ダークネットにおける Android 端末の通信分析

鈴木 貴之† 鈴木 男人† 笠間 貴弘†† 島村 隼平††† 井上 大介†† 宮保 憲治†

†東京電機大学大学院 情報環境学研究科 270-1382 千葉県印西市武西学園台 2-1200
††情報通信研究機構 184-8795 東京都小金井市貫井北町 4-2-1
†††株式会社クルウィット 181-0013 東京都三鷹市下連雀 3-34-8 三鷹ハイデンス 509 号

Email：†{14jkm10, 13jkm14}@ms.dendai.ac.jp, †miyaho@mail.dendai.ac.jp
††{kasama, dai}@nict.go.jp, †††shimamura@clwit.co.jp

あらまし Android OS上で動作するマルウェアが増加している．Windowsマルウェアとは異なり，Androidマルウェアの大半は正規のアプリケーションに偽装することにより感染を試みる．しかしながらWindows PCからAndroidへ感染しようとするマルウェアやAndroidからWindows PCへ感染しようとするマルウェアが出現するなど，マルウェアの感染経路は多様化しつつある．本稿では，TCP/IPヘッダ内の情報を活用して，Android端末からのトラフィックを識別する手法を提案する．また，本提案手法をダークネット観測データに適用することにより，Android端末からのスキャン活動を分析する．

Communication Analysis of Android Devices in the Darknet

Takayuki Suzuki† Nanto Suzuki† Takahiro Kasama††

Jumpei Shimamura††† Daisuke Inoue†† Noriharu Miyaho†

†Graduate School of Information Environment, Tokyo Denki University
2-1200, Busai-Gakuendai, Inzai, Chiba, 270-1382, JAPAN
{14jkm10, 13jkm14}@ms.dendai.ac.jp, miyaho@mail.dendai.ac.jp

††National Institute of Information and Communications Technology
4-2-1, Nukui-Kitamachi, Koganei, Tokyo 184-8795, JAPAN
{kasama, dai}@nict.go.jp

†††clwit Inc. 3-34-8-509 Shimo-Renjaku, Mitaka, Tokyo 181-0013, JAPAN
shimamura@clwit.co.jp

Abstract The number of malware on the Android OS is increasing rapidly. Unlike Windows malware, most of Android malware attempts to infect devices by disguising itself as a benign application and inducing users to install it. However, versatile infection process patterns are increasing. In fact both types of malware that attempt to infect the Android through Windows PC, and the one of the reverse direction exist. In this paper, we propose a method of identifying the traffic arising from the Android devices by using TCP/IP header information. By applying this method to the Darknet, we analyzed scanning activity of the Android devices.

1　はじめに

マルウェアを利用したサイバー攻撃による被害が社会的問題となっている．その中でも，Android OS[1]を採用したスマートフォン，家電などの普及によって，Android OS 上で動作するマルウェア(以下 Android マルウェア)が急増し，その被害は年々増加している．従来の Windows PC 上で動作するマルウェア（以下 PC マルウェア)の多くは，リモートエクスプロイト攻撃等の手段で脆弱性を悪用して感染を拡げるが，Android マルウェアの場合，正規のアプリケーションに偽装してアプリケーションストアに登録され，利用者がマルウェアと気づかぬままインストールを行うことで感染するケースが多い．このように主な感染形態やプラットフォームの差異によって，従来，PC マルウェアと Android マルウェアの間には特に関係性は見られなかった．しかし，近年 Windows PC から Android 端末へ感染を試みるマルウェア[2]や，Android 端末から Windows PC への感染を試みるマルウェア[3]の存在が報告されており，Android 端末から Windows PC に対してリモートエクスプロイト攻撃が行える可能性が報告されている[4]など，両者の垣根は無くなりつつある．そのため，今後は，Android マルウェアからの通信トラフィックを観測・解析することにより，Android マルウェアの感染活動や発現動向を把握することができる可能性がある．

そこで本稿では，TCP/IP ヘッダの特徴から Android OS を用いた通信トラフィックを識別する手法を提案する．本手法では，先ず日本，中国，韓国，アメリカ，インド，台湾の 6 カ国の移動体通信事業者に割り当てられた IP アドレスを基にトラフィックデータを抽出し，Passive OS Finger Printing ツールである p0f[5]を活用することによって OS 推定を行い，Linux と判断されたトラフィックを抽出する．さらに p0f によって Linux と判定されたトラフィックから，通常の Linux PC と Android のトラフィックを識別可能な特徴として，TCP ヘッダの Window サイズ，TCP ヘッダオプションの Window スケール，そしてタイムスタンプの更新周期といった 3 つの特徴を選択し，機械学習を用いて Android OS のトラフィックを分類する．本手法をダークネットに適用し，ダークネットにおいて観測された Android マルウェア活動の分析結果を述べる．

本稿の構成は以下の通りである．まず 2 章でダークネットの現状を述べ，3 章で関連研究を述べた後，4 章にて提案手法の具体的手順を述べる．5 章では提案手法をダークネットに適用した場合の分析結果を述べる．最後に，まとめを 5 章に述べる．

2　ダークネット

ダークネットは IP アドレス空間において到達可能かつ未使用の IP アドレス帯を指す．ダークネットの概要を図 1 に示す．ダークネット観測では通常，到達するパケットに対して応答を返さないブラックホールセンサが用いられるため，通常のライブネットの通信とは異なり，インターネットからダークネット宛ての片方向の通信しか観測されないのが特徴である．ダークネットに届くパケットは基本的には異常通信によるものであるため，ダークネットに到達する通信を観測・分析することによりマルウェアの感染活動や動向が把握できる．ダークネットに届くパケットの要因は主に以下の 4 つが挙げられる．

(1)　マルウェアによるポートスキャン

(2)　DDoS 攻撃のバックスキャタ

(3)　ネットワーク利用者のアドレス設定ミス

(4)　リフレクション型攻撃の準備活動

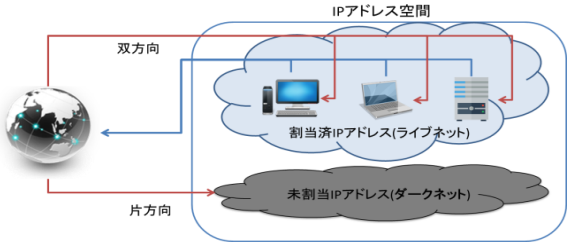

図 1：ダークネットの概要

3 関連研究

笹生ら[6]はダークネットトラフィックからマルウェアの動向や特徴など詳細情報を得るために p0f を活用し，OS 毎のパケット数の時間的推移を調査した．この結果，iPhone や iPad（Apple 社モバイル端末）からダークネットへの通信が増加していることを報告している．これらの結果から，近年，普及が著しいスマートフォンやタブレットなどのモバイル端末を対象として感染するマルウェアが，社会的問題となる危険性が高いと推測できる．Android に関わるマルウェアの急増が報告されているため，当該の感染活動を早急に把握する必要がある．

森ら[4]は PC マルウェアにおける脅威であるリモートエクスプロイト攻撃が Android 端末から行える可能性を調査するため，世界 14 カ国 20 キャリアのネットワークのポートの解放状況を調査した．この結果，一部のキャリアネットワークにおいて，Windows PC に対して脆弱性のあるポートが解放されており，Android 端末からリモートエクスプロイト攻撃が可能であることを報告している．以上述べたことから，Android マルウェアが Windows PC に対して感染活動を行い，感染が拡大する危険性があることが推測できる．しかしながら，現時点では実際に Android 端末からのスキャン活動が行われているのかどうかの分析は行われていない状況である．

4 提案手法

本章では，ダークネットトラフィックから Android 端末による通信を識別する手法を示す．まず，4.1 節において，既存手法の問題点を示し，その後，提案手法の流れを説明する．

4.1 Android パケット抽出の問題点

観測されたトラフィックデータから Android 端末による送信パケットを推定する手法として，HTTP リクエストのユーザエージェントヘッダを活用した推定方法がある．しかしながら，ダークネット観測では通常，到達するパケットに対して応答を返さないため TCP コネクションが確立されない．よって，上記の手法を用いることは困難であるため，レイヤ 4 以下の情報を用いて Android パケットを推定する必要がある．一方，p0f を用いれば TCP の SYN パケットのみで OS の推定が可能であるが，デフォルトのシグネチャリストには，Android を識別するシグネチャは存在せず，Android 端末はカーネルに Linux を使用していることから Linux と判定されてしまう．

そこで本稿では，ダークネットトラフィックから Android 端末による通信を識別するための手法を提案する．提案手法では，TCP SYN パケットを対象にした p0f による OS 推定と，4.3 節に示す TCP/IP ヘッダの特徴を基にした機械学習による Linux と Android のパケット識別の組み合わせによって Android からの通信の識別を実現する．また，本稿では，効率的に Android 端末の通信をフィルタリングするために，4.2 節で示すモバイルネットワークの IP アドレスによるフィルタリングを行う．

4.2 モバイルネットワーク IP アドレスによるフィルタリング

Android OS が搭載された端末は，スマートフォンとタブレットが大多数を占める．スマートフォンとタブレットは，Wi-Fi による通信か，または携帯電話事業者のネットワークによる通信を行っている．そこで本稿の実験では効率的に Android 端末の通信を抽出するために，移動体通信事業者が使用している IP アドレスを使用してトラフィックのフィルタリングを行う．フィルタリングに使用した移動体通信事業者は世界 6 ヶ国で 22 キャリアである．日本国内のキャリアは，ホームページで開示している公開[7-11]IP アドレス範囲を使用した．海外のキャリアは公開 IP アドレスの情報が無かったため，IP アドレスの whois 情報を公開している myip.ms[12]を使用し，IP アドレスを保有する組織を基に IP アドレス範囲を限定してフィルタリングを行った．フィルタリングに使用した各国のキャリアと IP アドレス数を表 1 に示す．

表 1：各国のキャリアと IP アドレス数

国名	キャリア名	IPアドレス数
日本	docomo	219,796
	au	720,982
	softbank	295,982
	willcom	60,070
	emobile	30
中国	ChinaMobile	35,160,030
	ChinaUnicom	36,244,164
韓国	LGTelecom	401,402
	SkTelecom	7,734,430
アメリカ	AT&TMobility	8,929,252
	VerizonWireless	28,033,462
	Sprint	18,747,350
	T-MobileUsa	12,697,578
インド	Aircel	1,159,164
	BhartiAirtel	9,366,721
	IdeaCellular	728,978
	RelianceCommunications	1,179,644
	TataTeleservice	669,140
	VodafoneEssar	358,620
台湾	ChunghwaTelecom	2,191,342
	FarEastone	2,105,324
	TaiwanMobile	622,580
合計		167,626,041

4.3 Passive OS Fingerprinting による Linux パケット抽出

モバイルネットワークからダークネットに届くパケットを抽出した後に，p0f を適用して送信元の OS が Linux と判定したパケットのみを抽出する．この手法により，テザリング経由やモバイルルータ経由でダークネットに到達する Windows PC や Apple 社端末等からの通信をフィルタリングすることが出来る．

4.4 サポートベクターマシンによる Android と Linux のパケット識別

4.1 節で述べたように，p0f では Android も Linux と誤判定される可能性がある．そこで，4.2 節で述べた p0f で Linux と推定されたパケットに対して，教師あり学習のサポートベクターマシン（以下 SVM)を活用し，Android と Linux の 2 値分類を行う．この結果 Android と分類されたパケットを最終的に Android からの通信と見なして抽出する．分類に活用できる特徴を見出すため，Android と Linux の通信をキャプチャし，TCP/IP ヘッダ情報を比較した．調査結果を表 2 に示す．表 2 から，調査した大半の Android 端末において，Window サイズの値が 14600 だったのに対し，Linux では Window サイズが 5840, 14600, 29200 など様々な値が見られた．また，Window スケールの値においても，Android 端末では 6 が大半であったのに対し，Linux では 7 が大半であった．よって，Android と Linux を識別出来る特徴として活用した．加えて，ネットワークスキャンツール Nmap[13]による Active OS Fingerprinting のテストの一つで，パケットのタイムスタンプ更新周期が用いられている点に着目し，Android と Linux のタイムスタンプの更新周期に関して調査した．その結果，表 2 より Android 端末の 9 割が 10ms であったのに対し，Linux は 1ms あるいは 4ms であった．これは，タイムスタンプの更新周期が OS のディストリビューションごとに異なる値が設定されているからだと推測される．よって，タイムスタンプの更新周期の差異は Android と Linux の識別に有効であると判断し，これも特徴として活用した．

表 2 : Linux と Android の通信調査結果

Linux				Android			
OS	windows size	windows scale	timestamp 周期(ms)	OS	windows size	windows scale	timestamp 周期(ms)
CentOS5.0	5840	7	1	Android2.2.1	5840	1	10
CentOS5.8	5840	7	1	Android2.3.3	5840	3	1
CentOS5.9	5840	7	1	Android2.3.3	5840	1	10
CentOS5.10	5840	7	1	Android2.3.7	5840	2	10
CentOS6.0	5840	6	1	Android3.2.1	5840	6	10
CentOS6.1	5840	7	1	Android4.0.4	14600	6	10
CentOS6.2	14600	7	1	Android4.1.2	14600	6	10
CentOS6.3	14600	7	1	Android4.1.2	14600	6	10
CentOS6.4	14600	7	1	Android4.1.2	65535	6	10
CentOS6.5 (kernel 2.6.32)	14600	7	1	Android4.2.2	5840	3	1
CentOS6.5 (kernrl 2.6.36.3)	5840	7	1	Android4.2.2	5840	3	1
CentOS6.5 (kernel 3.0.31)	14600	6	1	Android4.2.2	14600	6	10
CentOS6.5 (kernel 3.1.10)	14600	6	1	Android4.2.2	14600	6	10
CentOS6.5 (kernel 3.4.1)	14600	7	1	Android4.2.2	14600	6	10
CentOS7.0	14600	7	1	Android4.2.2	14600	6	10
Debian5.10	5840	7	4	Android4.2.2	14600	6	10
Debian6.10	5840	7	4	Android4.2.2	14600	6	10
Debian7	14600	3	4	Android4.2.2	14600	6	10
Fedora15	14600	7	1	Android4.2.2	14600	6	10
Fedora16	14600	5	1	Android4.2.2	14600	6	10
Fedora17	14600	7	1	Android4.2.2	14600	6	10
Fedora18	29200	7	1	Android4.2.2	14600	6	10
Fedora19	29200	7	1	Android4.2.2	14600	3	1
Fedora20	29200	7	1	Android4.3	14600	6	10
Linux Mint 11	14600	7	10	Android4.4.2	13800	7	10
Linux Mint 12	14600	5	4	Android4.4.2	13800	5	10
Linux Mint 13	14600	5	4	Android4.4.2	13820	6	10
Linux Mint 14	14600	7	4	Android4.4.2	14600	7	10
Linux Mint 15	14600	7	4	Android4.4.2	14600	6	10
Linux Mint 16	14600	7	4	Android4.4.2	14600	6	10
Linux Mint 17	14600	7	4	Android4.4.2	14600	6	10
OpenSUSE11.4	4380	7	1	Android4.4.2	14600	6	10
OpenSUSE12.3	14600	7	1	Android4.4.2	14600	6	10
OpenSUSE13.1	29200	7	1	Android4.4.2	14600	6	10
Ubuntu10.04	5840	6	10	Android4.4.2	14600	6	10
Ubuntu12.04	14600	7	4	Android4.4.2	14600	6	10
Ubuntu14.04	29200	7	4	Android4.4.2	14600	6	10
Vine Linux5.2	5840	7	4	Android4.4.2	14600	6	10
Vine Linux6.0	5840	7	1	Android4.4.3	14600	6	10
Vine Linux6.1	14600	5	1	Android4.4.4	65535	6	10
Vine Linux6.2.1	14600	7	1				

5　提案手法の適用結果と考察

本手法を適用するダークネットトラフィックデータとして，MWS2014 Datasets 2014[14]より提供された NICTER Darknet Dataset 2014 [15]を活用した．本データセットは，NICT が開発した NICTER[16]における，/20 の連続したダークネット空間で観測した pcap 形式のトラフィックデータである．本実験では，2011 年 4 月 1 日から 2014 年 7 月 31 日までのトラフィックを対象とした．実験概要を図 2 に示す．識別器は LIBSVM[17]で実装した．

識別器作成に必要な教師用データは，Android と Linux の通信データを独自に取得し，Window サイズ，Window スケールの値を抽出し，SYN パケットの送信間隔から更新周期を算出した値の計 3 つを特徴としたパターンデータ作成した．Linux は仮想マシンにイントールして，SYN パケットを収集し，合計 41 種類のパターンデータを用意した．Android は，Android 端末の実機による SYN パケットと，研究室のホームページにアクセスがあった Android 端末の SYN パケットを HTTP ヘッダのユーザエージェントを基に抽出し，合計 40 機種のパターンデータを用意した．Android と Linux で合計 81 種類の教師用データを使用し，識別器を作成した．作成した識別器に対して，教師用のデータを評価データとして活用し，n=5 の交差検定を行い，特徴の有効性と識別器の精度を評価した．結果，識別精度は 96.2963%となり，高い精度で Android と Linux を識別できることを確認できた．なお，パラメータについては，LIBSVM 付属のツールを使用し，適宜適切な値を設定した．

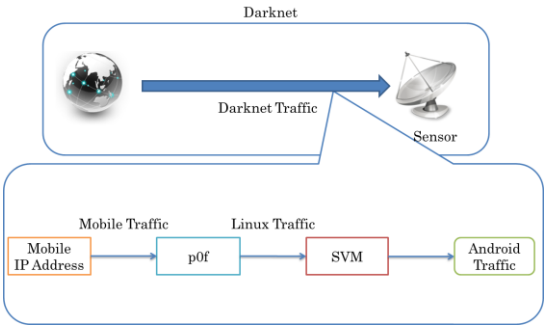

図 2：実験概要図

表 1 の 6 カ国全体から届いたダークネットトラフィックに対して各提案手法を適用した際のパケット数及びホスト数の推移を表 3 に示す．6 カ国全体から届いた Android パケット数及びユニークホスト数の推移を図 3 に示す．表 1 における国別の Android パケット数及びユニークホスト数の推移を図 4～図 9 に示す．Android パケットの宛先ポート番号の中でトップ 10 の宛先ポート番号のパケット数の推移を図 10 に示す．

表 3 より，モバイルネットワーク IP アドレスによるフィルタリングでパケット数を大幅に削減することができたため，一定の効果があると考えられる．Android からの通信については図 3 より，2012 年の 5 月から 9 月，2014 年の 1 月から 7 月までの期間において，パケット数とホスト数の増加が観測され，Android からの通信の変化を観測することができた．

6 カ国別の Android からの通信については図 4~図 9 より，全ての国で Android からの通信がダークネットへ届いていることが確認できた．国別では，中国からの Android 通信が最も多かった．これは，中国の携帯電話事業者 ChinaMobile の契約者数が 7 億人以上（世界最大）であり，Android マルウェアによるボットネットの形成が報告されている[18]ことから，中国からの Android の通信が最も多く観測されたと推定できる．インドからの Android 通信は中国に次いで多く，中国の Android 通信の推移と同じような傾向が確認できた．日本からの Android 通信は最大で 14 パケット，ホスト数も最大で 2 と非常に少なく，他の国々とも違った通信傾向が確認された．台湾，アメリカからの Android 通信では，両者において同じようなパケット数，ホスト数の推移が現れている事が確認された．韓国からの Android 通信では，2011 年の 2 月に 1 つのホストから 1000 パケット以上の Android 通信が観測されて以降，Android からの通信は少なかった．

また，図 10 に示すように，2014 年 1 月から TCP Port23 番(Telnet)と，TCP Port5000 番を宛先ポートとしたパケット数が急増した傾向を確認できた．この現象は，JPCERT/CC の定点観

測レポート[19]において telnet のポートが解放されているネットワーク機器を対象とした探索活動と，脆弱性がある NAS が使用する TCP Port5000 番に対するスキャンパケット数の増加があったと報告されていることと符合する．このような探索活動のパケットが Android 通信でも観測されたことから，Android のボットネット等によるスキャン活動が行われている可能性があったと考えられる．

表 3： 各提案手法適用後の
パケット数とホスト数

	パケット数	ホスト数
Darknet	267,475,999	6,847,666
Mobile IP Filtering	24,301,656	670,652
Linux(p0f)	6,420,290	180,328
Android(SVM)	3,809,641	165,245

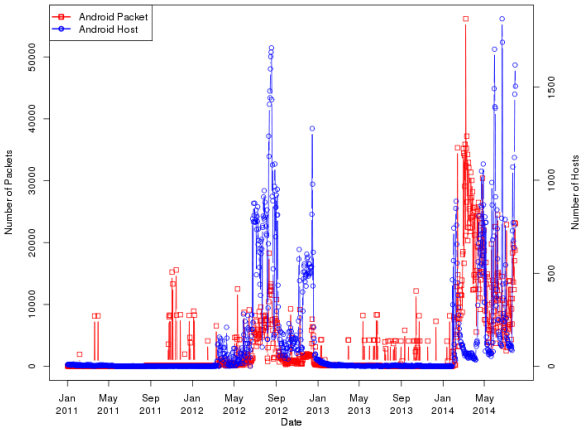

図 3： 全体の Android パケット数と
ホスト数の推移

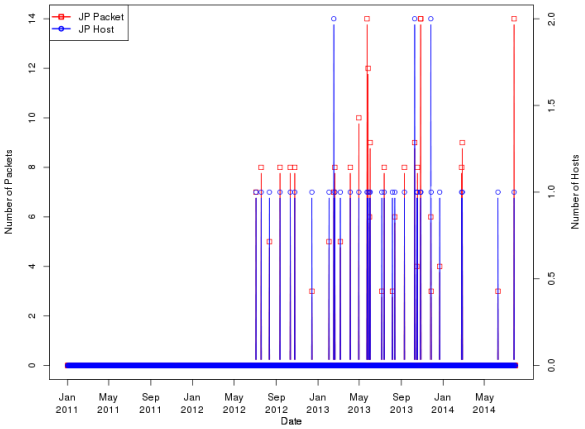

図 4： 日本からの Android パケット数と
ホスト数の推移

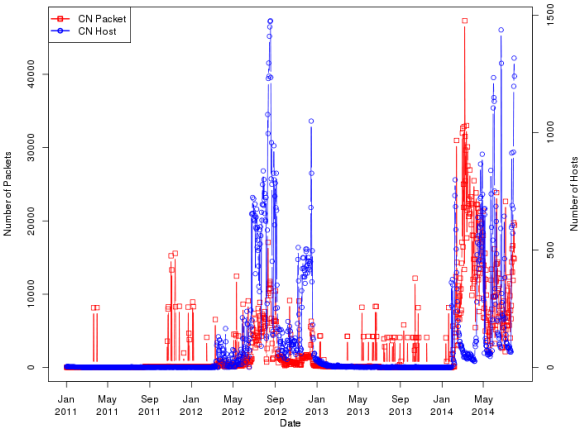

図 5： 中国からの Android パケット数と
ホスト数の推移

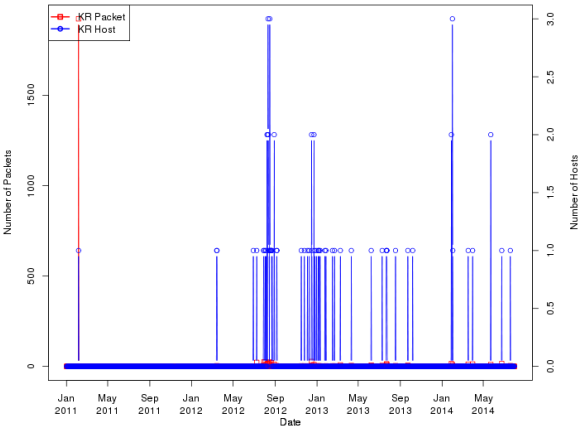

図 6： 韓国からの Android パケット数と
ホスト数の推移

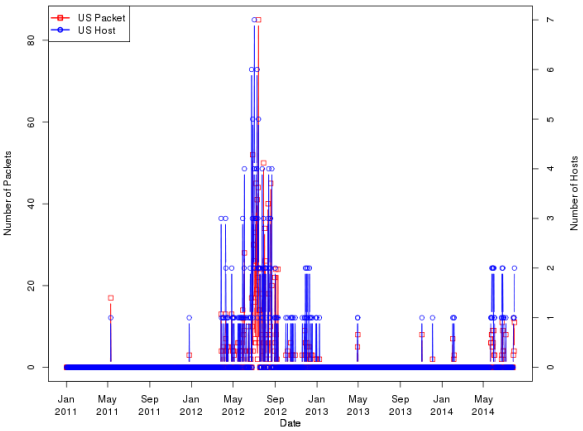

図 7： アメリカからの Android パケット数と
ホスト数の推移

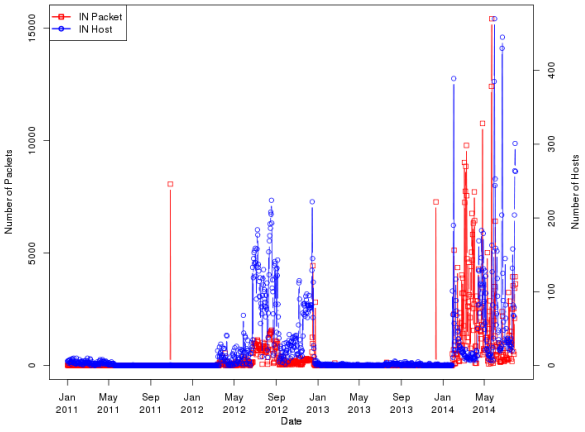

図 8：インドからの Android パケット数とホスト数の推移

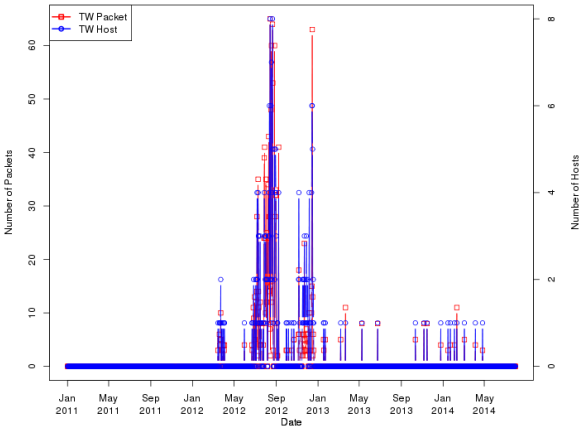

図 9：台湾からの Android パケット数とホスト数の推移

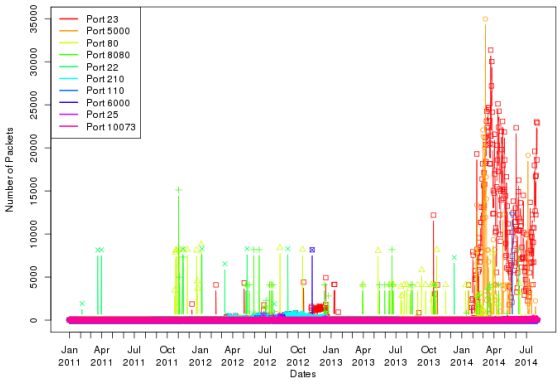

図 10 : Android からの通信における上位 10 個の宛先ポート番号の推移

6　おわりに

本稿では，ダークネット観測データから Android 端末のトラフィックを抽出する手法について提案し，当該手法をダークネットトラフィックに適用した．その結果，Android からダークネットへの通信が世界各国から行われていることが確認できた．また，Android から TCP Port23 番宛の通信が観測されているなど，何らかのスキャン行為が行われていることを観測することができた．しかしながら，どの Android マルウェアによる探索活動なのかを推定するところまでは出来ていない．

よって，今後は Android マルウェアの動的解析結果と観測パケットを突合させ，マルウェアの名称の推定を行う予定である．

参考文献

[1] Android: “Android”, http://www.android.com/

[2] Symantec, “Trojan.Droidpak”, http://www.symantec.com/ja/jp/security_response/writeup.jsp?docid=2014-012109-2723-99&tabid=2

[3] Symantec, “Android.Claco”, http://www.symantec.com/ja/jp/security_response/writeup.jsp?docid=2013-020415-5600-99&tabid=2

[4]森博志，金井文宏，庄田祐樹，吉岡克成，松本勉, "Android 携帯によるリモートエクスプロイト攻撃の可能性"，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 112(499), 43-48, 2013-03-18

[5] p0f :”p0fv3”, http://lcamtuf.coredump.cx/p0f3/

[6]笹生憲，森達哉，後藤滋樹，“通信源ホストの分類を利用したダークネット通信解析”，コンピュータセキュリティシンポジウム 2013 論文集, pp.729-736,2013-4

[7]NTT docomo: “sp モードサーバ情報”, https://www.nttdocomo.co.jp/service/developer/smart_phone/spmode/

[8]au: “ネットワークの接続条件”,
http://www.au.kddi.com/developer/android/kaihatsu/network/
[9]SoftBank: “Web 技術情報”,
https://www.support.softbankmobile.co.jp/partner/home_tech1/
[10]WILLCOM: “ウィルコムのセンター情報”,
http://www.willcom-inc.com/ja/service/contents_service/create/center_info/
[11]EMOBILE: “IP アドレス帯域 – 技術情報”,
http://developer.emnet.ne.jp/ipaddress.html
[12]myip.ms 外国のキャリアの公開 IP アドレス
[13]Nmap: “Nmap - Free Security Scanner For Network Exploration & Security Audits”,
http://nmap.org/
[14] 秋山満昭，神薗雅紀，松木隆宏，畑田光弘, "マルウェア対策のための研究用データセット～MWS Datasets 2014～", 情報処理学会 研究報告コンピュータセキュリティ（CSEC） Vol. 2014-CSEC-66, No. 19, pp. 1 - 7, 2014.
[15] 笠間貴弘，神薗雅紀, "NICTER Darknet Dataset 2014 / NONSTOP",
http://www.iwsec.org/mws/2014/files/NICTER_Darknet_Dataset_2014.pdf
[16] D. Inoue, etc., “nicter: An incident analysis system toward binding network monitoring with malware analysis” in WOMBAT Workshop on Information Security Threats Data Collection and Sharing, pp. 58-66, IEEE, 2008
[17] LIBSVM: “LIBSVM -- A Library for Support Vector Machines”,
http://www.csie.ntu.edu.tw/~cjlin/libsvm/
[18] Symantec,” MDK: The Largest Mobile Botnet in China”
http://www.symantec.com/connect/blogs/mdk-largest-mobile-botnet-china
[19] JPCERT/CC, “インターネット定点観測レポート(1～3 月) ”,
https://www.jpcert.or.jp/tsubame/report/report201401-03.html