TCP フィンガープリントによる悪意のある通信の分析

木佐森 幸太 †　　下田 晃弘 †　　森 達哉 †‡　　後藤 滋樹 †

† 早稲田大学理工学術院　基幹理工学研究科
169–8555 東京都新宿区大久保 3–4–1
{kisamori,shimo,tatsuya,goto}@goto.info.waseda.ac.jp
‡NTT サービスインテグレーション基盤研究所
180–8585 東京都武蔵野市緑町 3–9–11
mori.tatsuya@lab.ntt.co.jp

あらまし Trojan.Srizbi に代表されるフルカーネル・マルウェア (FKM) は独自のネットワークドライバを実装し，カーネルモードの通信を行うことで監視ツールからの隠匿を試みる．これらのネットワークドライバは独自の実装であるため，既存 OS とは異なる TCP ヘッダの特徴 (フィンガープリント，以下 FP) を有することが知られている．本研究では CCC DATAset の攻撃通信データを分析対象とし，FKM の可能性が高い独自な FP を抽出する．また，その中で出現頻度の高い FP を有する IP アドレス発の通信を詳細に分析する．さらに，発見した FP を他の実ネットワーク計測データに適用し，FKM 感染ホストの実態調査および通信パターン分析を行い，提案した技法の有効性を検討する．

Analysis of malicious traffic based on TCP fingerprinting

Kota Kisamori†　　Akihiro Shimoda†　　Tatsuya Mori†‡　　Shigeki Goto†

†School of Science and Engineering, Waseda University
3–4–1 Okubo, Shinjuku-ku, Tokyo, 169–8585 Japan
{kisamori,shimo,tatsuya,goto}@goto.info.waseda.ac.jp

‡NTT Service Integration Laboratories
3–9–11 Midori-Cho Musashino-Shi Tokyo 180-8585 Japan
mori.tatsuya@lab.ntt.co.jp

Abstract Modern *full-kernel malware* (FKM) such as Trojan.Srizbi is known to have its own network functionality and use it directly from kernel-mode for evasion purpose, concealing from recent alert tools. These network drivers tend to have its own implementation which exhibits intrinsic TCP fingerprints. We first attempt to extract FKM-possible TCP fingerprints using CCC (Cyber Clean Center) data sets, then we analyze traffic sequences which have frequretly-occured source IP addresses. We then study the properties of the fingerprints and the activities of the hosts that are likely infected with FKM through the analysis of CCC data sets and other traffic data sets of several production networks.

1 はじめに

ボットネットは悪意のあるソフトウェア (マルウェア) に感染したホストによって構成されるネットワークであり，大規模のもので全世界に遍在する 1000 万オーダーの感染ホストから構成される [1]．ボットネットは遠隔地の攻撃者によって独自の暗号化された通信チャネルを用いて操作され，スパム送信，DDoS 攻撃，フィッシング等，違法な目的で利用される．近年のマルウェアの高機能化に伴い，ボットネットの構造や隠蔽のための手段はますます巧妙化している．その為，攻撃者のみならず，攻撃の踏み台である感染ホスト自体の検出が困難な課題である．ボットネットによる加害元特定の困難さは社会的にも深刻な問題となっている．

近年のマルウェアの傾向のひとつに，マルウェアのカーネル化が報告されている [2]．カーネル・マルウェアとは最も権限の高いレベル (Ring0) で動作し，メモリ，CPU 命令，すべてのハードウェアデバイスへのフルアクセスが可能なマルウェアである．カーネル・マルウェアの内，すべてがカーネルモードドライバで実装され，コードのすべてが Ring0 で実行されるものをフル・カーネル・マルウェア (FKM) と呼ぶ．

FKM 自体の歴史は古く，1999 年には FKM として WinNT/Infis の存在が報告されているが，2006 年末頃までは数の上では極めてマイナーな存在であった [2]．2007 年中旬から 2008 年末までに猛威を振るい，全世界のスパムメールの約半分に貢献したとされる Srizbi.trojan は FKM の一種である Reactor Mailer を実装し，独自のネットワークドライバを用いて SMTP 通信を行う機能を有する [5]．これらの独自ネットワークドライバは OS 由来の TCP/IP とは異なる実装であるため，TCP/IP ヘッダの組み合わせを注意深く観測することで (後述する TCP フィンガープリント技術)FKM のネットワークドライバ発のパケットと通常の Winsock 等を経由した OS 由来のパケットの識別が可能である [5]．この性質を利用し，文献 [3, 5, 4] では TCP フィンガープリントを利用したスパムボットの検出およびスパムボットの全体像解明に向けた大域的な分析を提案している．

本研究は FKM の有する特徴に着目し，ハニーポットの通信分析による FKM の検出と挙動の分析，および実運用網における FKM の実態把握を行う．FKM の検出と挙動分析用のデータとして CCC DATAset 2008 および 2009 [8] を用いる．また，FKM の検出と実態把握のために大学およびエンタープライズで計測した TCP ヘッダデータを用いる．

表 1: p0f のシグネチャの構成要素

W	ウィンドウサイズ
T	TTL の初期値
D	Don't Fragment ビット
S	SYN パケット全体のサイズ
O	TCP オプション（NOP，最大セグメントサイズ等）
Q	その他特徴的な点など
V	OS のバージョンなど

本研究の特筆すべき貢献として，FKM の可能性が高いボットの存在を複数のデータセットで確認したこと，およびそれらの特徴を明らかにしたことがあげられる．特に CCC DATAset においては FKM の可能性が高いホストが他の攻撃ホスト以上に観測され，これは我々の当初の予想を上回るものであった．

今回の分析において我々が前提とした仮説は，「**一般的な *OS* では利用されない *TCP* フィンガープリントを有する攻撃パケットは *FKM* 感染ホストが発信した**」と解釈することである．上記の仮説に基づく我々の推論をより確固とした根拠に基いて検証するためには，例えば文献 [6] のようなカーネル・ルートキットのプロファイリング等，得られたマルウェア検体の詳細な分析に基づく裏付けが必要である．これらは我々の近い将来の課題である．

2 TCP フィンガープリントによる CCC DATAset の分析

2.1 Passive fingerprinting

TCP/IP の仕様は RFC で定義されているが，OS 毎にその実装は異なる．このため，TCP パケットにおける各種オプションの設定値を注意深く観測することで対象システムの OS を推定することが出来る．このような技術を passive TCP fingerprinting といい，主に攻撃元の素性を非侵襲的に推察するための手段として利用されている．　本研究では代表的な passive TCP fingerprinting ツールである p0f [7] を利用する．

p0f にはいくつかのモードがあるが，今回用いたのは SYN パケットを用いるモードである．p0f シグネチャは [W:T:D:S:O...:Q:V] のようにコロンで区切られたフォーマットとなっている．各フィールドの意味を表 1 に示す．

2.2 FKM の可能性が高いフィンガープリントの抽出

CCCDATAset2008，2009 [8] の攻撃通信データ (インバウンド) に対し p0f（SYN モード）を適用したところ，既存の OS ではない UNKNOWN と判定された通信が多数検出された[1]．これらの通信について，TTL を 2 の累乗の値に切り上げて初期値と推定することで，43 のシグネチャに集約した．本研究ではこれらのシグネチャを MWS シグネチャと呼ぶ．また個々のシグネチャの第一ブロック（ウィンドウサイズ）によって名称をつけた．例えば，ウィンドウサイズが 65535 のシグネチャであれば，MWS 65535_1，MWS 65535_2…となる．抽出した MWS シグネチャすべてに共通な特徴は DF ビットが 0 に設定されていることである[2]．以下では MWS シグネチャの統計および攻撃パターンを分析し，MWS シグネチャを有するホストが高い確率で攻撃を成立させていることを示す．

2.3 MWS シグネチャの統計

図 1 にシグネチャの出現回数およびシグネチャ毎のユニークな送信元 IP アドレスの数を示す．2008 年，2009 年ともに，MWS シグネチャによる通信の方が既存シグネチャと比較して出現回数が多いことがわかる．2008 年が同程度であったのに対し，2009 年は全体として通信回数が減少しているものの，MWS シグネチャによる通信が占める割合は高くなっている．

出現回数とは対照的に MWS シグネチャで通信してくるホストの割合はやや減少しているものの，ホストの半数以上が MWS シグネチャを有することがわかる．後述するように大学のネットワークにおいて観測された MWS シグネチャを有するホストの割合はわずかに 0.03% (=280/954,100) であり，CCC DATAset で観測された MWS シグネチャホストの比率はきわめて高いことがわかる．また，既存・MWS 両方のシグネチャで通信を試みたホストもごく少数ではあるが存在した．

つぎに個々のシグネチャを分析する．図 2，図 3 はシグネチャごとに出現回数および送信元

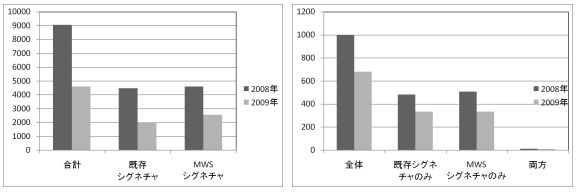

図 1: MWS シグネチャの統計: (左) 出現回数, (右) シグネチャ毎の送信元 IP アドレス数

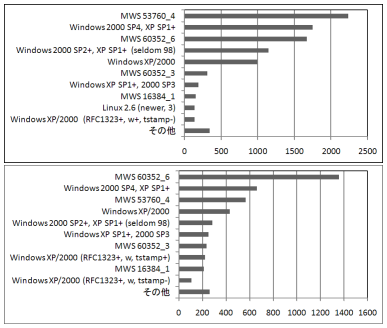

図 2: シグネチャ毎の出現回数: (上)2008 年, (下)2009 年.

IP アドレス数を集計したグラフである（上位 10 位まで）．2009 年の送信元 IP 数以外は MWS シグネチャがトップを占めており，それ以外にも複数の MWS シグネチャが上位に来ている．Windows 系シグネチャによる通信と比較しても，MWS シグネチャによる通信はインバウンドの通信においてかなりの割合を占めていることが分かる．

出現回数，送信元 IP ともに上位に位置している 4 種の MWS シグネチャについて，送信先

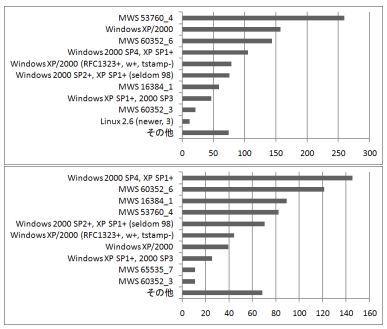

図 3: シグネチャ毎の送信元 IP 数: (上)2008 年, (下)2009 年.

[1] p0f のシグネチャは 2006 年以降更新されていないため，本研究ではマニュアルで収集した Windows Vista, Linux 2.6+，FreeBSD 7+，Mac OS 10.5+ 等のシグネチャを追加している．

[2] MWS シグネチャ以外のインバウンドの通信についても DF ビットが 0 であったため，この点については環境依存である可能性がある．

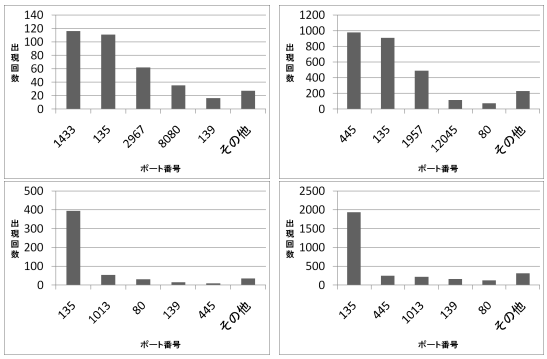

図 4: 送信先ポート番号分布: (左上) MWS 16384_1，(右上) MWS 53760_4，(左下) MWS 60352_3，(右下) MWS 60352_6

ポート番号を集計した結果を図 4 に示す (それぞれ上位 5 位まで)．135 番，139 番，445 番，1433 番，2967 番といった，著名な脆弱性に関連のあるポート番号の比率が高く，悪意のある通信が行われて可能性が疑われる．また，1013 番，1957 番，12045 番など未知のポート番号に対する通信もある程度存在することが分かる．また，シグネチャ毎に異なる傾向があることも読み取れる．次節ではこれらのシグネチャに対応する典型的な攻撃パターンを示す．

2.4 MWS シグネチャを有するホストの攻撃パターン

CCC DATAset 2009 の攻撃通信データにおいて，MWS シグネチャを有するホストの数は 345 個である．CCC DATAset におけるハニーポットは定期的にクリーンな状態にリセットされているが，ハニーポットへ SYN パケットを送信した後リセットされるまでの間にハニーポットからの SYN パケット送信があったホストは 345 個中 122 個であった．代表的な攻撃パターンの詳細を以下に例示する．

ケース I (MWS 60352_6)

```
21:26:41 ホスト A:9109 -> ハニーポット A:135 (scan)
21:26:41 ホスト A:9110 -> ハニーポット A:135 (rpc)
21:26:43 ホスト A:9197 -> ハニーポット A:135 (rpc)
21:26:43 ホスト A:9203 -> ハニーポット A:1013 (シェルコード送信)
21:26:43 ハニーポット A:1028 -> ホスト A:3450 (malware 要求)
21:26:43 ハニーポット A:1028 -> ホスト A:3450 (malware 要求)
```

最初の 135 番ポートに対する通信はスリーウェイハンドシェイク成立後に何もせず終了するが，2 度目・3 度目の 135 番ポートに対する通信では RPC プロトコルによる通信が行われていた．さらに，1013 番ポートに対する通信では，下記に示すようなシェルコードによって ftp でファイルをダウンロードしてそれを実行するよう命令を送っていた．ハニーポット A からホスト A に対する通信では ftp コマンドのやりとりによって実行ファイルが転送されるため，当該ファイルのダウンロードであると見られる．以上の観測結果より当該シグネチャの攻撃パターンは RPC のバッファ・オーバーフロー脆弱性に起因することが推察される．

シェルコードの例:

```
echo open xxx.xxx.xxx.xxx 2766 > i&echo user yyyyy zzzzz
>> i&echo get wmsoft05006.exe >> i&echo quit >> i&ftp -n
-s:i&startwmsoft05006.exe
```

ケース II (MWS 53760_4)

```
00:35:11 ホスト B:56101 -> ハニーポット B:135 (rpc)
00:35:13 ハニーポット B:1027 -> ホスト B:47602 (malware 要求)
00:35:13 ハニーポット B:1027 -> ホスト B:47602 (malware 要求)
```

ケース I とは異なり，最初の 135 番ポートに対する通信で RPC プロトコルによる通信を行っている．ハニーポット B から ホスト B に対する通信では，「This program cannot be run in DOS mode.」などの文字列が見えることから，実行ファイルのダウンロードが行われているものと考えられる．また，CCC DATAset 2009 の攻撃元データには，以下のように時刻・IP・ポート番号の合致する記録が存在している．

```
2009-03-13 00:35:13,ハニーポット B,1027,ホスト B,47602,
TCP,c925531e659206849bf7********************,
PE_VIRUT.AV,C:\WINNT\system32\csrs.exe
```

ケース III　(MWS 16384_1)

```
00:57:09 ホスト C:6000 -> ハニーポット B:135 (scan)
00:57:13 ホスト C:3197 -> ハニーポット B:135 (rpc)
00:57:15 ホスト C:4139 -> ハニーポット B:135 (rpc)
```

これはハニーポットからの SYN パケット送信がないケースである．最初の 135 番ポートに対する通信は，ハニーポットから SYN/ACK が返ってきた時点で RST しており，その後の 2 度の通信では RPC プロトコルによる通信を行っていた．なお，MWS 16384_1 では上記パターンの他 1433 番ポート，2967 番ポートに対する通信が多数存在したが，いずれもハニーポットから RST を返されて終了している様子が観測された．これは，それぞれ対応するアプリケーション（1433 番ポートは SQLServer，2967 番ポートは Symantec Client Security や Symantec AntiVirus）がハニーポット上で動いておらず，そのポートで LISTEN していなかったためと考えられる．

表 2: MWS シグネチャの通信内容

シグネチャ	ftp	http	irc	shell	smb	sql
MWS 60352_6	232	0	0	558	0	0
MWS 53760_4	50	1	0	307	0	0
MWS 60352_3	38	0	0	66	0	0
MWS 65535_7	12	0	0	21	0	0
MWS 60352_2	0	0	0	18	0	0
MWS 60352_1	0	0	0	6	0	0
すべての通信	694	563	202	1660	9234	723

2.5 シグネチャごとの通信内容分析

シグネチャ別に，通信として成立しているものの集計を行った．対象は，ボットの FTP ダウンロード，ボットの HTTP ダウンロード要求，IRC 通信，shell コード送信，smb 通信，SQL 攻撃の 6 種類である．一度以上通信が成立しているシグネチャは表にあるように 6 種類である．

すべてのインバウンド通信を対象に集計を行った結果と比べると，今回抽出した MWS シグネチャの通信には FTP 通信と SHELL コードが多く含まれているものがある一方，その他の 4 種の通信がほぼ存在しないことが分かる．

3 他のネットワークの分析結果

本章では，前章の分析の結果得られた FKM の可能性が高いシグネチャを他のネットワーク計測データに適用し，ボット感染ホストを検出・分析した結果を示す．

3.1 分析に利用したデータセット

本研究では二つのネットワークにおいて収集したデータを用いる．一つは大学で計測したデータであり，もう一方はエンタープライズで収集したデータである．簡単のため，本研究ではそれぞれを UNIV，CORP と表記する．

UNIV 早稲田大学の対外接続回線において収集した TCP ヘッダデータである．当該回線は学術 (帯域 10Gbps) および商用網 (帯域 300Mbps) を収容しており，収集データには両者の回線を総合したトラフィック情報が含まれる．収集時期は 2009 年 7 月 1 日から 7 月 8 日の 1 週間であり，TCP SYN パケットのみを収集の対象とした．

CORP ある企業網の電子メールサーバ (MTA) に接続したネットワークセグメントで収集した TCP ヘッダデータであり，この回線で観測可能な通信は SMTP のみである．すなわち，ボットネットのスパム活動が主な分析対象である．

表 3: UNIV における送信先ポート番号:(左)MWS 16384_1，(右)MWS 65535_13

送信先ポート番号	出現回数
2967	296855
3306	50638
4899	34831
8080	17181
8088	13371
その他	35845
合計	484566

送信先ポート番号	出現回数
443	9500
80	3710
143	437
8080	292
28080	218
その他	141
合計	14439

UNIV と同様に，TCP SYN パケットのみを収集した．データの収集時期は 2009 年 3 月 1 日から 3 月 31 日の 1ヶ月間である．上記の MTA ではスパムアプライアンスが動作しているため，ある IP アドレスから送信されたメールがスパムであったか否かの判定が可能である．

3.2 UNIV の分析結果

UNIV データにおいては，今回抽出した 43 のシグネチャのうち 20 が観測された．また，観測された全送信元 IP アドレス数は 954,100 であり，そのうち 280 の IP アドレスが今回発見したシグネチャを有するホストであった．通信の多かった上位 2 種のシグネチャは MWS 16384_1，MWS 65535_13 であり，それぞれの送信元 IP アドレス数は 38，148 であった．送信先ポート番号の分布を表 3 に示す．

MWS 16384_1 の送信先ポート番号として，CCC DATAset でも観測された 2967 番のほか，3306 番（mysql），4899 番（バックドアが使用するとされるポート）が多く，悪意のある通信である可能性が高い．MWS 65535_13 については，443 番ポート以外は IMAP や HTTP らしきポート番号となっており，ここまでに述べてきたシグネチャとは異なる傾向を見せている．

CCC DATAset で多く見つかった MWS 60352_x は見つからず，MWS 53760_x もごく少数にとどまったが，DF ビット・最大セグメントサイズのみが異なる通信は多数見つかっている．

総じて，今回抽出したシグネチャは検出されるものの，傾向は異なるといえる．これらの違いはネットワーク状況の差異によるものであると考えられる．より詳細な分析を得るためにはさらに広域かつ継続的な計測が必要であろう．

表 4: FKM の可能性が高いシグネチャを有するホストが発信したメールの統計 (上位 10 ホストのみ)．表中の S はスパム数，H は通常メール数，I は IP アドレス数.

signature	S	H	I
[65535:64:0:52:M1414,N,W3,N,N,S:.]	290	0	9
[65535:64:0:52:M1414,N,W0,N,N,S:.]	252	0	8
[65535:32:0:64:M1414,N,W3,N,N,T0,N,N,S:.]	188	0	3
[65535:64:0:52:M1400,N,W2,N,N,S:.]	90	0	4
[65535:64:0:52:M1414,N,W2,N,N,S:.]	64	0	6
[16384:128:0:60:M1414,N,N,T0,N,N,S:.]	25	0	3
[16384:16:0:40:.:.]	21	0	1
[65535:64:0:52:M1412,N,W2,N,N,S:.]	16	0	7
[53760:64:0:64:M1414,N,W3,N,N,T0,N,N,S:.]	16	0	2
[65535:64:0:64:M1414,N,W2,N,N,T0,N,N,S:.]	9	0	1

3.3 CORP 分析結果

CORP データにおいて観測された IP アドレス数は 1,230,830 であり，その内わずかに 53 アドレスが今回発見したシグネチャを有するホストであった．今回の検討外であるが，FKM の一種である Srizbi のシグネチャ [4] を有する IP アドレスの数は 40,322 であった．上記の 53 アドレスによる通信をシグネチャ毎にまとめたものが表 4 である．本データセットにおいて観測されたスパムメールの総数は約 1500 万であり，FKM と識別されたホスト発のスパムは非常に少ない．しかしながら，これらの識別されたホストが発信したメッセージの大多数がスパムであり，マルウェアの構成によってはスパム送信モジュールを搭載するものも存在することが窺える．

4 まとめと今後の課題

フルカーネル・マルウェアの可能性が高いホストの詳細分析，および実ネットワークにおける実態調査を行った．分析における鍵となるアイディアは TCP フィンガープリントを利用することである．一般の OS では利用されていないにも関わらず，出現頻度および送信 IP アドレス数の高い TCP フィンガープリントを抽出し，それらの抽出した TCP フィンガープリントを有する IP アドレス発の通信を詳細に分析することで FKM の可能性が高い通信およびホストを検出することが出来る．

分析の結果，CCC DATAset におけるハニーポットへの攻撃通信では FKM が当初の予想以上に存在することが明らかになった．また，FKM にはいくつかの種類が存在し，それぞれ異なる攻撃パターンを有すること，およびネットワークによって観測されるシグネチャが異なることが観測された． FKM は CCC Dataset のみならず，様々なネットワークで観察されていることから，マルウェアにおける FKM の割合は今後さらに増えていく可能性がある．FKM は Ring0 で動作するため，通常のアンチウィルスソフトウェア等の監視から隠匿する動作が可能である．新規に開発する OS のみならず，現存する大多数の OS に対して FKM の動作・実行を防止するための防衛メカニズムを構築することが重要である．

今回の研究で我々が前提とした仮説「**一般的な *OS* では利用されない *TCP* フィンガープリントを有する攻撃パケットは *FKM* 感染ホストが発信したものである**」をより確固とした根拠に基いて検証するためには，感染ホストのプロファイリングが必要不可欠であり，今後の我々の課題である．

謝辞 MWS に携る機会を与えて頂いた日立製作所の寺田真敏氏および本研究の初期段階において貴重な助言を頂いた NTT 情報流通プラットフォーム研究所の岩村誠氏に感謝します.

参考文献

[1] F-Secure Weblog, “Calculating the Size of the Downadup Outbreak,” `http://www.f-secure.com/weblog/archives/00001584.html`, Jan. 2009

[2] K. Kasslin, “Kernel Malware: The Attack from Within,” `http://www.f-secure.com/weblog/archives/kasslin_AVAR2006_KernelMalware_paper.pdf`, 2006

[3] H. Esquivel, T. Mori, and A. Akella, “Router-Level Spam Filtering Using TCP Fingerprints: Architecture and Measurement-Based Evaluation,” Sixth Conference on Email and Anti-spam, Jul., 2009

[4] T. Mori, H. Esquivel, A. Akella, A. Shimoda, and S. Goto, “Understanding the World’s Worst Spamming Botnet,” University of Wisconsin Madison Tech Report TR1660, June 2009.

[5] H. Stern, “The Rise and Fall of Reactor Mailer,” MIT Spam Conference 2009, Mar., 2009.

[6] R. Riley, X. Jiang, and D. Xu, “Multi-aspect profiling of kernel rootkit behavior,” Fourth ACM european conference on Computer systems, Apr., 2009

[7] Michal Zalewski, “the new p0f: 2.0.8,” `http://lcamtuf.coredump.cx/p0f.shtml`, 2006

[8] 畑田充弘・他，“マルウェア対策のための研究用データセットとワークショップを通じた研究成果の共有，” MWS2009（2009 年 10 月）