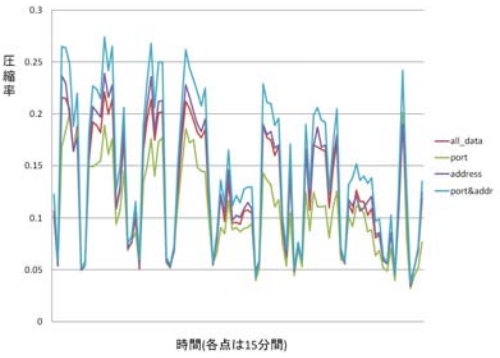

図 10: ボットが確認されていないデータについて各情報の圧縮率の比較

4. おわりに

本論文では, ダークネットにアクセスするパケット集合から, ボットネットの攻撃予兆を検出するシステムを構築するための基礎的考察として, ある時間帯に到着したパケット集合の中に含まれるボット由来のパケットの多寡を判定するためのデータ圧縮に基づく手法を提案し, 実験を通じてその効果を確かめた.

本研究の使命の困難さは第 1 節で述べたように, 1) 限られたパケット情報だけしか利用せず, 2) 検出手法を出し抜くボットの進化にも適応でき, しかも, 3) 攻撃初期挙動を検知しようとする所にある. 1) と 2) に対処する方針として, 我々はノンパラメトリック, すなわち, 事前の知識を必要としないデータ圧縮による手法を採用した. ボットネット由来のパケットがに集まっていれば, アクセスパターンなどの類似性から圧縮度が上がり, 圧縮度から目的とする情報が抽出できるのではないか, という著者達が当初描いたシナリオが崩れたことから, 手法の妥当性を疑うのは浅薄である.

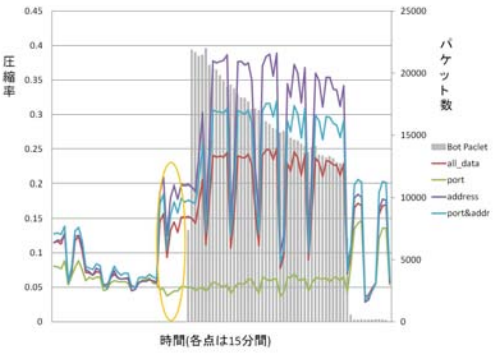

図 11: ボットが確認されているデータについて各情報の圧縮率の比較

そもそも我々が拠っている大前提は, 許されたパケット情報だけしか観察できないとても, ボット由来のパケットの集合はそれ以外のものとは「なんらかの意味で」違うということである. そして, 大雑把ではあるが, 違うものは違う圧縮度を持つ [8]. 本実験は, 見事にその事実を実証しているばかりか, ノンパラメトリックな手法の長所が発揮された例になっている. ボットが進化して, 今回圧縮率を下げた理由の除去に成功するかもしれない. そうなっても, ボット由来のパケットの集合はそれ以外のものとは違うという大前提が崩れない限りは, 別の特徴が浮かび上がってくることが期待できる.

最後に, 残された問題について触れておく. 本論文では, 本論文の対象に対して, 圧縮度が分離（クラスタリング）に利用できることを説明したが, オンライン的にどのようにして分離をするか説明していない. 先に紹介した文献 [1, 3, 14] を含めて, データ圧縮を用いた様々な分離手法が提案されているが, オンラインアルゴリズム的かつストリーミングアルゴリズム的 [12] な攻撃初期挙動検知に適した実用的な分離アルゴリズムは（著者らの知る限り）提案されておらず, 現在この方向での研究を検討中である.

今回, 圧縮度の計算には圧縮アルゴリズムとして Re-LZW を用いたが, 必ずしも Re-LZW である必要はない. 圧縮アルゴリズムによる検出精度の分析や, また正規圧縮距離の解釈を緩和する意味で, 非可逆圧縮アルゴリズムを採用した解析も興味深いテーマである [5].

謝辞

本研究に対して有益な助言をいただいた九州大学の来嶋秀治氏, 藤永直氏に感謝する. 本研究で利用したデータは情報通信研究機構 (NICT) の nicter の提供によるものである. NICT ネットワークセキュリティ研究所の井上大介氏, 衛藤将史氏, KDDI 株式会社の中尾康二氏に感謝する. また多くの有益なコメントをいただいた FIT 査読論文の査読委員に感謝する. 本研究の一部は NICT 委託研究「インシデント分析の広域化・高速化技術に関する研究開発」, 総務省委託研究「国際連携によるサイバー攻撃の予知技術の研究開発」の一環としてなされたものである.

参考文献

[1] Bailey, M., Oberheide, J., Andersen, J., Mao, Z., Jahanian, F. and Nazario, J.: Automated classification and analysis of internet malware, *Proc. Recent Advances in Intrusion Detection*, pp. 178–197 (2007).

[2] 衛藤将史　ネットワークインシデント対策センター～より安心・安全なインターネットの実現を目指して～, 独立行政法人情報通信研究機構（オンライン）, `http://www.nict.go.jp/publication/NICT-News/0607/research/index.html` 参照 2006-07.

[3] Gong, T., Tan, X. and Zhu, M.: Malware detection via classifying with compression, *Proc. 1st Conf. Information Science and Engineering*, pp. 1765–1768 (2009).

[4] 出雲教郎 振る舞い検知型 IPS の技術解説, 日本ラドウェア株式会社（オンライン）, `http://www.atmarkit.co.jp/fsecurity/special/142ips/ips01.html` 参照 2009-05-22.

[5] 岩本圭一郎 サイバー攻撃に対する非可逆圧縮を用いた検知法の提案, 九州大学工学部電気情報工学科, 2011 年度卒業論文.

[6] 小松優介 マルウェアと戦う技術–「Web からの脅威」とマルウェア検出・防御技術，情報処理，Vol. 51, No. 3, pp. 261–269 (2010). 特集「マルウェア」.

[7] Kulkarni, A. B., Bush, S. F., and Evans, S. C.: Detecting distributed denial-of-service attacks using kolmogorov complexity metrics, *GE Research & Development Center*, (2001).

[8] Li, M., Chen, X., Li, X., Ma, B. and Vitanyi, P.: The similarity metric, *IEEE Trans. Information Theory*, Vol. 50, No. 12, pp. 3250–3264 (2004).

[9] Storer, J. A.: *Data Compression: Methods and Theory*, Computer Science Press (1988).

[10] Takeuchi, J. and Yamanishi, K.: A unifying framework for detecting outliers and change points from time series, *IEEE Trans. Knowledge Data Engineering*, Vol. 18, No. 4, pp. 482–492 (2006).

[11] 寺田真敏, 他 特集「マルウェア」, 情報処理, Vol. 51, No. 3, pp. 235–303 (2010).

[12] 徳山豪 オンラインアルゴリズムとストリームアルゴリズム，共立出版 (2007).

[13] Vitanyi, P.: 圧縮度にもとづいた汎用な類似度測定法, 数理科学, Vol. 44, No. 9, pp. 54–59 (2006). 渡辺治訳.

[14] Wehner, S.: Analyzing worms and network traffic using compression, *J. Computer Security*, Vol. 15, No. 3, pp. 303–320 (2007).

[15] Welch, T.: A Technique for High-Performance Data Compression, *IEEE Computer*, Vol. 17, No. 6, pp. 8–19 (1984).

[16] Ziv, J. and Lempel, A.: A universal algorithm for sequential data compression, *IEEE Transactions on Information Theory*, Vol. 23, No. 3, pp. 337 – 343 (1977).

[17] Ziv, J. and Lempel, A.: Compression of individual sequences via variable-rate coding, *IEEE Transactions on Information Theory*, Vol. 24, pp. 530 – 536 (1978).