通信プロトコルのヘッダの特徴に基づく不正通信の検知・分類手法

小出 駿† 鈴木 将吾† 牧田 大佑†‡ 村上 洸介* 笠間 貴弘‡
島村 隼平§ 衛藤 将史‡ 井上 大介‡ 吉岡 克成†‡ 松本 勉†

† 横浜国立大学 240-8501 神奈川県横浜市 保土ケ谷区常盤台 79-1
{koide-takashi-mx, suzuki-shogo-mb}@ynu.jp, {yoshioka, tsutomu}@ynu.ac.jp
‡ 情報通信研究機構 184-8795 東京都小金井市貫井北町 4-2-1
{makita, kasama, eto, dai}@nict.go.jp
* KDDI 株式会社 163-8003 東京都新宿区 西新宿 2-3-2KDDI ビル
ko-murakami@kddi.com
§ 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
shimamura@clwit.co.jp

あらまし OSの機能を使わずに独自のネットワークスタックを用いた通信を行うマルウェアやツールはTCP/IPヘッダやアプリケーションプロトコルヘッダに固有の特徴を持つ場合がある. 本稿では, TCP初期シーケンス番号, IPヘッダのID値, DNSヘッダのIDなどに固有値が設定されている通信パケットを抽出することで, ネットワーク上で観測される通信を分類する手法を提案する. ダークネット・ハニーポット観測とマルウェア動的解析によって得られた通信の分析に提案手法を適用することで, マルウェアやツールによる不正な通信の特定が可能であることを確認し, 新規のマルウェア発見にも応用できることを示す.

Detection and Classification Method for Malicious Packets with Characteristic Network Protocol Header

Takashi Koide† Shogo Suzuki† Daisuke Makita†‡ Kosuke Murakami*
Takahiro Kasama‡ Jumpei Shimamura§ Masashi Eto‡ Daisuke Inoue‡
Katsunari Yoshioka†‡ Tsutomu Matsumoto†

† Yokohama National University
79-1 Tokiwadai, Hodogaya-ku, Yokohama-shi, Kanagawa, 240-8501, Japan
{koide-takashi-mx, suzuki-shogo-mb}@ynu.jp, {yoshioka, tsutomu}@ynu.ac.jp
‡ National Institute of Information and Communications Technology
4-2-1, Nukui-Kitamachi, Koganei-shi, Tokyo, 184-8795, Japan
{makita, kasama, eto, dai}@nict.go.jp
* KDDI Corporation
KDDI Bldg. 2-3-2 Nishishinjuku-ku, Tokyo, 163-8003, Japan
ko-murakami@kddi.com
§ clwit Inc.
3−34−8 Shimorenjaku, Mitaka, Tokyo, 181-0013, Japan
shimamura@clwit.co.jp

Abstract Since some malware and network tools have their own implementation of network stack, the packets from them may have characteristic TCP/IP headers and application protocol headers. In this paper, we propose a technique for packet classification by generating signatures using initial sequence number in the TCP header, identification in the IP header, ID in the DNS header and so on. By analyzing darknet traffic, honeypot traffic, and packets from malware sandbox analysis with this method, we show that it is possible to identify packets from these software and possibly detect new malware.

1 はじめに

パケット生成の高速化，効率化を重視して作成されたマルウェアやネットワークスキャンツールは，OS の通信機能を使用せずに独自に実装されたネットワークスタックによる通信を行うことがある．これまで，このような独自の通信機能により生成されたパケットを OS フィンガープリントツールである p0f[1]を用いて分類する研究[2]や IP ヘッダの TTL (Time to live)値に着目して悪性判定を行う研究[3]が行われているが，これらに加えて TCP ヘッダの初期シーケンス番号，IP ヘッダの ID フィールド，送信元ポート番号などの固有値に着目することで，パケットの発生源となったマルウェアやツールをより高い精度で特定できる可能性がある．

そこで本研究では，ネットワーク観測やマルウェア動的解析，攻撃ツールの動的解析によって得られたトラフィックから，TCP/IP ヘッダや DNS ヘッダのパターンに特徴を持つパケットを抽出しシグネチャとすることで，送信元のマルウェアやツールを特定する手法を提案する．本手法はネットワークを観測できる中継機器，エンドポイントなどあらゆる地点で適用が可能であり，特に TCP SYN パケットのみの特徴で特定可能なシグネチャを利用すれば，3WAY ハンドシェイクを確立し攻撃を開始する前に，不正パケットの遮断を行うことができるという利点がある．さらに既知のパターンを蓄積することで新規性の高い攻撃の発見にも応用できる可能性がある．

提案手法の有効性を示すためにダークネットやハニーポットのトラフィック，マルウェア動的解析時に発生したトラフィックの分析を行った．その結果として，Windows のリモートデスクトッププロトコル (3389/tcp)を利用して感染する Morto ワームに関して，2011 年の発生時期においてダークネットの 3389/tcp 宛の通信が急増していた事実が論文[4]等で指摘されているが，当該時期に急増した通信は，提案手法による分類では 3 パターンの異なる特徴を持つ通信に分類が可能であり，そのうちの一つは，我々が所持する Morto 検体の動的解析時に観測される通信と同様の特徴を有していることが確認できた．残りの 2 パターンのうち，一方はその後も長期に渡りダークネットで観測されていることから，Morto の亜種や何らかの脆弱性攻撃ツールが生成する通信である可能性がある．もう一方は，Morto の発生時期以降はダークネットで観測されていない．このことから，当該時期に Windows のリモートデスクトップを狙った様々なマルウェアやツールの実装が試みられていたことが推察される．また，近年脅威が拡大している DRDoS (Distributed Reflection Denial of Service)攻撃を行う Iptables Backdoor[5]などの Linux マルウェアや，高速ネットワークスキャンツール Zmap などから発生したパケットの特徴的なヘッダパターンをシグネチャ化すると共に Zmap を使った通信が当該ツールの公開後に実際にダークネットで増加していることを確認した．

本論文の構成は以下の通りである．まず 2 章で関連研究について述べ，3 章で提案手法を説明する．その後，4 章で検証実験により提案手法の有効性を示し，5 章でまとめと今後の課題についてまとめる．

2 関連研究

OS によって提供されるソケット API を利用せず独自のネットワークスタックを実装したマルウェアによる通信パケットを分析するために TCP/IP ヘッダの値を用いる研究が行われている[2, 3, 6, 7]．p0f は通信パケットの TCP/IP ヘッダの値から受動的に送信元の OS を判定する TCP フィンガープリント技術を用いたツールであり，このツールを用いてハニーポットなどへの攻撃通信を分析し，OS 判定の出来なかったパケットを独自に作成した p0f のシグネチャとして検知を行う研究が行われている[2]．さらに，文献[6]では新規に作成した p0f のシグネチャをもとに観測した通信を分析し，スパムメール送信元 IP アドレスのブラックリストと照合することで，独自のパケット生成によって SMTP 通信を行うマルウェア Srizbi botnet を特定している．また，IP ヘッダの TTL 値に着目し，異常に大きいホップ数の TTL を持つパケットを文献[2]と同様に p0f を用いた TCP フィンガープリント技術と IP アドレスのブラックリストを用いて検知する手法が提案されている[3]．しかし，これらの研究は p0f が利用するフィールドである初期 TTL，ウィンドウサイズおよび MSS (最大セグメントサイズ)などの TCP/IP ヘッダを分析するに留まっており TCP SYN パケットの初期シーケンス番号，IP ヘッダの ID 値または DNS の ID などアプリケーションプロトコルのヘッダフィールドを分析することで，送信元の通信機能の実装の違いを区別できる可能性については言及していない．文献[7]では UDP パケットについて，IP ヘッダの ID 値がパケットごとに増加される事と TTL 値の分析から，偽装されている IP アドレスを持つ一連のパケットが共通の攻撃ホストから送信されているものであると予想している．

3 提案手法

IP ヘッダの初期 TTL，TCP ヘッダのウィンドウサイズ，MSS などのフィールドは通常，OS ごとに異なる値が設定されており，パケット送信ごとに変化しない．

しかし，TCP ヘッダのシーケンス番号は通常 SYN パケット送信ごとにランダムな初期値が割り当てられ，送信元ポート番号は他のプロセスと重複しないように空いているポートから OS によって動的に選択される．また，パケット送信ごとに異なる値が設定されるフィールドとして，IP のフラグメントパケットの識別のために利用される ID 値，ICMP Echo Request ヘッダの ID，シーケンス番号などがある．以上より，これらのヘッダフィールドが常にある特定の固定値に設定されたパケットを観測できた場合，OS の通信機能を使用しない独自のネットワークスタックを用いた通信によるものと考えられ，さらに送信元の通信機能の実装の差異を識別できる可能性が高い．そこで本研究では通信パケットに対し，シグネチャを独自に生成し，パターンマッ

表 1 シグネチャのパラメータとなるヘッダフィールド

IP ヘッダ	ID (IP ID)
	フラグ, フラグメントオフセット (Flags,Offset)
	TTL
TCP ヘッダ	送信元ポート番号 (Sport)
	宛先ポート番号 (Dport)
	シーケンス番号 (Seq)
	ACK 番号 (Ack)
	ウィンドウサイズ (Win)
	オプション (Option)
UDP ヘッダ	送信元ポート番号 (SPort)
	宛先ポート番号 (DPort)
アプリケーション プロトコルヘッダ	パケット送信ごとに変化する値
ICMP Echo Request ヘッダ	ID (ICMP ID)
	シーケンス番号 (ICMP Seq)

チングを行うことで, マルウェアやツールを特定する手法を提案する.

ネットワークの観測データを用いてシグネチャを生成する場合, まず通信パケットごとに通信プロトコルに合わせて表 1 のヘッダフィールドの値を抽出しヘッダパターンとする. 得られた複数のヘッダパターンから共通の特徴を持つ独自実装によるパケットをパケット数の閾値Th_p, 送信元 IP アドレス数の閾値Th_sを用いて分類し, シグネチャを生成する. シグネチャは表 1 のヘッダフィールドをパラメータとして持ち, それぞれの設定値は単一の値, 複数値, ワイルドカード*または値無しとなる. ただし, TTL は OS ごとに設定される異なる初期値の大半は 64,128,255 の 3 種であるため[3], 観測された TTL 値以上の最も近い 3 種のうち 1 つを初期 TTL 値と予想して設定する.

TCP SYN パケットのシグネチャ生成は,

- (シーケンス番号, IP ヘッダの ID 値)
- (シーケンス番号, 送信元ポート番号)
- (IP ヘッダの ID 値, 送信元ポート番号)

の 3 種の組についてヘッダフィールドの固定値による組み合わせを基にシグネチャを生成する. シグネチャ生成手順を以下に示す.

Step 1)表 1 の IP ヘッダ, TCP ヘッダの 9 つのフィールドの値をパケットごとに抽出しヘッダパターンとする. 観測データの規模に応じてパケット数の閾値Th_p, 送信元 IP アドレスの閾値Th_sを設定する.

Step 2)上記の 3 種の組を基にヘッダパターンを調べる.

2-1)シーケンス番号と IP ヘッダの ID 値が同じヘッダパターンを持つパケット群のパケット数と送信元 IP アドレス数を調べ, それぞれ閾値Th_p, Th_s以上の場合, その組み合わせ持つヘッダパターン全てをシグネチャ候補とする.

2-2)シーケンス番号と送信元ポート番号が同じヘッダパターンを持つパケット群について 2-1 と同様の閾値判定を行い, シグネチャ候補とする.

2-3)IP ヘッダの ID 値と送信元ポート番号が同じヘッダパターンを持つパケット群について 2-1 と同様の閾値判定を行い, シグネチャ候補とする.

Step 3)Step2 でシグネチャ候補となった 2 値が同一であるがその他のフィールドが異なるヘッダパターンをまとめてシグネチャとするために, 2 値以外の各フィールドについて, 単一の値であればその値をシグネチャパラメータ値とする. また, ヘッダパターンごとに値が異なるとき, 特定の範囲やいくつかの値にのみパケット数が集中している場合, 設定された値のパターンから選択されていると判断し複数値をシグネチャパラメータ値とする. それ以外の場合はランダムによる設定や任意の値設定と判断し, ワイルドカード*をシグネチャパラメータ値とする. 最後に, オプション値が存在しない場合は値無しとして, シグネチャを生成する. 現在は上記のように属性値が1つの固定値とならない場合については手動で判断しているが, 今後は観測された属性値の分布から自動的に上記のケースを判別する手法を検討したい.

UDP, ICMP パケットのシグネチャ生成は下記のプロトコルごとのヘッダフィールドの組を基に以下に示す手順で行う.

UDP ヘッダフィールドの組

- (IP ヘッダの ID 値, 送信元ポート番号)
- (IP ヘッダの ID 値, アプリケーションプロトコルヘッダの値)
- (送信元ポート番号, アプリケーションプロトコルヘッダの値)

ICMP Echo Request ヘッダフィールドの組

- (IP ヘッダの ID 値, ICMP ヘッダの ID 値),
- (IP ヘッダの ID 値, ICMP ヘッダのシーケンス番号),
- (ICMP ヘッダの ID 値, ICMP ヘッダのシーケンス番号)

Step 1)プロトコルごとに表 1 のヘッダフィールドの値を抽出し, ヘッダパターンとする. 観測データの規模に応じてパケット数の閾値Th_p, 送信元 IP アドレスの閾値Th_sを設定する.

Step 2)TCP SYN パケットのシグネチャ生成の Step2 と同様にプロトコルごとにヘッダフィールドの組が同じヘッダパターンを持つパケット群を閾値判定することでシグネチャ候補を得る.

Step 3)シグネチャ候補のヘッダパターンに対し, TCP SYN パケットのシグネチャ生成の Step3 と同様に 2 値以外のフィールドの値を設定し, シグネチャとする.

生成したシグネチャによるパケットのパターンマッチングは, シグネチャのパラメータと照合し, 全て一致したものを検出する. また, 送信元 IP アドレスを詐称することによって跳ね返りのパケットとして観測されるバックスキャタは, 以下に述べる方法でシグネチャを変換し, パターンマッチングを行う. TCP SYN パケットのバックスキャタとなる SYN ACK パケットの ACK 番号と宛先ポート番号は, 受信した SYN パケットのシーケンス番号に 1 を足した値と送信元ポート番号がそれぞれ設定され, DNS パケットのバックスキャタは, DNS ヘッダの ID 値が同じ値であり, TCP と同様に UDP ヘッダの送信元ポート番号と宛先ポート番号が入れ替わる. この事を利用し, ミクロ解析で得られた TCP SYN パケットや DNS パケットのシグネチャに上記のパラメータの処理を行い, それ以外のパラメータにワイルドカード*を設定することでバックスキャタ用シグネチャとする.

表 2 Morto 検体から発生したグローバルアドレスへの 3389/tcp 宛スキャンパケットのシグネチャ

シグネチャ名	IP ID	Flags, offset	TTL	Seq	Ack	SPort	DPort	Win	Option
Morto_scan	9496	DF,0	64	2406000322	0	4935	3389	65535	mss1240,nop,ws0,nop,nop,sacOK

表 3 ダークネットで観測された 3 パターンの 3389/tcp 宛パケットのシグネチャ

シグネチャ名	IP ID	Flags,offset	TTL	Seq	Ack	SPort	DPort	Win	Option
Morto_scan	9496	DF,0	64	2406000322	0	4935	3389	65535	mss1240,nop,ws0,nop,nop,sacOK
Dark_Dst3389_1	256	0,0	128	1210253312	0	6000	3389	16384	無し
Dark_Dst3389_2	256	DF,0	128	2284205602	0	*	3389	512	無し

4 検証実験

本章では，独自実装の通信機能によって生成されたパケットについて，マルウェア動的解析やツールの動的解析によるミクロ解析と，ダークネットトラフィックの分析によるマクロ解析を行い，解析結果の相関分析から提案手法の有効性を示す．本研究で分析対象とするダークネットトラフィックは，NITCER[8]のNONSTOP[9]で提供されている/16 のネットワークを観測したものであり，このダークネットセンサは到達した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である．また，DNS ハニーポットのトラフィックは論文[10]で分析対象としたものと同様のオープンリゾルバとして動作する DNS ハニーポットで観測したものである．

4.1 Morto 関連通信の分析

Morto はリモートデスクトッププロトコル（RDP)を利用して Windows 端末やサーバに感染活動を行うワームである．マシンが感染すると，ローカルネットワークをスキャンし，リモートデスクトップサーバを見つけると，RDP ポート（3389/tcp)に通信し「admin」「1234」などの安易なアカウント名，パスワードによるログインを試みる．またインターネット上のリモートデスクトップサーバに対して，ランダムに生成された宛先 IP アドレスに対してアクセスを行うことが知られている．このマルウェアに関して，マルウェア動的解析を行い，得られたシグネチャを用いてダークネットトラフィックの分析を行う．

Morto 検体のマルウェア動的解析を下記の条件で行った結果，特徴的な通信を観測した．この実験は論文[11]と同様のマルウェア動的解析環境を使用した．以降 Windows マルウェアの動的解析はこの環境で行う．

検体ハッシュ値（MD5)：0475c97ddb96252febff864fb778b460

解析日時：2012 年 8 月 26 日 10:52~16:54 (6 時間)

実行環境：WindowsXP SP2

発生した通信を分析すると，80，445，3389/tcp 宛のパケットがマルウェア検体から発生しており，このうち 80,445/tcp 宛の通信は OS の通信機能を使ったと推定されるパケットであった(図 1)．3389/tcp 宛の通信は OS の通信機能を使ったと推定されるパケットと，独自生成を行ったと思われるパケットの 2 種に分類することができ，WindowsXP による 3 回の SYN パケット再送処理を考慮して，閾値Th_p=4, Th_s=1 と設定し，提案手法

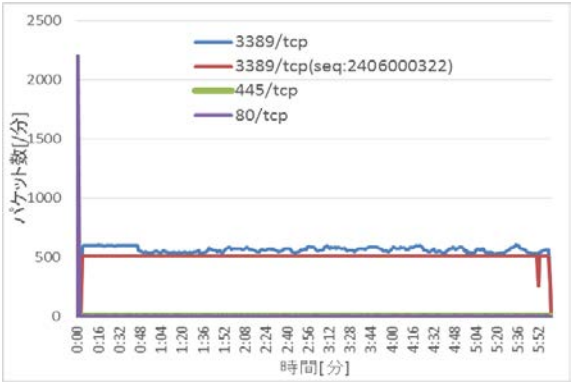

図 1 動的解析時に Morto 検体から発生した通信のポート番号別パケット数

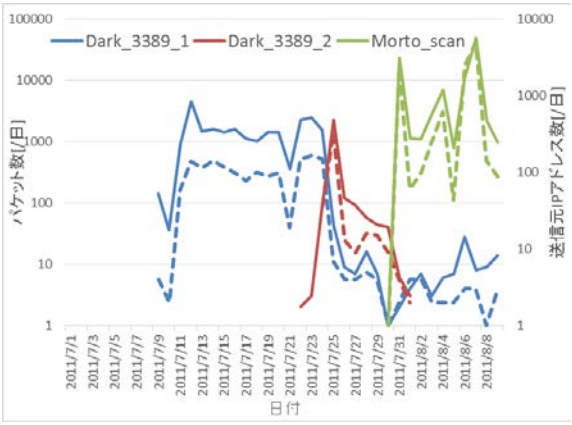

図 2 ダークネット上の 3 パターンの 3389/tcp 宛通信のパケット数（実線)・送信元 IP アドレス数（点線)

によるシグネチャ生成を行ったところ，後者のパケットのヘッダパターンをシグネチャとすることができた（表 2)．OS の通信機能による 3389/tcp 宛パケットは検体の実行環境のローカルアドレスである 192.168.228.40 に近い 192.168.226.0 から 192.168.230.254 のアドレスとグローバルアドレスに対して送信していることから，プライベートネットワークへ向けたスキャンと外部へのセッション確立を並行して行っている事が分かる．また，独自実装による 3389/tcp 宛パケットはシーケンス番号，IP ヘッダの ID 値および送信元ポート番号などに固定値が設定されており，ランダムなグローバルアドレスに対する毎分 512 回のスキャンを送信していることから，生成されたシグネチャ名を **Morto_scan** とした．当該検体に感染したホストの通信のうち，ダークネットで観測される可能性がある通信はグローバルアド

表 4 Zmap から発生したパケットのシグネチャ

(a) TCP パケットのシグネチャ

シグネチャ名	IP ID	Flags,offset	TTL	Seq	Ack	SPort	DPort	Win	Option
ZMap_tcp	54321	0,0	255	*	0	*	*	65535	無し

(b) UDP パケットのシグネチャ

シグネチャ名	IP ID	Flags,offset	TTL	SPort	DPort
ZMap_udp	54321	0,0	255	*	*

(c) ICMP Echo Request パケットのシグネチャ

シグネチャ名	IP ID	Flags,offset	TTL	ICMPID	ICMPSeq
ZMap_icmp	54321	0,0	255	*	0

レスへのスキャンであることから，この通信がダークネットで観測されているかを調べる．

Morto に対する注意喚起は Microsoft，F-Secure によって 2011 年 8 月 28 日に行われており[12]，これに近い時期を Morto 発生時期と予想し，2011 年 7 月 1 日から 2011 年 8 月 9 日までのダークネットトラフィックの 3389/tcp 宛 SYN パケットを閾値Th_p=8, Th_s=2 と設定し分析した．この結果，シーケンス番号，IP ヘッダ ID などに固定値を持つ 3 種のシグネチャを生成した（表 3）．各シグネチャによって検出された分析期間の通信の一日ごとのパケット数と送信元 IP アドレス数を図 2 に示す．シーケンス番号 2406000322 の SYN パケットは，動的解析から得られたシグネチャ **Morto_scan** のヘッダパターンと一致することが確認できた．この通信は 2011 年 7 月 30 日から発生しているため，本手法を用いることで Microsoft 等による注意喚起が行われる1ヶ月前のトラフィックに Morto によるスキャンと推定される通信が既に発生していた事が分かる．

シグネチャ **Dark_Dst3389_2** の通信が観測されたのは Morto 発生時期前後のみであることから，当該時期に試みられていたリモートデスクトップを狙った攻撃の 1 つであると考えられる．一方，**Morto_scan, Dark_Dst3389_1** のシグネチャで検出されるパケットは 2014 年 8 月現在も観測されていることから Morto の亜種やリモートデスクトップを狙う攻撃ツールの可能性がある．

4.2 ZMap 関連通信の分析

ZMap とはミシガン大学により 2013 年 8 月に公開された独自のパケット生成機能を持つ高速ネットワークスキャンツールである[13]．開発元によると実行環境によっては IPv4 の全アドレス空間を 45 分でスキャン可能といわれている．

このツールを用いて TCP，UDP，ICMP Echo Request パケットの送信を行い，Morto 検体の動的解析と同様に閾値を Th_p=4, Th_s=1 と設定し，送信されたパケットを，提案手法を用いて分析した結果，ICMP パケットのシグネチャのみ得られたが，IP ヘッダの ID 値が固定値である事に加え，IP ヘッダのフラグ・フラグメントオフセット，初期 TTL 値，TCP ヘッダのウィンドウサイズが固定値であることは/16 規模のダークネットの分析において十分に特徴的であると考え，TCP，UDP パケットのヘッダパターンもシグネチャとした（表 4 (a), (b), (c)）．ただし，ZMap はオープンソースソフトウェアであるため，ソースコードの編集によりヘッダパターンは変化し得る点に注意が必要である．

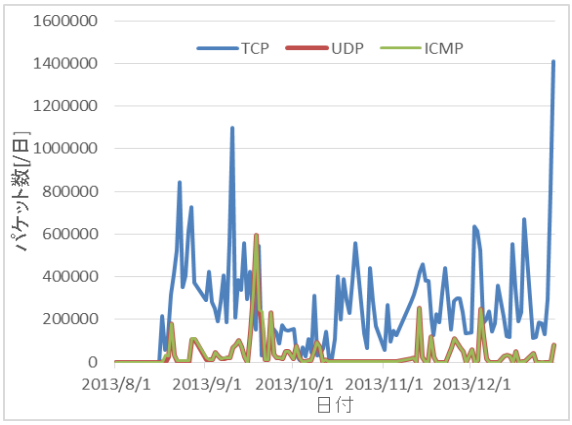

図 3 ZMap シグネチャに適合したダークネット上のパケット数

次に，ZMap の公開時期前後のダークネットトラフィックと，動的解析によって得られたシグネチャの照合を行った．検出された通信のプロトコル別のパケット数の推移を図 3 に示す．ZMap が公開された 2013 年 8 月半ばまでは ZMap のシグネチャによって検出される通信は一切見られなかったが，8 月 16 日から通信の発生とパケット数の増加を確認した．特に通信量の多い送信元アドレスを分析したところ，ある/24 ネットワークからの継続的な通信を観測した．この IP アドレスは全てミシガン大学に割り当てられているものであり，ZMap を使用した実験や評価を行っている可能性は高いと思われる．また，IPv4 Scan[14]，Project Sonar by Rapid7[15]などのスキャンやセキュリティ調査を目的としたプロジェクトによる通信も同様に確認でき，どちらも ZMap の使用に言及しているため，ZMap による通信が正しく識別されていると言える．

4.3 Iptables Backdoor 関連通信の分析

Iptables Backdoor[5]は SYN Flood，DNS DoS 攻撃を行う Linux マルウェアである．

送信元 IP アドレスを偽装し DoS 攻撃を行う機能を有していることから，近年脅威が拡大している DRDoS（Distributed Reflection Denial of Service)攻撃に使用されている可能性がある．DRDoS 攻撃とは，標的の IP アドレスを送信元 IP アドレスに詐称したパケットを生成し，送信パケットより応答パケットが大きく増幅される可能性のある DNS や NTP などのサービスを悪用することで，ペイロードサイズの大きいパケットを大量に送りつける攻撃である．

このマルウェア検体の動的解析を行い，特徴的な通信の抽出を行った．論文[16]と同様の Linux マルウェア用動的解析

表 5 TCP SYN Flood パケットのシグネチャ

シグネチャ名	IP ID	Flags,offset	TTL	Seq	Ack	SPort	DPort	Win	Option
Iptables_tcp_1	0	0,0	150~229	848	0	*	*	1600～1899	無し（ペイロード: 0x0000…:848byte）

表 6 DNS DoS パケットのシグネチャ

シグネチャ名	IP ID	TTL	SPort	DPort	DNS ID
Iptables_dns_1	*	200～215	*	53	64 種

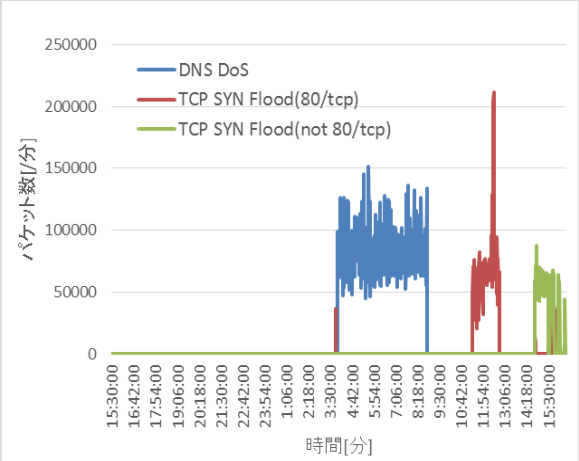

図 4 検体から発生した SYN Flood, DNS DoS 通信のパケット数

環境を用いて，以下の条件で実験を行った．

検体ハッシュ値 (MD5)：b826fb1253a52a3b53afa3b7543d7694

解析日時：2014 年 7 月 17 日 15:30~18 日 16:29 (25 時間)

実行環境：Ubuntu10.04 LTS

閾値Th_p=4, Th_s=1 と設定し提案手法に基づき検体から発生した通信を分析した結果，TCP SYN Flood 攻撃を目的としたと思われる通信からシグネチャが得られた（表 5）．このヘッダパターンはシーケンス番号，IP ヘッダ ID が固定の SYN パケットであるが，ウィンドウサイズ，初期 TTL 値が送信されるパケットごとに変化し，848byte のペイロードが付加されていることが確認できた．

DNS DoS 攻撃と思われる通信は，提案手法による精度の高いシグネチャを生成することはできなかったが，DNS ヘッダの ID は常に固定の 64 種から設定され，初期 TTL も 200~215 の範囲のランダム値という特徴が見られたため，このヘッダパターンは本研究で対象とするダークネットと DNS ハニーポットのトラフィックから検出するのに十分特徴的と考え，シグネチャとした（表 6）．TCP SYN Flood, DNS DoS 攻撃の通信量を図 4 に示す．

このマルウェアから発生した通信の相関分析を行うため，マクロ解析として，文献[5]のレポートが公開された 2014 年 7 月 16 日から 8 月 15 日までの期間にダークネットと DNS ハニーポットで観測されたトラフィックを対象に，TCP, DNS パケットのシグネチャを用いて通信の検出を試みた．また，動的解析では送信元 IP アドレスを詐称した DRDoS 攻撃のパケットを観測できなかったが，生成したシグネチャと同様のヘッダパターンをそれらが持つと仮定して，提案手法によりバックスキャタのシグネチャを作成し，同様に検出を試みた．その結果，ダークネット，DNS ハニーポット共に全期間でこのマルウェアの攻撃通信やバックスキャタと思われるパケットを検出することはできなかった．しかし，当該検体が特徴的な通信を発生することは確認できたため，ミクロ解析を継続することで DRDoS 攻撃通信の観測を行い，新たなシグネチャを生成し，マクロ解析における観測データと比較を今後行いたい．

4.4 ダークネットに到達した TCP パケットの分析

2014 年 6 日 10 日から 19 日までの 10 日間観測されたダークネットのトラフィックに対し，提案手法を用いたシグネチャ生成と通信の分析を行った．

一日ごとの TCP SYN パケットを入力とし，複数ホストから送信された同一の実装による通信を分析するため，閾値を Th_p=100, Th_s=2 と設定しシグネチャ生成を試みたところ，5 種のシグネチャを得る事ができた．Morto 検体の動的解析で得られたシグネチャ **Morto_scan** と Morto 発生時期のダークネット分析で得られたシグネチャ **Dark_Dst3389_1** で検出できる通信の他，3 種の未知のシグネチャが得られ，それぞれヘッダパターンの特徴に基づきシグネチャ名を付けた（表 7）．5 種のシグネチャで検出できる通信のうち，**Dark_Dst23_1** で検出できる通信は特に多くの送信元 IP アドレスからの大量のパケットが確認できたため，この通信の長期的な分析を行う．

4.4.1 組み込み機器を狙う攻撃

シグネチャ **Dark_Dst23_1** のヘッダパターンを持つ通信を 2014 年 1 月 1 日から 8 月 15 日のダークネットトラフィックから分析した結果，2014 年 2 月 15 日から 6 月 25 日の期間に，同様のヘッダパターンによる 23/tcp を含む 5 種の宛先ポート番号を持つ通信を確認し，そのうち IP ヘッダ ID に固定値を持たないという点のみ異なる類似のヘッダパターンの通信も確認した．この通信のパケット数と送信元 IP アドレス数の推移を図 5 に示す．ただし 4 月 26 日から 5 月 11 日の期間はセンサが停止していたためトラフィックは存在しない．5 種の宛先ポート番号のうち 32764/tcp は特定のルータに対し，メーカーが意図的に作成したバックドアのポート番号として報告されており[17]，58455/tcp はルータやモデムなどを狙ったマルウェアである Linux.Darlloz[18]が作成するバックドアであるため，このヘッダパターンを持つ通信は組み込み系 Linux を狙った攻撃のためのスキャンである可能性が高いと思われる．これらの通信をまとめてシグネチャ **Dark_Embedded_Linux_1** とする（表 8）．

表 7 ダークネットに到達したパケットのシグネチャ

シグネチャ名	IP ID	Flags, offset	TTL	Seq	Ack	SPort	DPort	Win	Option
Morto_scan	9496	DF,0	64	2406000322	0	4935	3389	65535	mss1240,nop,ws0, nop,nop,sacOK
Dark_Dst3389_1	256	0,0	128	1210253312	0	*	3389	16384	無し
Dark_Dst23_1	0	DF,0	64	1112425812	0	*	23	300	無し
Dark_IPID256_1	256	0,0	128	*	0	6000	3306,1433,80,8009,22,2222,9080	16384	無し
Dark_IPID0_1	0	DF,0	64	*	0	12200～12219	0	8192	無し

表 8 組み込み機器を狙うスキャンパケットのシグネチャ

シグネチャ名	IP ID	Flags,offset	TTL	Seq	Ack	SPort	DPort	Win	Option
Dark_Embedded_Linux_1	0	DF,0	64	1112425812	0	*	22,23,8080,32764,58455	300	無し

表 9 Srizbi 検体による C&C への通信パケットのシグネチャ

シグネチャ名	IP ID	Flags,offset	TTL	Seq	Ack	SPort	DPort	Win	Option
Srizbi_1	*	0,0	128	6509	0	48001	4099	24000	mss536

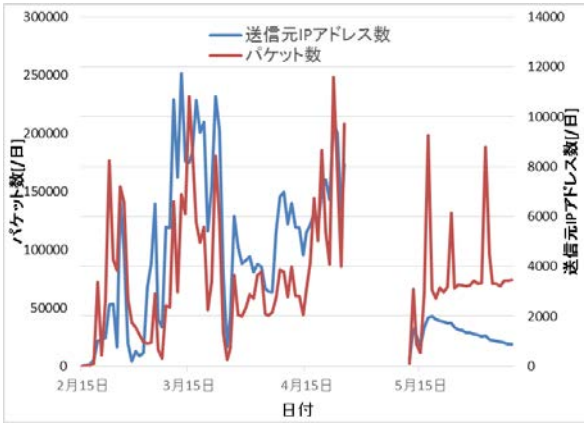

図 5 組み込み機器を狙うスキャンのパケット数と送信元 IP アドレス数

この通信の相関分析を行うため，ミクロ解析として Linux.Darlloz 検体 (MD5: 00a299fd149939cec860c71224b77209)のマルウェア動的解析を 4.3 節と同様の環境で行った．2014 年 6 月 19 日から 2014 年 7 月 17 日までの期間にこの検体を実行したが，58455/tcp 宛の通信の発生は確認できたものの，シグネチャ Dark_Embedded_Linux_1 と合致する通信は確認できなかった．この他に，数種類の linux マルウェア検体の動的解析を試み，Linux.BackDoor.Gates[19]などのマルウェアから独自実装の通信機能を用いたパケットを確認しているが，現在まで当該シグネチャで検出できる通信は確認できていない．

4.5 Srizbi botnet 関連通信の分析

独自のパケット生成機能によってスパムメールを送信するマルウェア Srizbi botnet について，発生する通信を確認するため，検体の動的解析を下記条件で実行した．

検体ハッシュ値 (MD5)：ddd86c0c74511202256807a44e26ce9d

解析日時：2013 年 12 月 24 日 15:54~16:13 (20 分)

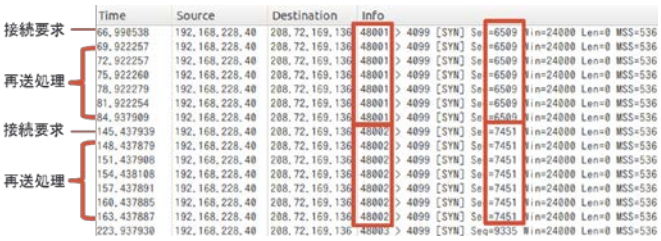

図 6 Srizbi 検体による C&C サーバへの通信

実行環境：WindowsXP SP2

TCP 通信では，宛先ポート番号 80，4099/tcp の通信が確認された．閾値を，Th_p=4, Th_s=1 と設定し提案手法に基づき分析したところ 80/tcp 宛パケットは OS (WindowsXP)の通信機能，C&C サーバの 4099/tcp へ向けたパケットは独自実装によるパケット生成であり，表 9 のヘッダパターンをもつシグネチャが得られた．WindowsXP の SYN パケット再送処理は，通常 3，6，12 秒の待ち時間によって送信されるが，4099/tcp 宛通信は 3 秒間隔で 6 回の再送パケットを送信しており，検体を実行するたびに初期シーケンス番号は 650，送信元ポート番号は 48001 から開始する (図 6)．

論文[4]で示されている p0f でシグネチャ作成が可能となるスパムメール送信パケットを観測することはできなかったが，C&C サーバとの通信に関しても特徴的な通信を行うことが確認され，シグネチャを作成することができた．当該シグネチャによる通信は C&C サーバへの接続要求という性質からダークネットに到達する可能性は低いと考えられ，実際に 4.4 節と同様の期間でダークネットトラフィックを分析した結果，この通信は確認できなかった．ダークネットを用いたマクロ解析を行うことはできなかったが，このシグネチャは，ゲートウェイ等のネットワークの中継地点での Srizbi botnet の感染ホストの特定に利用できると考えられる．

5 まとめと今後の課題

OS の通信機能を使用せずに独自のネットワークスタックによる通信を行うマルウェアやツールから発生したパケットを TCP/IP ヘッダ, アプリケーションプロトコルのヘッダに固有値が使われているといった特徴から特定する手法を提案した. 送信元IPアドレスを詐称し大量のパケットを送信するDRDoS攻撃を行うマルウェアや, 高速なスキャンを行うマルウェアやツールなどはOSに依存しない独自実装の通信機能を持つ可能性が高く, これらの実装から発生した通信の特定に提案手法が特に有効である事を, マクロ解析とミクロ解析による相関分析の結果から示した. ダークネットやハニーポットの観測データを用いたシグネチャ生成の際のネットワーク規模や観測期間の違いに合わせた閾値の設定方法, 閾値の変更に伴う生成されたシグネチャによって特定できる通信の相違に関する考察および本研究で相関分析の出来なかったシグネチャに関する調査・実験が今後の課題である.

謝辞

本研究の一部は, 総務省情報通信分野における研究開発委託／国際連携によるサイバー攻撃の予知技術の研究開発／サイバー攻撃情報とマルウェア実体の突合分析技術／類似判定に関する研究開発により行われた.

また, 本研究では, NICTER が保有しているサイバーセキュリティ情報を遠隔から安全に利用するための分析基盤(NONSTOP)にて提供されるダークネットデータを利用した. 貴重なデータセットを提供して頂いたNICTERの関係者各位に深く感謝します.

参考文献

[1] M. Zalewski, p0f v3 (version 3.07b), http://lcamtuf.coredump.cx/p0f3/(最終閲覧日:2014/8/1).
[2] 木佐森幸太, 下田晃弘, 森達哉, 後藤滋樹, “TCP フィンガープリントによる悪意のある通信の分析”, 情報処理学会論文誌, vol. 52, no. 6, pp. 2009-2018, 2011.
[3] R. Yamada and S. Goto, "Using abnormal TTL values to detect malicious IP packets", in Proceedings of the Asia-Pacific Advanced Network (APAN), ISSN 2227-3026, doi:10.7125/APAN.34.4 ,Volume 34, p. 27-34, 2013.
[4] 中里純二, 島村隼平, 衛藤将史, 井上大介, 中尾康二, “nicter によるネットワーク観測および分析レポート ～ ネットワークインシデントの前兆 ～”, 信学技報, vol. 113, no. 95, ICSS2013-14, pp. 79-84, 2013.
[5] Iptables Backdoor: Even Linux Is At Risk of Intrusion ‹ Palo Alto Networks BlogPalo Alto Networks Blog, http://researchcenter.paloaltonetworks.com/2014/07/iptables-backdoor-even-linux-risk-intrusion/(最終閲覧日:2014/8/9).
[6] DiBenedetto. S, Gadkari. K., Diel. N, Steiner. A, Massey. D, and Papadopoulos. C, “Fingerprinting custom botnet protocol stacks”, In Secure Network Protocols (NPSec),2010 6th IEEE Workshop on, pp. 61–66. IEEE, 2010.
[7] 中里純二, 島村隼平, 衛藤将史, 井上大介, “パケットヘッダの特徴に基づいたダークネットトラフィックのパケット生成手法の分類”, 信学技報, vol. 109, no. 285, ICSS2009-61, pp.43-48, 2009.
[8] NICTER, http://www.nicter.jp/(最終閲覧日:2014/8/9).
[9] 竹久達也, 井上大介, 衛藤将史, 吉岡克成, 笠間貴弘, 中里純二, 中尾康二, “サイバーセキュリティ情報遠隔分析基盤 NONSTOP ”, 信学技報, vol. 113, no. 95, ICSS2013-15, pp. 85-90, 2013.
[10] 牧田大佑, 吉岡克成, 松本勉, 中里純二, 島村隼平, 井上大介, “DNS アンプ攻撃の早期対策を目的とした DNS ハニーポットとダークネットの突合分析”, 2014 年暗号と情報セキュリティシンポジウム CD-ROM 論文集, セッション 3A5-3, 2014.
[11] 吉岡克成, 村上洸介, 松本勉, “マルウェア感染ホスト検出のためのネットワークスキャン手法と検出用シグネチャの自動生成”, 情報処理学会論文誌, vol 51, no. 9, pp. 1633-1644, 2010.
[12] Encyclopedia entry: Worm: Win32/Morto.A, Malware Protection Center (Microsoft), http://www.microsoft.com/security/portal/threat/encyclopedia/entry.aspx?Name=Worm:Win32/Morto.A (最終閲覧日:2014/8/20).
[13] Z. Durumeric, E. Wustrow, and J. A. Halderman, ”ZMap: Fast Internet-wide scanning and its security applications.” in Proeedings of the 22nd USENIX Security Symposium, 2013.
[14] IPv4 Scan: Scanning the Web for Open Proxy Servers”, http://ipv4scan.com/ (最終閲覧日:2014/8/9).
[15] Project Sonar by Rapid7, https://sonar.labs.rapid7.com/ (最終閲覧日:2014/8/9).
[16] 田辺瑠偉, 筒見拓也, 小出駿, 牧田大佑, 吉岡克成, 松本勉, “Linux 上で動作するマルウェアを安全に観測可能なマルウェア動的解析手法の提案”, 情報処理学会, コンピュータセキュリティシンポジウム 2014 (発表予定).
[17] Backdoor in wireless DSL routers lets attacker reset router, get admin | Ars Technica, http://arstechnica.com/security/2014/01/backdoor-in-wireless-dsl-routers-lets-attacker-reset-router-get-admin/ (最終閲覧日:2014/8/9).
[18] Linux.Aidra vs Linux.Darlloz: War of the Worms, http://blogs.avg.com/news-threats/war-of-the-worms/ (最終閲覧日:2014/8/9).
[19] Linux.BackDoor.Gates.5 — yet another Linux Trojan — Dr.Web http://news.drweb.com/show/?i=5801&lng=en (最終閲覧日:2014/8/20).