L-003

パケット情報を用いたトラフィック可視化システムの作成
A development of a traffic visualization system based on packet-capture information

田村 尚規† 甲斐 博† 森井 昌克‡
Naoki Tamura Hiroshi Kai Masakatsu Morii

1. はじめに

インターネットの利用者数の増加等を背景に，悪意を持ったソフトウェア（マルウェア）による被害も増加している．このため，マルウェア検知は以前にも増して重要になっている．マルウェアに感染したコンピュータはユーザの意図しない通信を行うことが知られている．この特徴から，通信状況，つまりパケットの送受信状況を観察することで，マルウェア感染の有無を検知することができる．一般にパケットの送受信状況を観察するにはパケットキャプチャツールが用いられる．しかしツールを扱うには専門知識が必要であり，万人に向けた手法とは言い難い．このため，通信状況を可視化するなどして直感的理解を助けるための手法が必要になる．

ネットワークのトラフィックやアクセス状況を可視化する研究に，独立行政法人情報通信研究機構（NICT）が推進する NICTER（Network Incident analysis Center for Tactical Emergency Response）[1]や DAEDALUS[2]がある．特に NICTER では地図や地球儀上に線や矢印を表示し，これらの動きによってトラフィックを表現し，色わけ等による表示方法の多様化によって視覚的に状況の把握を容易にしている．しかしながら，NICTER はダークネット（未使用のアドレス空間）を観測する仕様となっており，DAEDALUS は個々のネットワーク(IP アドレス）が送出する異常パケットの観測を対象としている，他方，標的多メール攻撃等，サイバー攻撃の初期段階として，個人のコンピュータを対象とした攻撃が問題となっており，まず個々のパソコン等のネットワーク端末の異常を検知する方法が希求されている．

本研究では個人のパソコン，あるいはネットワーク端末を対象とし，その直接的なパケットの送受信状況から異常を検知し，特にマルウェアの感染を早期に検出することを目的としている．その第一段階として，そこから送受信されるパケットを可視化し，そのパケットの異常状態を直感的には把握できるシステムの提案を行い，実装を試みる．

2. ツールを用いたトラフィック分析

個人がコンピュータ上のトラフィックを分析する際には，専らパケットキャプチャツールが利用される．例として，パケットキャプチャツールの TCPEye[3]はプロトコルや IP アドレスの識別といった一般的なパケットキャプチャツールが持つ機能に加え，通信先の地域の表示や，外部サービスと連携したウィルスチェックといった有用な機能を備えている．TCPEye の他にも，Wireshark[4]等，多くのパケットキャプチャツールが存在する．

ツールがパケット情報を取得する際には，内部でパケット解析用の API が利用される場合も多い．代表的なパケット解析 API である pcap[5]は，OSI 参照モデルにおけるデータリンク層上にパケットを送り出し，利用可能なインタフェースのリストを検出することができる．また，キャプチャ結果をファイルに保存し，そのファイルを他のアプリケーションから読み込むこともできる．これは前述した Wireshark の内部にも使われている．

しかし，これらのツールの出力からプロトコルや IP アドレスから異常を見分けるには専門的な知識が必要になる．このため，一見して異常を見分けるのは難しいという問題がある．

3. 提案手法

本研究で構築するシステムの概要を述べる．システムの機能を 3 つに大別し，図 1 に示す流れで処理を行う．

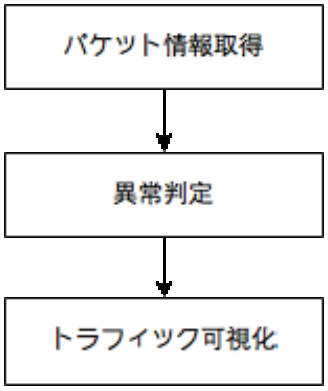

図 1　処理の流れ

3.1 パケット情報取得

異常検知に必要となる情報として，本研究では送受信されるパケットから取得する情報は，取得日時（年，月，日，時，分，秒），送信元 IP アドレス，宛先 IP アドレスを取得した．なお，今回対象とするパケットは IPv4 パケットに限定した．また，パケット情報を取得する手段として WinPcap[5]を用いた．これは，Windows 用の pcap の API である．

3.2 異常判定

前節で述べたパケット情報が異常なパケットであるかを判定するために，本研究では，事前にホワイトリストを設定しておき，リストとパケット情報の比較によって異常の有無を判定する方法を用いる．本研究で作成したホワイトリストでは，過去 2 日分の通信履歴を正しい通信とし，リストを作成する．

† 愛媛大学　Ehime University
‡ 神戸大学　Kobe University

リストに保存する情報は取得日時および IP アドレスとする．可視化時において送信と受信それぞれを個別に可視化できるようにするため，IP アドレスは受信パケットと送信パケットで区別する．観測したパケットの IP アドレスとホワイトリストの IP アドレスが一致していれば正常，それ以外の場合は異常と判定する．

3.3 トラフィック可視化

異常判定を終えたパケット情報をウェブブラウザで読み込み，可視化処理を行う． ウェブ上での動画表現には 3D グラフィクス API である WebGL を利用して行う． ブラウザの表示領域上に，座標（緯度・経度）を設定した地球儀オブジェクトを表示する． パケットの送信元もしくは宛先の IP アドレスから座標を割り出し，送信元と宛先をアーチ状のフローで結ぶ．トラフィックの多い通信間では，フローを重ねることで表現する．正常であると判定された箇所は青，それ以外は赤色で表示させる．座標の特定には GeoIP[7]を用いる．GeoIP は MaxMind 社が提供する API であり，IP アドレスから地理情報を得ることができる．本システムでは，PHP から GeoIP 関数を呼び出すことで利用する．GeoIP 関数にパケット情報の送信元 IP アドレスおよび宛先 IP アドレスを与え，返ってきた位置情報を地球儀の座標と対応させる．

4. 実行画面

構築したシステムの動作例を示す．動作例に用いた計算機環境を表 1 に示す．

表 1　計算機環境

OS など	Windows 8.1 （WinPcap 4.1.3, PHP 5.6.8, GeoIP 1.14）
ウェブブラウザ	Google Chrome 43.0.2357.124 m
システム設置場所	愛媛大学城北キャンパス 工学部 4 号館内

システムを実行し，パケット情報から異常の有無を判定し，可視化を行った．今回の検証では，トラフィックの観察は 1 分間とした．可視化の様子を図 2 に示す．図 2 を見ると，フローが日本と通信先とを結んでいることから，検証に用いたコンピュータと各地のコンピュータがインターネットを介して通信していることが理解できる．また，赤色のフローが見られることから，ホワイトリストに合致しないパケットが検出されたものと考えられる．取得したパケット情報とホワイトリストを比較すると，ホワイトリストにない IP アドレスとシステムで赤く可視化されたフローの出現地域と一致していることがわかり，可視化により異常の確認が容易になった．

5. おわりに

本研究では，個人のパソコン，あるいはネットワーク端末を対象とし，その直接的なパケットの送受信状況から異常を検知し，特にマルウェアの感染を早期に検出することを目的として，そこから送受信されるパケットを可視化し，そのパケットの異常状態を直感的には把握できるシステムの提案を行い，実装を試みた．

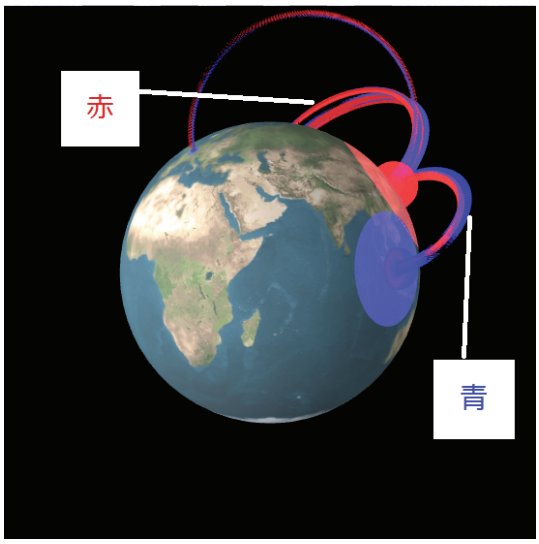

図 2　トラフィック可視化の様子

現時点では，送受されたパケットの送受信アドレスに対して，異常判定を行うためのホワイトリストに関しては十分な精査が行われておらず，単に過去に良性と判定されたアドレスを保存しているのみである．今後はホワイトリストに関しては，学習機能やクラウド上の多数の集合知からリストの更新や形成を行う等によって改善する予定である．また，ネットワーク上で公開している悪性サイトリスト等を利用する事によって，ブラックリストを構成，更新し，パケットの良性悪性判定の効率化，高精度化を試みる予定である．さらに合わせて可視化手法の改善やスマートフォンやタブレット端末を想定したユーザビリティの向上を行う．

参考文献

[1] 中尾康二，松本文子，井上大介，馬場俊輔，鈴木和也，衛藤将史，吉岡克成，力武健次，堀良彰，“インシデント分析センタ nicter の可視化技術”，情報処理学会 研究報告 CSEC， 34， pp.313-319 (2006).
[2] 鈴木未央, 井上大介, 衛藤将史[他], 宇多仁, 中尾康二, “大規模ダークネット観測に基づくアラートシステムの実装と運用”, 電子情報通信学会技術研究報告 ICSS, 情報通信システムセキュリティ，110(475), pp.59-64 (2011).
[3] TCPEye Network Tools, http://tcpmonitor.altervista.org/tcpeye-network-monitoring/.
[4] Wireshark, https://www.wireshark.org/.
[5] WinPcap, http://www.winpcap.org/.
[6] 北川直哉, “プロトコル検証に基づく不正通信ホスト識別手法の研究”, 名古屋大学博士論文(2014).
[7] GeoIP, http://dev.maxmind.com/geoip/.