2009 年度 修士論文

TCP フィンガープリントによる
悪意のある通信の分析

提出日： 2010 年 2 月 5 日

指導：後藤滋樹教授

早稲田大学大学院 基幹理工学研究科 情報理工学専攻
学籍番号： 5108B034-7

木佐森　幸太

目次

図一覧

表一覧

第 1 章

序論

1.1 研究の背景

インターネットの普及・発展は情報通信の世界を大きく変え、様々な恩恵を現代社会にもたらした。しかし、その普及・発展に伴い、ネットワーク・セキュリティ上の問題点を突いてユーザに害をなそうとする悪意のある試みも増加してきた。インターネットが重要な社会インフラの一つとなっている現在、これらの悪意のある試みが社会に及ぼす影響も大きくなってきている。

今日のインターネットにおける代表的な脅威の一つがボットネットである。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，大規模なもので 10 万オーダーの感染ホストから構成される．ボットネットは攻撃者によって独自の通信チャネルを用いて遠隔操作され，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．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おり、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になっている。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きている．

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．カーネルモード（CPU の動作モードの中で最も権限が高いもの）で動作するマルウェアは、ウィルス対策ソフトによる検知並びに駆除を困難にし、かつファイアウォールを迂回した通信を可能にする恐れがある。このようなマルウェアの中でも、一切ユーザモードを使わずに、すべての動作をカーネルモードで実行できるマルウェアをフルカーネルマルウェアと呼ぶ．

1.2 研究の目的

悪用目的として初めてのフルカーネルマルウェアとされる Srizbi というマルウェアには、独自のネットワークドライバを用いて通信を行う機能を有していた。この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ったため、通信の特徴を分析することで、既存の OS による通信と Srizbi による通信を区別することが可能である。

本研究では、サイバークリーンセンター（CCC) で運用されているハニーポットの通信を分析して既存 OS とは異なる特徴を有する通信を抽出し、その挙動を分析して、カーネルマルウェアの可能性があるホストを検出する手法を提案する。また、実運用網の通信データにそれを適用して、その有効性を実証する。

1.3 本論文の構成

本論文は以下の章により構成される。

第 1章 序論

本研究の概要について述べる。

第 2章 カーネルマルウェアと TCP fingerprinting

カーネルマルウェアの概要、 TCP Fingerprinting の概要、提案手法について説明する。

第 3章 CCC DATAset の分析

CCC で運用されているハニーポットの通信を分析した結果を報告する。

第 4章 他のネットワークのデータへの応用

第 4 章での分析結果を他の実ネットワークにおける通信データに適用した結果を述べる。

第 5章 まとめ

本論文のまとめと今後の課題を述べる。

第 2 章

カーネルマルウェアと TCP fingerprinting

本章ではまずカーネルマルウェアについて解説する。次に TCP fingerprinting と関連研究について述べ、最後に提案手法を説明する。

2.1 カーネルマルウェア

CPU には、権限の異なる複数の動作モードが設定されていることがある。もっとも特権レベルの高いモードはカーネルモードと呼ばれ、命令の実行やコンピュータの各リソースへのアクセスなどが完全に無制限となる。他のモードはユーザモードと呼ばれる。特権レベルの低いモードから、特権レベルの高い側のリソースに直接アクセスすることはできない。このカーネルモードで実行されるマルウェアがカーネルマルウェアである。

これらのマルウェアは、カーネルモードで動作することによりシステムの様々なリソースに変更を加えることができるため、侵入した形跡を消して感染したことを隠ぺいすることが可能である。

こういった技術は、従来ルートキットと呼ばれるものに使われてきた。ルートキットとは、もともと不正侵入するためのソフトウェアツールのセットを指す言葉であったが、徐々に「OS を改ざんして侵入を隠ぺいする機能」がルートキットの特徴として大きな部分を占めるようになった。

インストール後一切ユーザモードを使わずに、すべての動作をカーネルモードで実行できるマルウェアをフルカーネルマルウェア（FKM）という。 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。 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をもたらしたとされる 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。カーネルモードでネットワーク接続を行う利点は、ユーザモードを介さないことで、パーソナルファイアウォール等に妨げら

れることなく通信ができるという点である。

こ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 fingerprinting 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。

2.2 TCP fingerprinting

TCP/IP の仕様は RFC で定義されているが、 OS ごとにその実装は異なっている。このため、通信データを分析することで対象システムの OS を推定することが可能である。このような技術を fingerprinting という。 fingerprinting は、対象システムに対し通信を行うか否かによって Active と Passive に分類することができる。

2.2.1 Active fingerprinting

対象システムへの通信を行い、それによって得られた通信データから OS を推定するのが Active fingerprinting である。後述の Passive fingerprinting に比べると、 OS の判別に利用できるデータ量が多い点で優れているが、対象システムに不要な ICMP パケットをブロックするなどの対策が施されている可能性がある。そのため、常に有効とは限らない。

代表的なツールとして、ポートスキャナとしても有名な nmap が挙げられる。

2.2.2 Passive fingerprinting

対象システムへの通信を行わず、受信した通信データのみを分析することで OS を推定するのが Passive fingerprinting である。 Active fingerprinting に比べ利用できるデータ量は少ないが、新たにトラフィックを発生させる必要がないのが利点である。

代表的なツールとして p0f がある。

2.2.3 p0f

ここでは p0f の詳細について述べる。 p0f ではいくつかのモードがあるが、最もよく用いられるのは SYN パケットのみを分析対象として対象ホストの OS を推定するモードである。 SYN モードでは「wwww:ttt:D:ss:OOO...:QQ:OS:Details」という形式のシグネチャとして各 OS のデータが集約されている。それぞれの意味を表 2.1に示す。

なお、ウィンドウサイズについては Snn （nn は数値）となることがある。これは、最大セグメントサイズオプションの値の nn 倍という意味である。

表 2.1: p0f のシグネチャの構成要素

wwww	TCP ヘッダにおけるウィンドウサイズ
ttt	IP ヘッダにおける TTL の初期値
D	IP ヘッダにおける DF（Don't Fragment）ビット
ss	SYN パケット全体のサイズ
OOO	TCP オプション
QQ	その他特徴的な点など
OS	OS の種類（Windows、 Linux など）
Details	OS の詳細（バージョンなど）

表 2.2: p0f のシグネチャにおける TCP オプションリスト

N	NOP オプション
E	EOL オプション（オプションリストの終了）
Wnnn	ウィンドウスケールオプション（nnn は値を表す）
Mnnn	最大セグメントサイズオプション（nnn は値を表す）
S	Selective ACK オプション
T	タイムスタンプオプション
T0	タイムスタンプオプション（タイムスタンプ値が 0）
?n	上記以外のオプション（n はオプションを表す番号）

TCP オプションの詳細を表 2.2に示す。複数のオプションが設定されている場合は、現れた順にコンマ区切りで表わされる。同じオプションが設定されていても、現れる順序が異なる場合は別のシグネチャとなる。

QQ の部分に現れるその他の特徴は表 2.3のとおりである。（特に記載がない場合は TCP ヘッダにおけるもの）

シグネチャの例を以下に示す。

```
65535:128:1:48:M*,N,N,S:.:Windows:2000 SP4, XP SP1+
```

これは、

- ウィンドウサイズが 65535 バイト
- TTL の初期値が 128

表 2.3: p0f のシグネチャにおけるその他の特徴リスト

E	EOL オプションの後にオプションがある
Z	IP パケットにおける ID フィールドが 0 である
I	IP パケットでオプションが設定されている
U	緊急ポインタフィールドが 0 でない
X	未使用領域が 0 でない
A	ACK 番号が 0 でない
T	タイムスタンプ・エコー応答の値が 0 でない
F	通常設定されないフラグ（URG、 PSH など）が設定されている
D	TCP ヘッダの後にデータが存在する
!	TCP オプション部分が正常に読めない
.	上記すべてに該当しない

- DF ビットが 1
- SYN パケット全体のサイズが 48 バイト
- 以下の順に TCP オプションが設定されている
  - 最大セグメントサイズオプション（値は問わない）
  - NOP オプション
  - NOP オプション
  - Selective ACK オプション
- その他特徴的な点はなし

以上のような特徴を持つ SYN パケットを送信するホストは、 OS が Windows 2000 SP4 または Windows XP SP1 以降と推定できるということを表す。

2.2.4 fingerprinting の利用例

fingerprinting を行う効用としてまず考えられるのが、悪意のあるユーザによる攻撃対象の調査である。攻撃対象の OS やそのバージョンが分かれば、その OS の脆弱性を突いた効果的な攻撃を行うことができるためである。また、外部からの攻撃に対する備えとして、セキュリティ診断時に fingerprinting 対策が施されているかの確認として行われることもある。

また、 Passive fingerprinting がスパムメール対策に利用される例もある。通常のメールでは送信元ホストがメールサーバとなるため、 OS はサーバ向けのもの（Unix 系 OS、 Windows Server など）である可能性が高い。逆に、送信元ホストがサーバ用ではなく一般の PC 向け Windows であった場合、それはボット感染ホストである可能性が高い。この点から、送信元ホストの推定結果をスパムメール判定用のスコアリングに利用するというものである。

2.3 既存研究

2.1 節で述べたように、 Srizbi.trojan に実装されているネットワークドライバは既存の OS とは異なる独自の実装であったため、 fingerprinting を行うことで Srizbi による通信を識別することが可能であった。参考文献 [5,6,7] では、 p0f を利用した場合の Srizbi のシグネチャを特定したうえで、 Passive TCP fingerprinting 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。

2.4 提案手法

Srizbi.trojan に限らず、独自のネットワークドライバを持つカーネルマルウェアによる通信は、既存 OS と異なる独自の TCP/IP 実装を用いている可能性がある。 Passive TCP fingerprinting において、これらのカーネルマルウェアによる通信を特定することができれば、攻撃通信に対処したり、攻撃元ホストを特定したりする上で有用である。

本研究では、まずハニーポットの通信データを分析の対象とする。ハニーポットは脆弱性に対するパッチを当てていないシステム、または既存の脆弱性をシミュレートすることができるシステムであり、わざと外部からの攻撃を受けてマルウェアを収集したり、感染時・感染後の挙動を観察したりする目的で用いられる。ハニーポットは基本的に自発的な通信を行わないため、外部からの通信は悪意のあるものである可能性が高い。この通信データに対して p0f を適用して既存の OS による通信ではないと推定されたものを抽出し、シグネチャという形に集約する。これらの通信は、既存の OS ではなく、独自のネットワークドライバを用いたカーネルマルウェアによる悪意のある通信であると考えられるため、シグネチャベースでの分析を行い、その仮説を検証する。

次に、ハニーポットの通信データから抽出したシグネチャを外部の実運用ネットワーク通信データに適用し、その結果を分析する。これによって、シグネチャによる悪意のある通信の特定という手法の有効性を検討する。

第 3 章

CCC DATAset の分析

3.1 CCC DATAset とは

CCC DATAset は、サイバークリーンセンター（CCC）で運用しているハニーポットのデータから作成された、マルウェア研究用データセットである。 2008 年、 2009 年の 2 年分があり、それぞれマルウェア検体、攻撃通信データ、攻撃元データからなる。これらはもともとマルウェア対策研究人材育成ワークショップ（MWS） 2008、 2009 のために作成されたものである。

3.1.1 マルウェア検体

ハニーポットで収集したマルウェア検体のハッシュ値（MD5、 SHA1）をテキスト形式で記載したファイルである。 CCC DATAset 2008 では 1 個のハッシュ値が、 CCC DATAset 2009 では 10 個のハッシュ値が記載されている。

3.1.2 攻撃通信データ

ハニーポット 2 台の通信を tcpdump でパケットキャプチャしたファイルである。 OS は Windows 2000 と Windows XP SP1 である。ハニーポットは仮想マシンで作成されており、定期的にクリーンな状態にリセットされる。

CCC DATAset 2008 におけるデータ収集日は 2008 年 4 月 28 日と 2008 年 4 月 29 日であり、総パケット数が 15,901,943 パケット、データサイズは約 2.8GB である。

CCC DATAset 2009 におけるデータ収集日は 2009 年 3 月 13 日と 2009 年 3 月 14 日であり、総パケット数が 3,511,850 パケット、データサイズは約 580MB である。

3.1.3 攻撃元データ

ハニーポットで記録したマルウェア取得時のログである。マルウェア検体の取得時刻、送信元 IP アドレス、ポート番号、ウィルス名称などが記録されている。 CCC DATAset 2008 における

データ収集期間は 2007 年 11 月 1 日から 2008 年 4 月 30 日までの 6ヶ月間であり、 CCC DATAset 2009 におけるデータ収集期間は 2008 年 5 月 1 日から 2009 年 4 月 30 日までの 1 年間である。

3.2 カーネルマルウェアの可能性があるシグネチャの抽出

CCC DATAset 2008、 2009 の攻撃通信データに対し 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。また、分析結果の中には類似しているものが多数見受けられた。例を以下に挙げる。

```
[60352:49:0:52:M1414,N,W2,N,N,S:.:?:?]
[60352:50:0:52:M1414,N,W2,N,N,S:.:?:?]
[60352:52:0:52:M1360,N,W2,N,N,S:.:?:?]
[60352:52:0:52:M1414,N,W2,N,N,S:.:?:?]
[60352:53:0:52:M1360,N,W2,N,N,S:.:?:?]
[60352:53:0:52:M1414,N,W2,N,N,S:.:?:?]
[60352:54:0:52:M1332,N,W2,N,N,S:.:?:?]
```

2.2.3 節でも述べたとおり、シグネチャの第 2 フィールドは TTL の初期値であるが、分析時に直接参照できるデータは受信時の TTL であり、対象ホストからのホップ数に応じて減少しているものである。したがって、シグネチャとするためにはこれらの値を直接用いるのではなく、適当な値に切り上げなければならない。本研究では、最も近い 2 の累乗の値に切り上げることとした。

また、上記例でもう一つ異なっている点として、 Mnnn （最大セグメントサイズオプション）の数字部分がある。この点については第 4 章でも取り上げるが、まずは集約することなくそのままとした。

UNKNOWN と判定された通信すべてについて TTL を 2 の累乗の値に切り上げた結果、インバウンド 43 種、アウトバウンド 1 種の合計 44 種のシグネチャが得られた。これらを MWS シグネチャと総称し、ウィンドウサイズ（シグネチャの第 1 フィールド）の値によって分類して名称を付けた。以下シグネチャの一覧を挙げる。

```
16384:128:0:40:.:.:MWS:16384_1
16384:128:0:44:M1414:.:MWS:16384_2
16384:128:0:60:M1414,N,N,T0,N,N,S:.:MWS:16384_3

32768:255:0:52:M1414,N,N,S,W0,N:.:MWS:32768_1
```

```
32768:64:0:64:M1414,N,W0,N,N,T,S,E:P:MWS:32768_2

44620:64:0:64:M1414,N,W3,N,N,T0,N,N,S:.:MWS:44620_1

53760:64:0:64:M1360,N,W3,N,N,T0,N,N,S:.:MWS:53760_1
53760:64:0:64:M1380,N,W3,N,N,T0,N,N,S:.:MWS:53760_2
53760:64:0:64:M1398,N,W3,N,N,T0,N,N,S:.:MWS:53760_3
53760:64:0:64:M1414,N,W3,N,N,T0,N,N,S:.:MWS:53760_4
53760:32:0:64:M1414,N,W3,N,N,T0,N,N,S:.:MWS:53760_41

5808:64:0:60:M1396,S,T,N,W0:.:MWS:5808_1
5808:64:0:60:M1414,S,T,N,W0:.:MWS:5808_2

60352:64:0:52:M1240,N,W2,N,N,S:.:MWS:60352_1
60352:64:0:52:M1332,N,W2,N,N,S:.:MWS:60352_2
60352:64:0:52:M1360,N,W2,N,N,S:.:MWS:60352_3
60352:64:0:52:M1380,N,W2,N,N,S:.:MWS:60352_4
60352:64:0:52:M1398,N,W2,N,N,S:.:MWS:60352_5
60352:64:0:52:M1414,N,W2,N,N,S:.:MWS:60352_6

65535:128:0:48:M1414,N,N,N,N:.:MWS:65535_1
65535:64:0:48:M1414,N,N,S:.:MWS:65535_2
65535:64:0:52:M1400,N,W2,N,N,S:.:MWS:65535_3
65535:64:0:52:M1412,N,W2,N,N,S:.:MWS:65535_4
65535:64:0:52:M1414,N,W0,N,N,S:.:MWS:65535_5
65535:128:0:52:M1414,N,W2,N,N,N,N:.:MWS:65535_6
65535:64:0:52:M1414,N,W2,N,N,S:.:MWS:65535_7
65535:255:0:52:M1414,N,W2,N,N,S:.:MWS:65535_71
65535:64:0:52:M1414,N,W3,N,N,S:.:MWS:65535_8
65535:128:0:60:M1398,N,N,T0,N,N,S:.:MWS:65535_9
65535:64:0:64:M1380,N,W3,N,N,T0,N,N,S:.:MWS:65535_10
65535:64:0:64:M1414,N,W0,N,N,T0,N,N,S:.:MWS:65535_11
65535:64:0:64:M1414,N,W2,N,N,T0,N,N,S:.:MWS:65535_12
65535:64:0:64:M1414,N,W3,N,N,T,S,E:P:MWS:65535_13
```

```
65535:64:0:64:M1414,N,W3,N,N,T0,N,N,S:.:MWS:65535_14

8192:128:0:40:.:.:MWS:8192_1
8192:128:0:56:M1414,S,T:.:MWS:8192_2
8192:64:0:64:M1408,N,W0,N,N,T0,N,N,S:.:MWS:8192_3
8192:64:0:64:M1414,N,W0,N,N,T0,N,N,S:.:MWS:8192_4

8576:64:0:48:M1414,N,N,S:.:MWS:8576_1

S122:32:0:64:M536,N,W0,N,N,T0,N,N,S:.:MWS:S122_1

S4:64:0:52:M1414,N,N,S,N,W8:.:MWS:S4_1

S44:128:0:44:M1414:.:MWS:S44_1
S44:128:1:48:M1460,N,N,S:Z:MWS:S44_2
S44:64:0:60:M1414,S,T,N,W13:.:MWS:S44_3
```

アウトバウンド通信から抽出されたシグネチャは MWS S44_2 のみである。インバウンド通信から抽出されたシグネチャについては、 DF ビットがすべて 0 となっている。

以下に、抽出したシグネチャをベースとした分析の結果について述べる。

3.3 MWS シグネチャの統計

図 3.1にシグネチャ種別（既存 OS、 MWS シグネチャ）ごとの SYN パケット数を示す。 SYN パケット数については、 2008 年、 2009 年ともにほぼ半数ずつであるが、 MWS シグネチャの方が既存シグネチャと比較して若干多いことがわかる。 2009 年は 2008 年と比較して SYN パケット数全体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。

図 3.2にシグネチャ種別ごとの送信元 IP アドレス数を示す。 SYN パケット数についてと同様に、既存 OS と MWS シグネチャとでほぼ半数ずつとなっている。 2009 年を 2008 年と比較すると MWS シグネチャの送信元 IP の割合が減少しているが、 MWS シグネチャの方が若干多い点は同じである。また、既存 OS・ MWS の両方のシグネチャで SYN パケットを送信してきたホストもごく少数ではあるが存在した。

次に個々のシグネチャについて分析する。図 3.3、 3.4はシグネチャごとに SYN パケット数を集計したグラフであり、図 3.5、 3.6は送信元 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

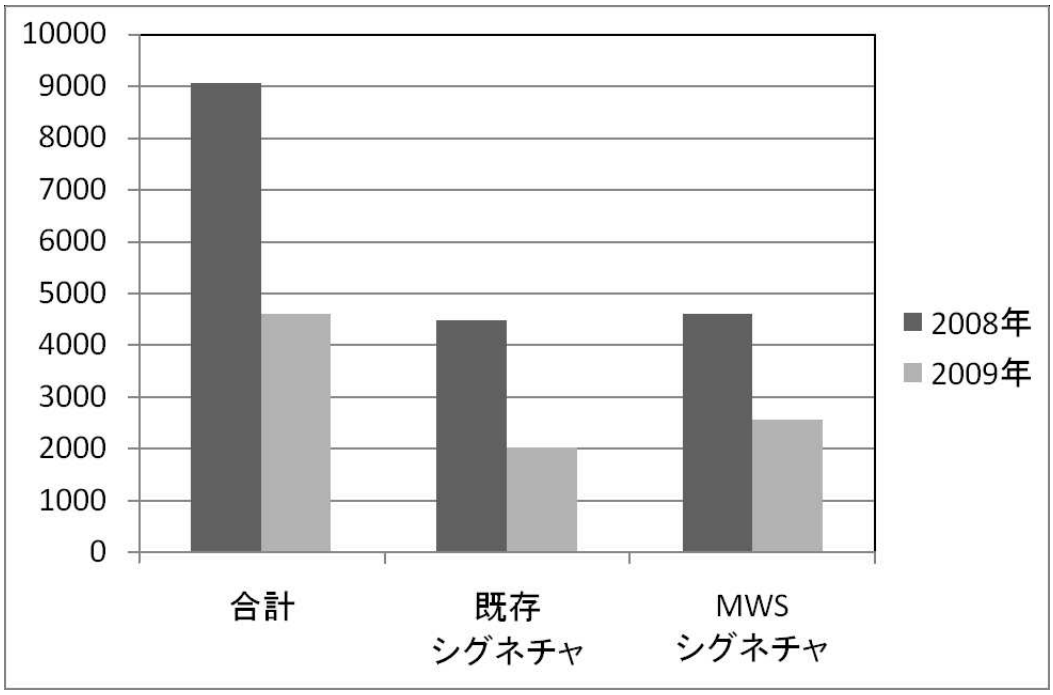

図 3.1: シグネチャ種別ごとの SYN パケット数

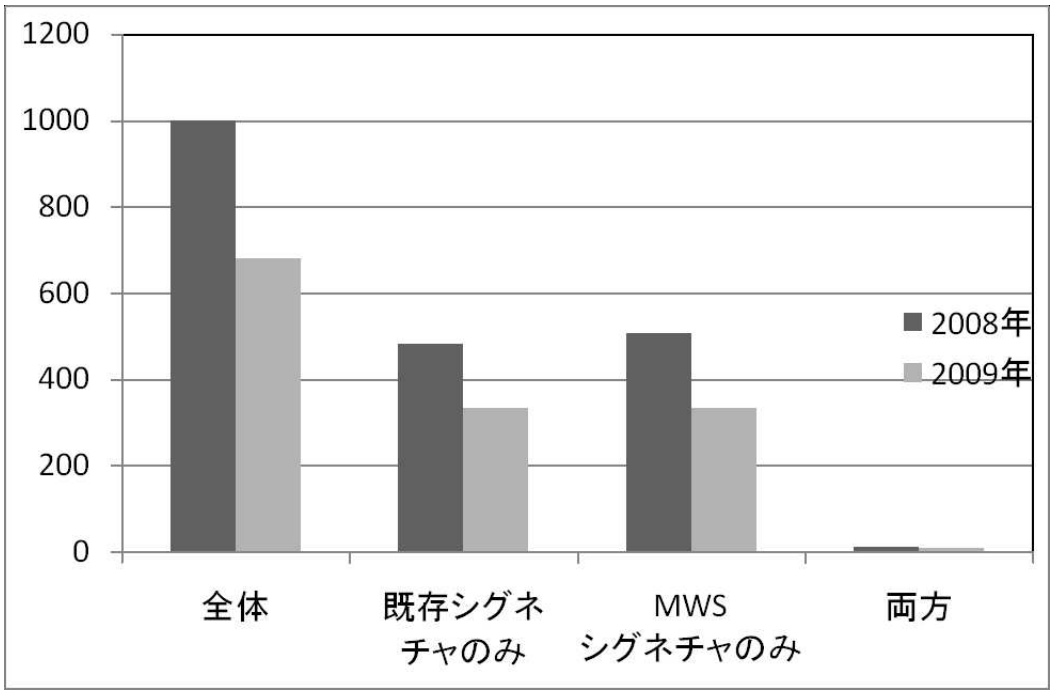

図 3.2: シグネチャ種別ごとの送信元 IP 数

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が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わかる。

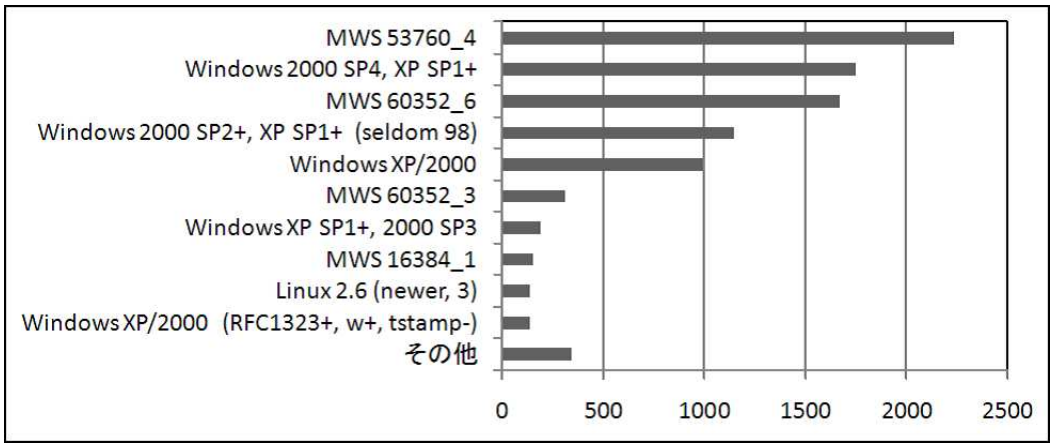

図 3.3: シグネチャごとの SYN パケット数（2008 年）

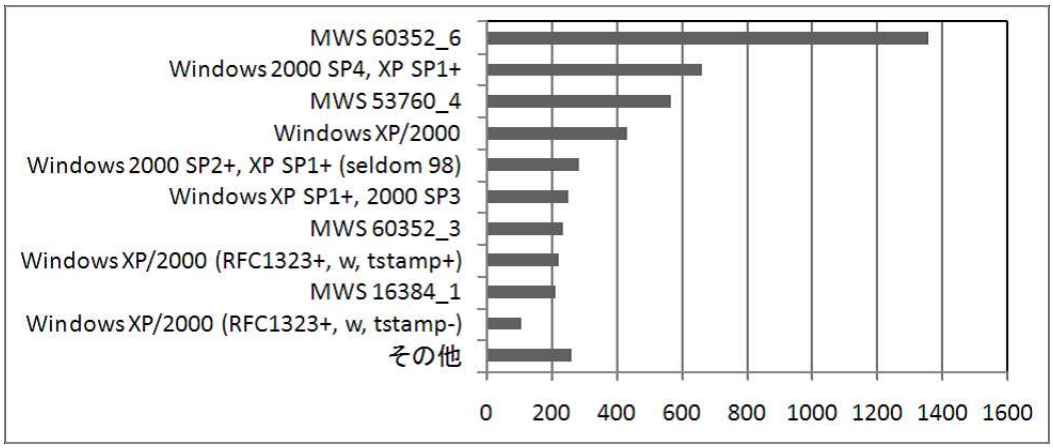

図 3.4: シグネチャごとの SYN パケット数（2009 年）

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、 CCC DATAset 2009 における送信先ポート番号を集計した結果を図 3.7、 3.8、 3.9、 3.10に示す（それぞれ上位 5 位まで）。ここに登場するポート番号のうち、 135 番、 139 番、 445 番については Windows における Remote Procedure Call やファイル共有における脆弱性があることが知られている。また、 1433 番については Microsoft SQL Server が使用するポートであり、 2967 番についてはシマンテックのセキュリティ製品の脆弱性があることが知られている。いずれもポートスキャンの対象として著名なものであり、特に 135 番ポートは Blaster が狙うポート番号であった。こう

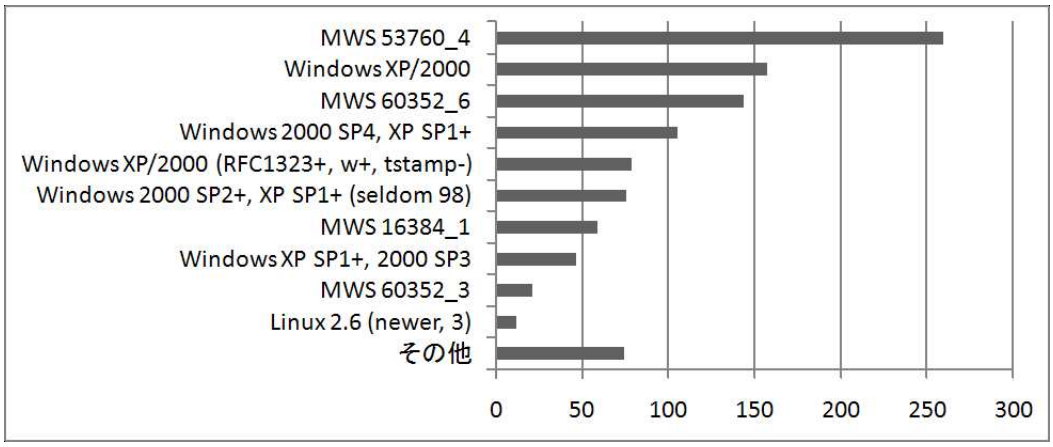

図 3.5: シグネチャごとの送信元 IP 数（2008 年）

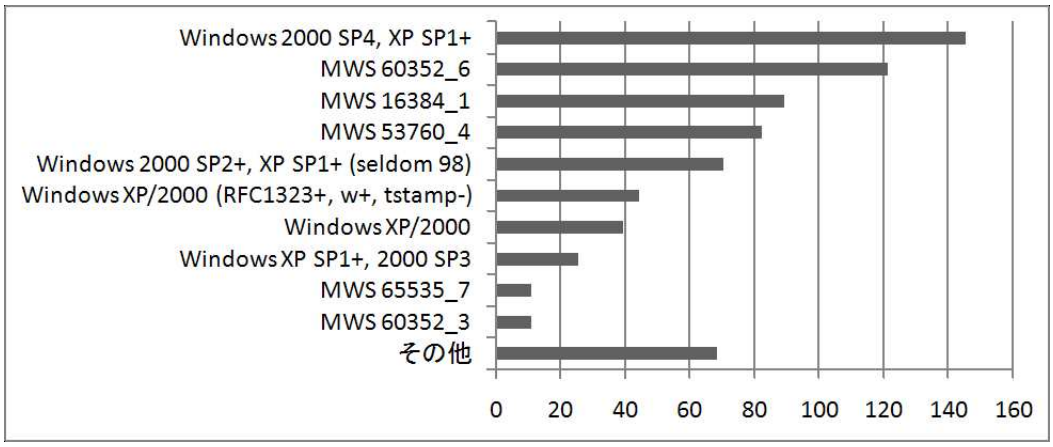

図 3.6: シグネチャごとの送信元 IP 数（2009 年）

いったポート番号への通信の比率が高いことから、悪意のある通信が行われている可能性が高いと考えられる。

他にも、 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、シグネチャ毎に異なる傾向があることが読み取れる。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。

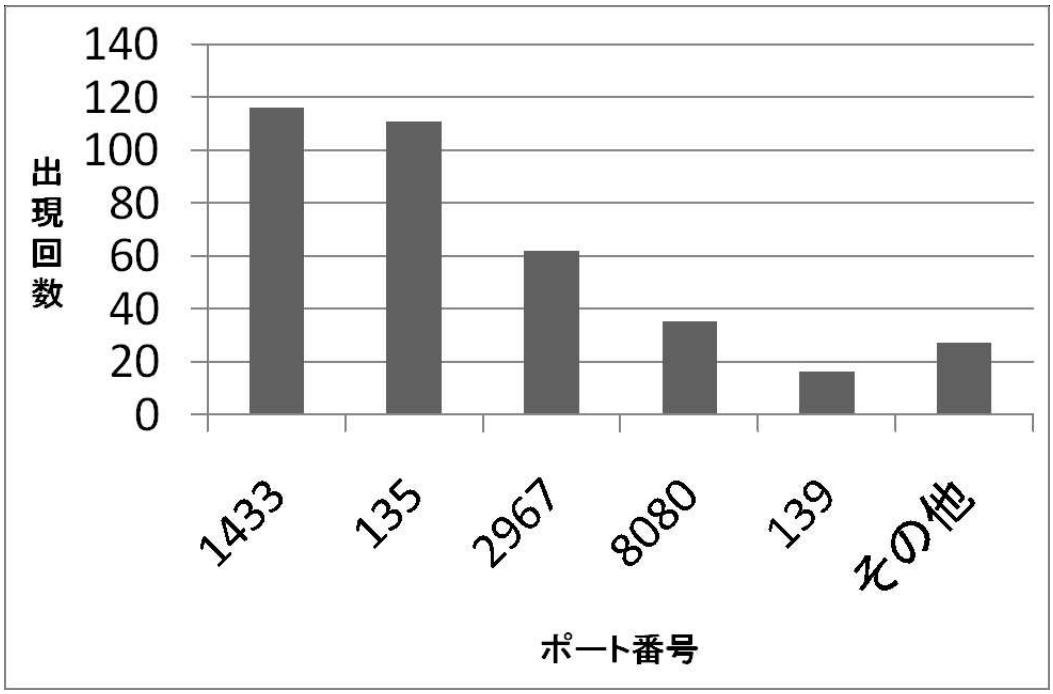

図 3.7: 送信先ポート番号分布 MWS 16384_1 （2009 年）

3.4 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送信元 IP アドレスの数は 345 個である。攻撃通信データを取得してい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。

3.4.1 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)

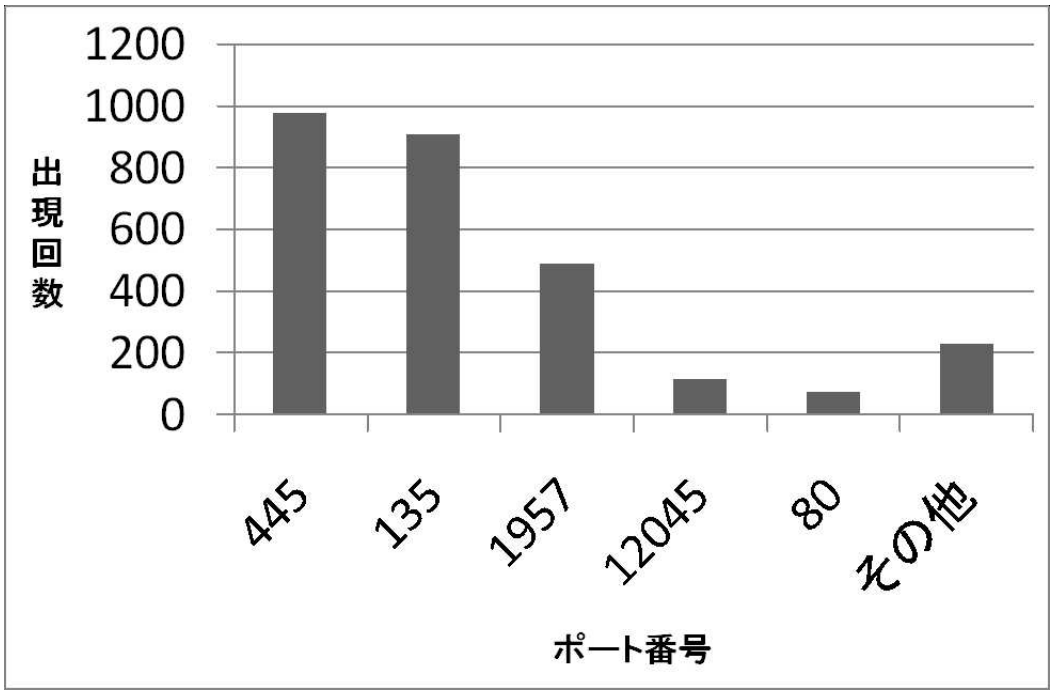

図 3.8: 送信先ポート番号分布 MWS 53760_4 （2009 年）

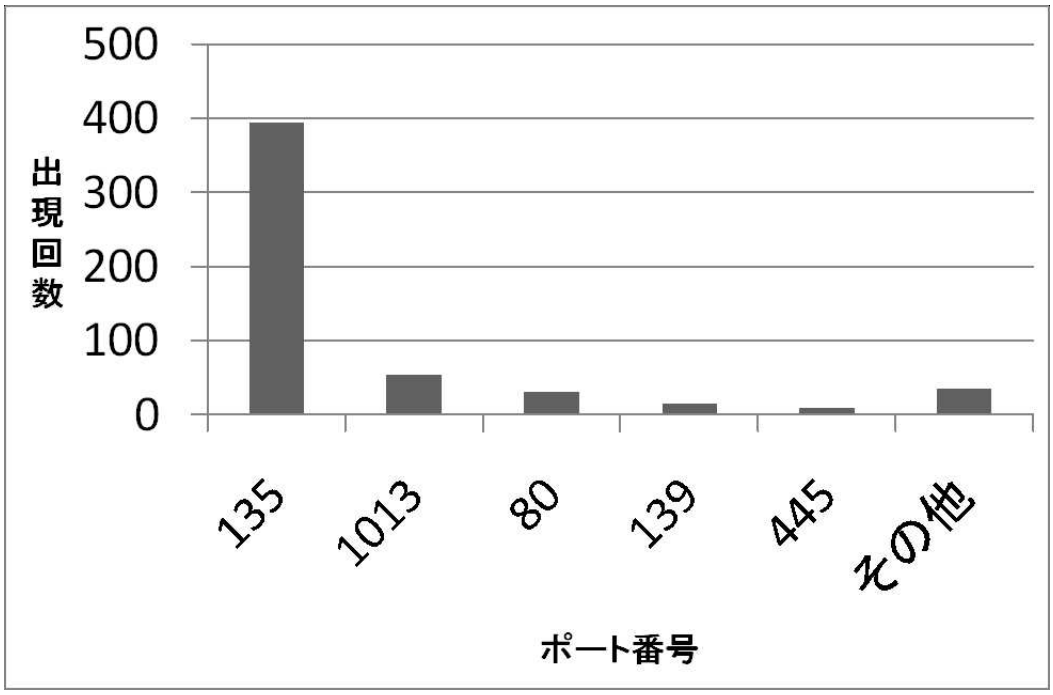

図 3.9: 送信先ポート番号分布 MWS 60352_3 （2009 年）

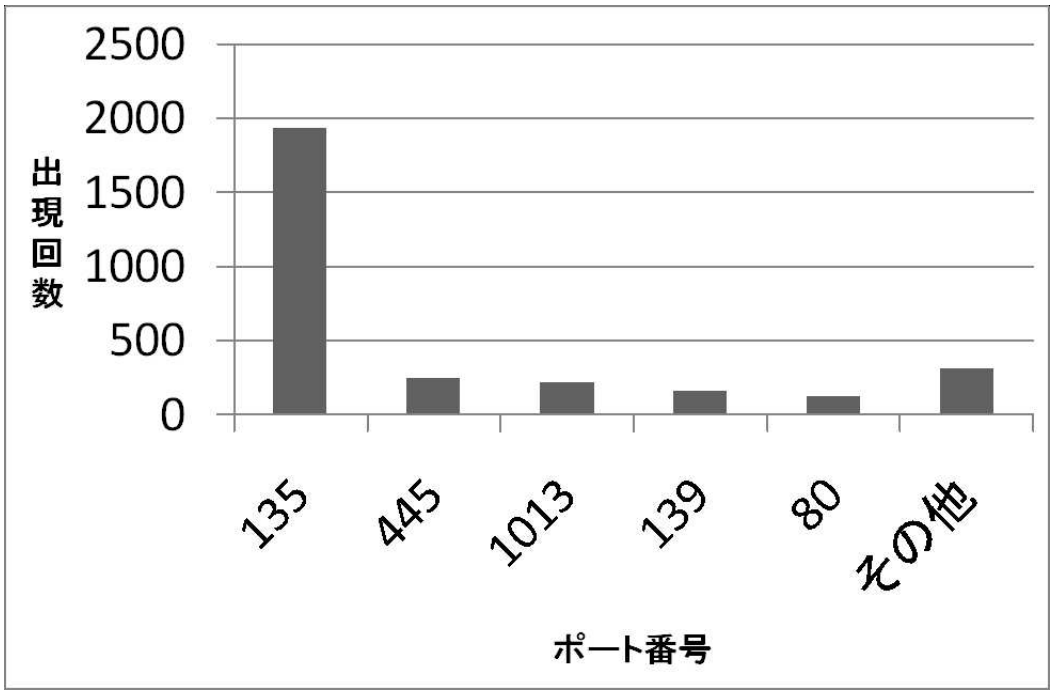

図 3.10: 送信先ポート番号分布 MWS 60352_6 （2009 年）

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。これは RPC のバッファ・オーバーフロー脆弱性を突いた攻撃を行っているものと推察される。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、その命令に従って ftp コマンドのやりとりによる実行ファイルの転送が行われている。

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
```

3.4.2 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,c925531e65920
6849bf7********************, PE_VIRUT.AV,C:\WINNT\system32\csrs.exe

3.4.3 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 MWS 16384_1 ではなく、 WindowsOS のシグネチャにて 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 SQL Server、 2967 番ポートは Symantec Client Security や SymantecAntiVirus）がハニーポット上で動いておらず、そのポートを LISTEN していなかったためと考えられる。

3.5 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 3.1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わかる。

表 3.1: 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	1,660	9,234	723

第 4 章

他のネットワークのデータへの応用

本章では、第 3 章にて抽出したシグネチャの拡張について、さらにシグネチャを用いて他の実運用ネットワーク通信データを分析した結果について述べる。

4.1 MWS シグネチャの拡張

4.1.1 DF ビット

IP ヘッダにおいて DF ビットが 1 である場合、そのパケットはフラグメンテーション（断片化）してはいけないということになる。通信経路上で MTU の値がパケットのサイズを下回っておりかつ DF ビットが 1 の場合、ルータはそのパケットを一度破棄して ICMP パケットでエラーを返すようになっている。送信元ホストはその ICMP のエラーを受けて MTU の値を通信可能なものに変更し、再送信を行う。このメカニズムをパス MTU ディスカバリという。

しかし、 IP マスカレード処理が行われたために ICMP パケットが送信元まで届かなかったり、経路の途中に ICMP パケットを破棄するように設定されたルータがあったりする場合、送信元が MTU の値を変更することができないため通信ができないままになってしまうという問題が起こる。一部のルータやファイアウォールには、この問題を回避するために DF ビットを削除する機能がある。

3.2 節において、インバウンド通信から抽出された MWS シグネチャはすべて DF ビットが 0 だったと述べたが、 CCC DATAset ではインバウンドのすべての通信において DF ビットが 0 であった。既存 OS でシグネチャの上では DF ビットが 1 のものについてもすべて 0 になっていたため、 CCC のハニーポットはインバウンドの IP パケットの DF ビットをすべて削除するという環境にある可能性が考えられる。実際、 MWS シグネチャを用いて他の通信データを分析したところ、 MWS シグネチャとは DF ビットのみが異なる通信が見られた。

この場合、本来 DF ビットが 1 だったのか 0 だったのかを推定することはできない。そこで、MWS シグネチャの DF ビットをすべて 1 にしたものを MWS+DF シグネチャとして作成した。

以下にその一覧を挙げる。

以降、 MWS+DF シグネチャについては MWS シグネチャよりも検出の優先度を低くして分析を行った。なお、 MWS+DF シグネチャも、 MWS シグネチャ同様に既存 OS のシグネチャとは合致しないシグネチャである。

```
16384:128:1:40:.:.:MWS+DF:16384_1
16384:128:1:44:M1414:.:MWS+DF:16384_2
16384:128:1:60:M1414,N,N,T0,N,N,S:.:MWS+DF:16384_3

32768:255:1:52:M1414,N,N,S,W0,N:.:MWS+DF:32768_1
32768:64:1:64:M1414,N,W0,N,N,T,S,E:P:MWS+DF:32768_2

44620:64:1:64:M1414,N,W3,N,N,T0,N,N,S:.:MWS+DF:44620_1

53760:64:1:64:M1360,N,W3,N,N,T0,N,N,S:.:MWS+DF:53760_1
53760:64:1:64:M1380,N,W3,N,N,T0,N,N,S:.:MWS+DF:53760_2
53760:64:1:64:M1398,N,W3,N,N,T0,N,N,S:.:MWS+DF:53760_3
53760:64:1:64:M1414,N,W3,N,N,T0,N,N,S:.:MWS+DF:53760_4
53760:32:1:64:M1414,N,W3,N,N,T0,N,N,S:.:MWS+DF:53760_41

5808:64:1:60:M1396,S,T,N,W0:.:MWS+DF:5808_1
5808:64:1:60:M1414,S,T,N,W0:.:MWS+DF:5808_2

60352:64:1:52:M1240,N,W2,N,N,S:.:MWS+DF:60352_1
60352:64:1:52:M1332,N,W2,N,N,S:.:MWS+DF:60352_2
60352:64:1:52:M1360,N,W2,N,N,S:.:MWS+DF:60352_3
60352:64:1:52:M1380,N,W2,N,N,S:.:MWS+DF:60352_4
60352:64:1:52:M1398,N,W2,N,N,S:.:MWS+DF:60352_5
60352:64:1:52:M1414,N,W2,N,N,S:.:MWS+DF:60352_6

65535:128:1:48:M1414,N,N,N,N:.:MWS+DF:65535_1
65535:64:1:48:M1414,N,N,S:.:MWS+DF:65535_2
65535:64:1:52:M1400,N,W2,N,N,S:.:MWS+DF:65535_3
65535:64:1:52:M1412,N,W2,N,N,S:.:MWS+DF:65535_4
```

```
65535:64:1:52:M1414,N,W0,N,N,S:.:MWS+DF:65535_5
65535:128:1:52:M1414,N,W2,N,N,N,N:.:MWS+DF:65535_6
65535:64:1:52:M1414,N,W2,N,N,S:.:MWS+DF:65535_7
65535:255:1:52:M1414,N,W2,N,N,S:.:MWS+DF:65535_71
65535:64:1:52:M1414,N,W3,N,N,S:.:MWS+DF:65535_8
65535:128:1:60:M1398,N,N,T0,N,N,S:.:MWS+DF:65535_9
65535:64:1:64:M1380,N,W3,N,N,T0,N,N,S:.:MWS+DF:65535_10
65535:64:1:64:M1414,N,W0,N,N,T0,N,N,S:.:MWS+DF:65535_11
65535:64:1:64:M1414,N,W2,N,N,T0,N,N,S:.:MWS+DF:65535_12
65535:64:1:64:M1414,N,W3,N,N,T,S,E:P:MWS+DF:65535_13
65535:64:1:64:M1414,N,W3,N,N,T0,N,N,S:.:MWS+DF:65535_14

8192:128:1:40:.:.:MWS+DF:8192_1
8192:128:1:56:M1414,S,T:.:MWS+DF:8192_2
8192:64:1:64:M1408,N,W0,N,N,T0,N,N,S:.:MWS+DF:8192_3
8192:64:1:64:M1414,N,W0,N,N,T0,N,N,S:.:MWS+DF:8192_4

8576:64:1:48:M1414,N,N,S:.:MWS+DF:8576_1

S122:32:1:64:M536,N,W0,N,N,T0,N,N,S:.:MWS+DF:S122_1

S4:64:1:52:M1414,N,N,S,N,W8:.:MWS+DF:S4_1

S44:128:1:44:M1414:.:MWS+DF:S44_1
S44:128:1:48:M1460,N,N,S:Z:MWS+DF:S44_2
S44:64:1:60:M1414,S,T,N,W13:.:MWS+DF:S44_3
```

4.1.2 最大セグメントサイズオプション

MWS シグネチャのうち、最大セグメントサイズオプションの値のみが異なるものがあった。以下の 6 群である。

```
53760:64:0:64:M1360,N,W3,N,N,T0,N,N,S:.:MWS:53760_1
53760:64:0:64:M1380,N,W3,N,N,T0,N,N,S:.:MWS:53760_2
53760:64:0:64:M1398,N,W3,N,N,T0,N,N,S:.:MWS:53760_3
```

```
53760:64:0:64:M1414,N,W3,N,N,T0,N,N,S:.:MWS:53760_4

5808:64:0:60:M1396,S,T,N,W0:.:MWS:5808_1
5808:64:0:60:M1414,S,T,N,W0:.:MWS:5808_2

60352:64:0:52:M1240,N,W2,N,N,S:.:MWS:60352_1
60352:64:0:52:M1332,N,W2,N,N,S:.:MWS:60352_2
60352:64:0:52:M1360,N,W2,N,N,S:.:MWS:60352_3
60352:64:0:52:M1380,N,W2,N,N,S:.:MWS:60352_4
60352:64:0:52:M1398,N,W2,N,N,S:.:MWS:60352_5
60352:64:0:52:M1414,N,W2,N,N,S:.:MWS:60352_6

65535:64:0:52:M1400,N,W2,N,N,S:.:MWS:65535_3
65535:64:0:52:M1412,N,W2,N,N,S:.:MWS:65535_4
65535:64:0:52:M1414,N,W2,N,N,S:.:MWS:65535_7

65535:64:0:64:M1380,N,W3,N,N,T0,N,N,S:.:MWS:65535_10
65535:64:0:64:M1414,N,W3,N,N,T0,N,N,S:.:MWS:65535_14

8192:64:0:64:M1408,N,W0,N,N,T0,N,N,S:.:MWS:8192_3
8192:64:0:64:M1414,N,W0,N,N,T0,N,N,S:.:MWS:8192_4
```

最大セグメントサイズ（MSS）の値は、通信環境によって左右されるところが大きい。たとえば、 Ethernet の場合は MTU が 1500 バイトであるため、その値から IP ヘッダ（20 バイト）と TCP ヘッダ（20 バイト）を除いた 1460 バイトが MSS の最大値となる。また、フレッツ ADSL では MTU が 1454 バイトであるため、 IP ヘッダと TCP ヘッダを除いた 1414 バイトが MSS の最大値となる。

p0f の設定ファイルには、新たなシグネチャを追加する場合の指針が書かれている。それによると、 MSS の値は接続に依存する（link-dependent）ものであり、固定値を用いる機器というまれなケースを除き常に値をワイルドカードとする（Always wiledcard this value, except for rare cases when you have an appliance with a fixed value）ということであった。

したがって、上記 6 群のシグネチャについては固定値ではないものと考えワイルドカードとし、MWS_Gen シグネチャとして以下 6 種のシグネチャを作成した。 Gen は Generic の略である。なお、前節の内容も鑑み、 DF ビットも 1 としている。

```
53760:64:1:64:M*,N,W3,N,N,T0,N,N,S:.:MWS_Gen:53760
5808:64:1:60:M*,S,T,N,W0:.:MWS_Gen:5808
60352:64:1:52:M*,N,W2,N,N,S:.:MWS_Gen:60352
65535:64:1:52:M*,N,W2,N,N,S:.:MWS_Gen:65535_1
65535:64:1:64:M*,N,W3,N,N,T0,N,N,S:.:MWS_Gen:65535_2
8192:64:1:64:M*,N,W0,N,N,T0,N,N,S:.:MWS_Gen:8192
```

最大セグメントサイズオプションが設定されているシグネチャは上記 6 群以外にもあったが、それらは固定値である可能性があるため、ワイルドカードとはしなかった。

以降、 MWS_Gen シグネチャについては MWS シグネチャ、 MWS+DF シグネチャよりも検出の優先度を低くして分析を行った。これにより、 MWS シグネチャ、 MWS+DF シグネチャとは MSS の値だけが異なる通信を MWS_Gen シグネチャとして検出することができる。

4.2 早稲田大学の通信データ

4.2.1 データの概要

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 12 月 25 日から 12 月 31 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． SYN パケットの総数は 236,011,263 であった。

4.2.2 分析結果

p0f を適用した結果を MWS シグネチャ、 MWS+DF シグネチャ、 MWS_Gen シグネチャ、各種 MWS シグネチャ以外の UNKNOWN、既存 OS の 5 種に分類したものが表 4.1、 4.2である。このうち、送信元 IP については複数種別のシグネチャで通信しているものがあるため、各種別の値を合計しても全体の送信元 IP 数とは一致しない。

MWS シグネチャについて、送信元 IP 数の割合は少ないにもかかわらず SYN パケット数が多いのは、少数の送信元 IP から MWS 16384_1 シグネチャによる大量の SYN パケットが送信されているためである。一つの送信元 IP あたりの SYN パケット数は、多いもので 100 万以上にもなる。この MWS 16384_1 シグネチャについての分析は後述する。

MWS シグネチャ、 MWS+DF シグネチャ、 MWS_Gen シグネチャについて、 SYN パケット数による順位を表 4.12に示す（上位 10 位まで）。 MWS 16384_1 シグネチャ以外では、 MWS_Gen

表 4.1: 早稲田大学の通信データ統計 (SYN パケット数)

	SYN パケット数	全体に占める割合
MWS	12,132,095	5.140%
MWS+DF	2,132,886	0.904%
MWS_Gen	6,062,306	2.569%
UNKNOWN	47,627,301	20.180%
既存 OS	168,056,675	71.207%
合計	236,011,263	100.000%

表 4.2: 早稲田大学の通信データ統計 (送信元 IP 数)

	送信元 IP 数	全体に占める割合
MWS	401	0.007%
MWS+DF	44,520	0.770%
MWS_Gen	269,373	4.656%
UNKNOWN	372,833	6.444%
既存 OS	5,222,683	90.272%
合計	5,785,478	100.000%

表 4.3: 各種 MWS シグネチャごとの SYN パケット数（早稲田大学）

シグネチャ	SYN パケット数
MWS 16384_1	12,058,445
MWS_Gen 65535_1	2,088,113
MWS_Gen 53760	1,398,351
MWS_Gen 60352	1,335,506
MWS_Gen 65535_2	1,101,716
MWS+DF 8192_1	952,104
MWS+DF 60352_6	244,401
MWS+DF 65535_13	241,613
MWS+DF 53760_4	194,928
MWS_Gen 8192	136,019
その他合計	576,091

シグネチャが総じて上位に来ていることがわかる。逆に、 MWS シグネチャのうち 22 種、 MWS+DF シグネチャのうち 11 種については SYN パケット数が 0 であった。

MWS 16384_1 シグネチャ以外に上位に来ている 4 シグネチャについて、送信先ポート番号を集計したのが表 4.4、 4.5、 4.6、 4.7である（上位 10 位まで）。 4 シグネチャのすべてで 1 位となっているのが 445 番である。これは前章でも述べたとおり、ポートスキャンの対象として著名なもののひとつである。 135 番、 139 番、 2967 番、 1433 番といったポートも同様である。

これ以外では、 HTTP/HTTPS （80 番、 8080 番、 443 番）に対するアクセスがあるほか、 CCC DATAset で見られなかったものとして SMTP （25 番）に対するアクセスがあることがわかる。また、 6886 番、 6889 番については bittorent で用いられるポートであることが知られているが、他のポートについては何をターゲットとしているか不明である。

MWS_Gen 53760 シグネチャと MWS_Gen 60352 シグネチャは 445 番ポートに対する SYN パケットの数が突出して多く、順位が下がるにつれて SYN パケット数が急激に減少しているが、 MWS_Gen 65535_1 シグネチャと MWS_Gen 65535_2 シグネチャは緩やかに減少している。

4.3 企業における SMTP 通信データ

4.3.1 データの概要

ある企業の電子メールサーバ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。収集時期は 2009 年 3 月 1 日から 3

表 4.4: MWS_Gen 65535_1 シグネチャの送信先ポート番号（早稲田大学）

ポート番号	SYN パケット数
445	1,827,882
80	120,384
6889	48,207
21053	11,828
8080	10,566
25	10,119
6649	9,778
28582	5,822
443	4,069
6886	3,009
その他合計	36,449

表 4.5: MWS_Gen 53760 シグネチャの送信先ポート番号（早稲田大学）

ポート番号	SYN パケット数
445	1,391,316
139	2,301
25	879
80	871
135	736
443	203
2967	66
21053	63
23032	27
1433	26
その他合計	1,863

表 4.6: MWS_Gen 60352 シグネチャの送信先ポート番号（早稲田大学）

ポート番号	SYN パケット数
445	1,322,036
1433	7,660
80	1,843
25	925
139	368
135	302
2967	134
6649	54
21053	51
6889	41
その他合計	2,092

表 4.7: MWS_Gen 65535_2 シグネチャの送信先ポート番号（早稲田大学）

ポート番号	SYN パケット数
445	533,677
80	378,072
6889	75,996
21053	20,479
6649	12,757
28582	10,356
443	7,436
8088	6,930
14229	6,648
25	5,000
その他合計	44,365

表 4.8: 企業における SMTP 通信データ統計 (SYN パケット数)

	SYN パケット数	全体に占める割合
MWS	253	0.004%
MWS+DF	55,160	0.794%
MWS_Gen	146,393	2.107%
UNKNOWN	930,768	13.398%
既存 OS	5,814,601	83.697%
合計	6,947,175	100.000%

表 4.9: 企業における SMTP 通信データ統計 (送信元 IP 数)

	送信元 IP 数	全体に占める割合
MWS	54	0.004%
MWS+DF	2,877	0.240%
MWS_Gen	42,030	3.500%
UNKNOWN	116,827	9.728%
既存 OS	1,053,875	87.758%
合計	1,200,882	100.000%

月 31 日の 1 カ月間であり、 TCP SYN パケットのみを収集の対象とした。 SYN パケットの総数は 6,947,175 であった。なお、 SMTP 通信であるので送信先ポート番号はすべて 25 番である。

4.3.2 分析結果

p0f を適用した結果を MWS シグネチャ、 MWS+DF シグネチャ、 MWS_Gen シグネチャ、 UNKNOWN、既存 OS の 5 種に分類したものが表 4.8、 4.9である。このうち、送信元 IP については複数種別のシグネチャで通信しているものがあるため、各種別の値を合計しても全体の送信元 IP 数とは一致しない。

MWS シグネチャ、 MWS+DF シグネチャ、 MWS_Gen シグネチャについて、 SYN パケット数による順位を表 4.10に示す（上位 10 位まで）。 MWS_Gen シグネチャが総じて上位に来ている点については早稲田大学の通信データと同様であるが、 MWS 16384_1 による SYN パケット数が 0 である点が大きく異なる。 MWS シグネチャのうち MWS 16384_1 を含む 30 種、 MWS+DF シグネチャのうち 18 種については SYN パケット数が 0 であった。

いくつかの MWS シグネチャによる SYN パケット送信があった IP アドレスについて、発信

表 4.10: 各種 MWS シグネチャごとの SYN パケット数（SMTP データ）

シグネチャ	SYN パケット数
MWS_Gen 65535_1	65,108
MWS+DF 8192_1	46,206
MWS_Gen 65535_2	40,396
MWS_Gen 60352	24,131
MWS_Gen 53760	13,004
MWS_Gen 8192	3,726
MWS+DF 65535_3	1,624
MWS+DF 60352_3	1,306
MWS+DF 65535_4	1,163
MWS+DF 65535_7	1,016
その他合計	4,126

したメールの内容を判別したのが表 4.11である。数は少ないが、これらの IP アドレスから送信されたメールがスパムメールであり、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と推定できる。

4.4 MAWI データセット

4.4.1 データの概要

MAWI データセットとは、 WIDE Project によって行われているインターネットの定点観測において取得されているものである。いくつかの種類のデータが存在するが、今回利用したのは samplepoint-F である。これは太平洋を横断するネットワーク回線において毎日 14:00 〜 14:15 の 15 分間取得されているパケットのフルキャプチャであり、統計データとともに公開されている。今回用いたのは 2006 年 11 月分から 2009 年 11 月分の 37 カ月間のデータである。

4.4.2 分析結果

このデータでは長期間の状態の変化を比較することを主な目的として、 1 カ月ごとに集計をしたうえで分析を行った。月ごとに日数が違うことに加え、データがない日があることから、基本的にはパケット等の数そのものではなく全体に占める割合によって比較を行う。

MWS シグネチャ、 MWS+DF シグネチャ、 MWS_Gen シグネチャについて、 SYN パケット数の割合の推移を示したのが図 4.1、 4.2である。 MWS+DF シグネチャの割合が低い一方で MWS

表 4.11: MWS シグネチャによる SYN パケット送信があった IP アドレスからのメールの統計

シグネチャ名	スパム	通常メール	送信元 IP アドレス数
65535_8	290	0	9
65535_5	252	0	8
65535_3	90	0	4
65535_7	64	0	6
16384_3	25	0	3
65535_4	16	0	7
53760_4	16	0	2
65535_12	9	0	1

シグネチャの割合が高いことがわかるが、すべての月において MWS シグネチャによる SYN パケットのおよそ 99% 以上が MWS 16384_1 シグネチャによるものであった。 MWS 16384_1 シグネチャによる SYN パケットのみをカウントした結果が図 4.3、 4.4、 MWS 16384_1 シグネチャによる SYN パケットを除いた結果が図 4.5、 4.6である。 MWS 16384_1 シグネチャ以外の MWS シグネチャによる SYN パケットは、存在しているもののほぼ 0% である。また、月による変動が激しいが、多くの月において MWS+DF シグネチャ、 MWS_Gen シグネチャによる SYN パケットが全体のおよそ 1.5% 以上存在していることがわかる。

送信元 IP の割合の推移を示したのが図 4.7、 4.8である。 MWS シグネチャの送信元 IP はどの月でも 0.1% 未満であり、非常に少ない。しかしながら、 MWS+DF シグネチャ、 MWS_Gen シグネチャの送信元 IP を合計すると、 2007 年 7 月を除いては 2% を上回っており、 SYN パケット数の割合の推移よりも安定して推移していると言える。 2007 年 7 月以降は増加傾向にあり、 2009 年は 4 〜 5% で推移している。

MWS シグネチャ、 MWS+DF シグネチャ、 MWS_Gen シグネチャについて、 37 カ月の合計 SYN パケット数による順位を表 4.10に示す（上位 10 位まで）。 MWS 16384_1 シグネチャと、 MWS_Gen シグネチャが上位に来ている点は早稲田大学の通信データと類似している。

上位に来ている MWS_Gen シグネチャ 4 種について、 SYN パケット全体に占める割合の推移を示したのが図 4.9、 4.10である。 MWS_Gen 53760 による通信は 2007 年 4 月と 2007 年 5 月に突出して多くなっているがその後は低い割合で推移している。逆に、 MWS_Gen 65535_1 と MWS_Gen 65535_2 は最近になるにつれて割合が増えていることがわかる。

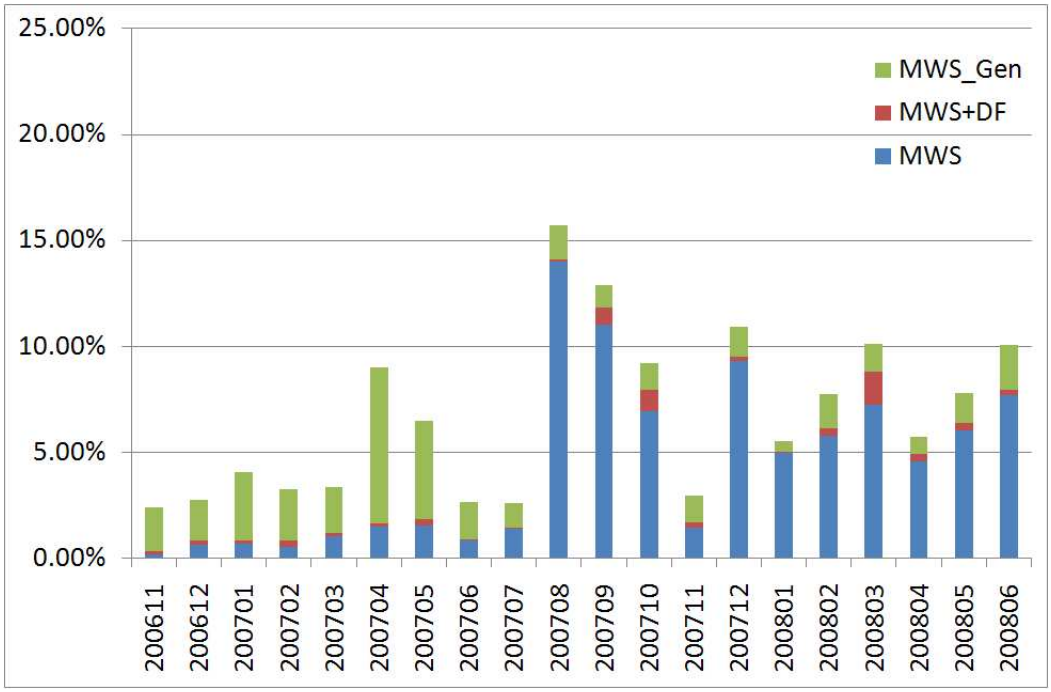

図 4.1: 各種 MWS シグネチャによる SYN パケット割合の推移（MAWI） 1

4.5 MWS 16384_1 シグネチャ

早稲田大学、 MAWI の双方で MWS 16384 シグネチャによる大量の SYN パケットが観測された。ここでは特にそのシグネチャによる通信のみに焦点を絞った分析を行う。 MAWI データセットにおいては、最も新しいデータである 2009 年 11 月のデータを分析の対象とする。

まず、 SYN パケット数と送信元 IP 数を表 4.13に示す。

一つの送信元 IP から、 MAWI データセットでは平均 19000 程度、早稲田大学の通信データでは平均 72000 程度の SYN パケットが送信されていることになる。実際には、 SYN パケット数は各送信元 IP によって大きく異なっており、 MAWI データセットでは 1 〜 196601、早稲田大学の通信データでは 1 〜 1768785 となっている。分布は一様でなく、 2 の階乗付近の数になることが多くみられた。これは、あるネットワークセグメントに対するスキャン行為である可能性をうかがわせるデータであり、実際にそのような通信が行われていることが確認できた。

送信先ポート番号を表 4.14、 4.15に示す（上位 8 位まで）。上位に来ているポート番号のうち、 1521 番は Oracle データベースで用いられるポート番号であり、 3306 番は MySQL で用いられるポート番号である。

このシグネチャによる通信のうち最も特徴的なのが、大多数の SYN パケットの送信元ポートが 6000 番であることである。その割合は、 MAWI データセットにおいては 94%、早稲田大学の通信データでは 96% 程度になる。 CCC DATAset でも改めて調査したところ、送信元ポート

表 4.12: 各種 MWS シグネチャごとの SYN パケット数（MAWI）

シグネチャ	SYN パケット数
MWS 16384_1	46,226,393
MWS_Gen 65535_2	3,977,229
MWS_Gen 53760	3,709,951
MWS_Gen 65535_1	2,844,464
MWS_Gen 8192	2,286,077
MWS+DF 8192_1	1,267,833
MWS_Gen 60352	751,435
MWS+DF 53760_4	463,724
MWS+DF 65535_12	254,946
MWS+DF 65535_7	209,522
その他合計	1,121,401

表 4.13: MWS 16384_1 シグネチャによる SYN パケットと送信元 IP 数

通信データ	SYN パケット数	送信元 IP 数
早稲田大学	12,058,445	166
MAWI （2009 年 11 月）	2,030,839	106

表 4.14: MWS 16384_1 シグネチャによる SYN パケットの送信先ポート番号（早稲田大学）

ポート番号	SYN パケット数
2967	5,827,791
1433	2,968,309
135	1,460,904
3306	344,411
1521	223,939
8088	201,510
8080	196,786
445	84,127
その他合計	750,668

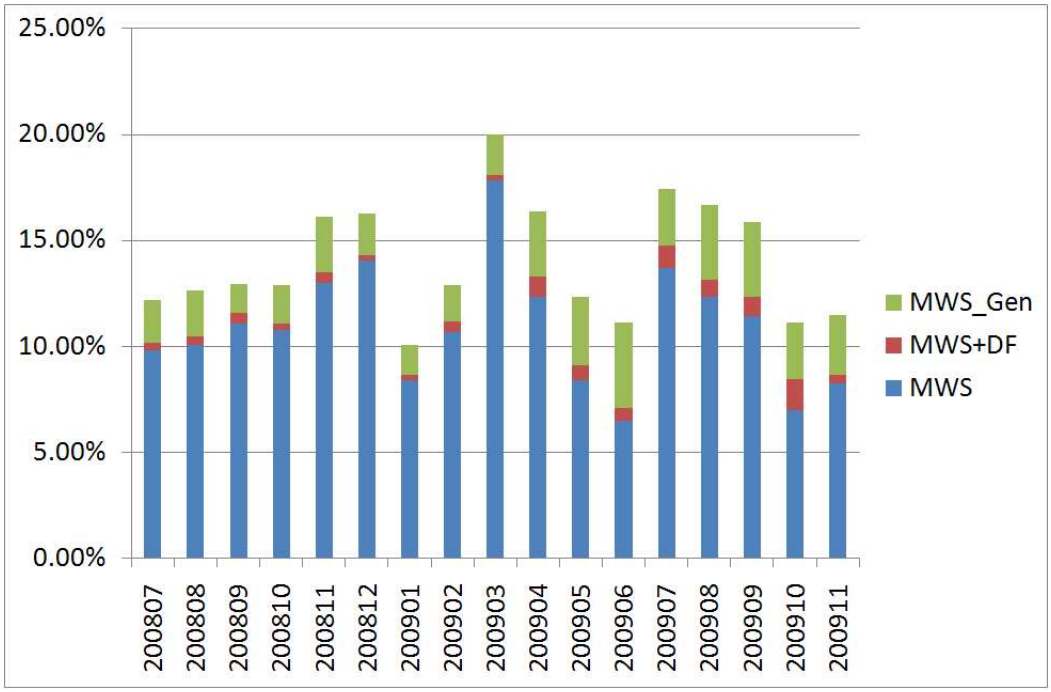

図 4.2: 各種 MWS シグネチャによる SYN パケット割合の推移（MAWI） 2

の大多数が 6000 番であるという結果が得られた。

CCC DATAset では、この MWS 16384_1 シグネチャによる SYN パケット送信の後で既存 OS のシグネチャによる SYN パケット送信が行われるパターンが見られたが、 MAWI データセット、早稲田大学の通信データ双方で同じ現象が見られた。 MWS 16384_1 シグネチャと既存 OS のシグネチャの両方で SYN パケット送信があった IP アドレスは、 MAWI データセットにおいては 14、早稲田大学の通信データでは 13 存在した。

4.6 考察

これまでの分析結果についての考察を述べる。

4.6.1 MWS シグネチャの拡張について

MWS 16384_1 シグネチャを除けば、他の通信データ上での MWS シグネチャそのものによる通信はごく少数であった。しかし、 MWS+DF シグネチャ、 MWS_Gen シグネチャによる通信は一定の割合で存在することが分かった。これにより、 MWS 16384_1 以外の MWS シグネチャの DF ビットは本来 1 であり、 CCC DATAset の収集環境によって DF ビットが 0 になっている可能性が高い。

また、 MWS_Gen シグネチャについては、通信先ポート番号の上位に著名な脆弱性のあるポー

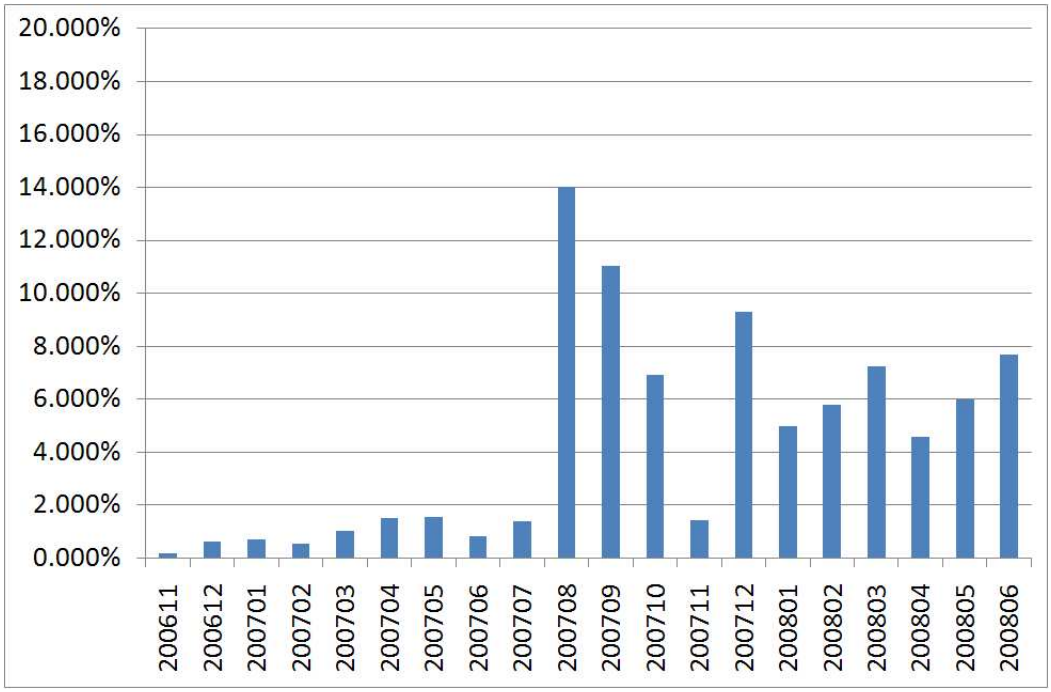

図 4.3: MWS 16384_1 シグネチャによる SYN パケット割合の推移（MAWI） 1

ト番号が存在することから、やはり悪意のある通信が行われていることが疑われる。

4.6.2 各データセット間の差異について

それぞれのデータセットにおけるシグネチャの出方には差異があった。

企業における SMTP 通信データとその他のデータにおける違いは送信先ポート番号が限定されているか否かであり、これは非常に大きな違いである。悪意のある通信の中でもスパムメール送信だけを対象にしたものと考えられるためである。

企業における SMTP 通信データ以外の 3 種間の差異として、データの収集時期の違いというのはもちろんあるのだが、ネットワーク環境の違いも影響していると考えられる。 CCC DATAset の収集に用いられたハニーポットは、一般の（個人向け） ISP に接続されている。これに対し、MAWI データセットの収集環境は、戦術の通り太平洋を横断するネットワーク回線であり、通信トラフィックの送信元が大きく異なる可能性が高い。

早稲田大学については学術用ネットワークと商用のネットワークの両方につながっているが、それ以上に IP アドレスの違いが大きいと考えられる。早稲田大学のネットワークアドレスは 133.9.0.0/16 と大きなものである。通常、マルウェアの感染活動は /16 のネットワーク内など IP アドレスで見たときに近い点に対して行われることが知られている。したがって、早稲田大学内のネットワークへの悪意のある通信は、感染ホストによる自動的なものよりも、ある程度意図的に行われているものの比率が高いことが考えられる。

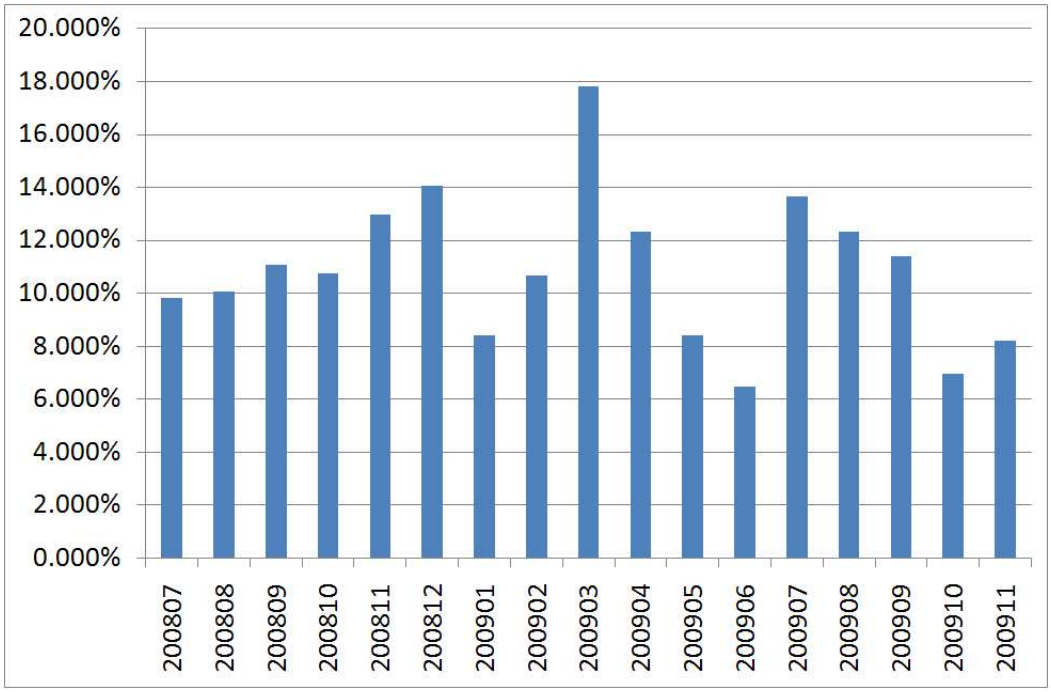

図 4.4: MWS 16384_1 シグネチャによる SYN パケット割合の推移（MAWI） 2

以上のようなデータセット間の違いが、シグネチャの出方の違いにつながっているものと考えられる。

4.6.3 MWS 16384_1 シグネチャについて

MWS 16384_1 については、 DF ビットを 1 にしたものよりも DF ビットが 0 のものの方が圧倒的に多数であったことから、このシグネチャについてはもともと DF ビットが 0 であったことが推測できる。

本シグネチャについては、 SYN パケットの量だけでなく、送信元ポート番号や、既存 OS のシグネチャによる通信が別途行われる点など特徴が多い。さらなる観測、分析を継続することで、このシグネチャが何によるものなのかという正体に迫ることができると考えられる。

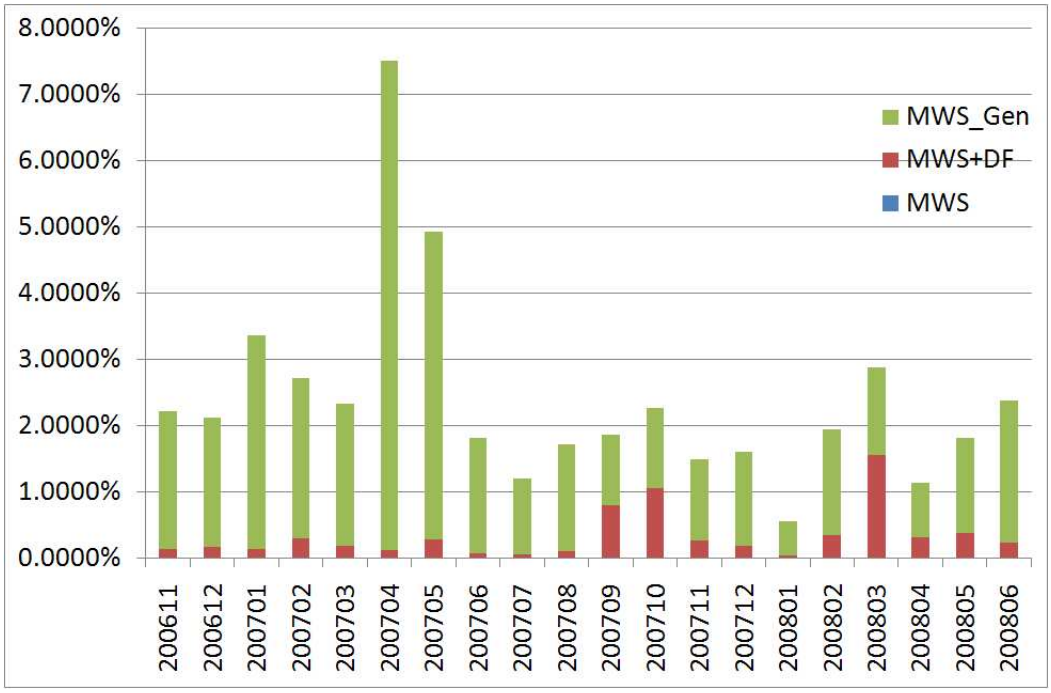

図 4.5: MWS 16384_1 シグネチャ以外の各種 MWS シグネチャによる SYN パケット割合の推移（MAWI） 1

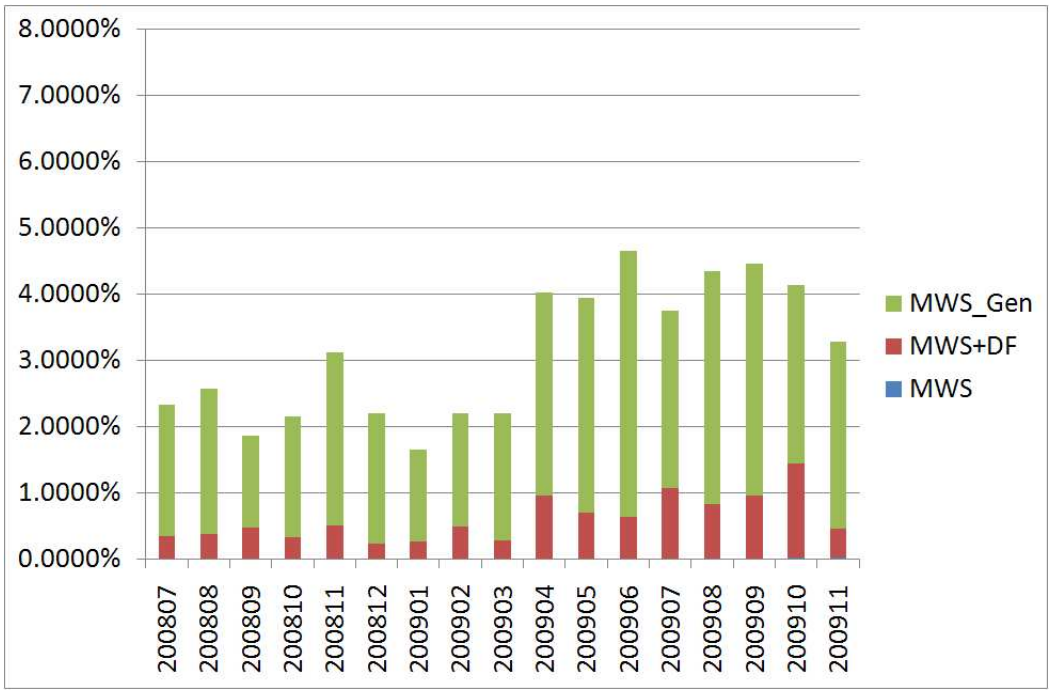

図 4.6: MWS 16384_1 シグネチャ以外の各種 MWS シグネチャによる SYN パケット割合の推移（MAWI） 2

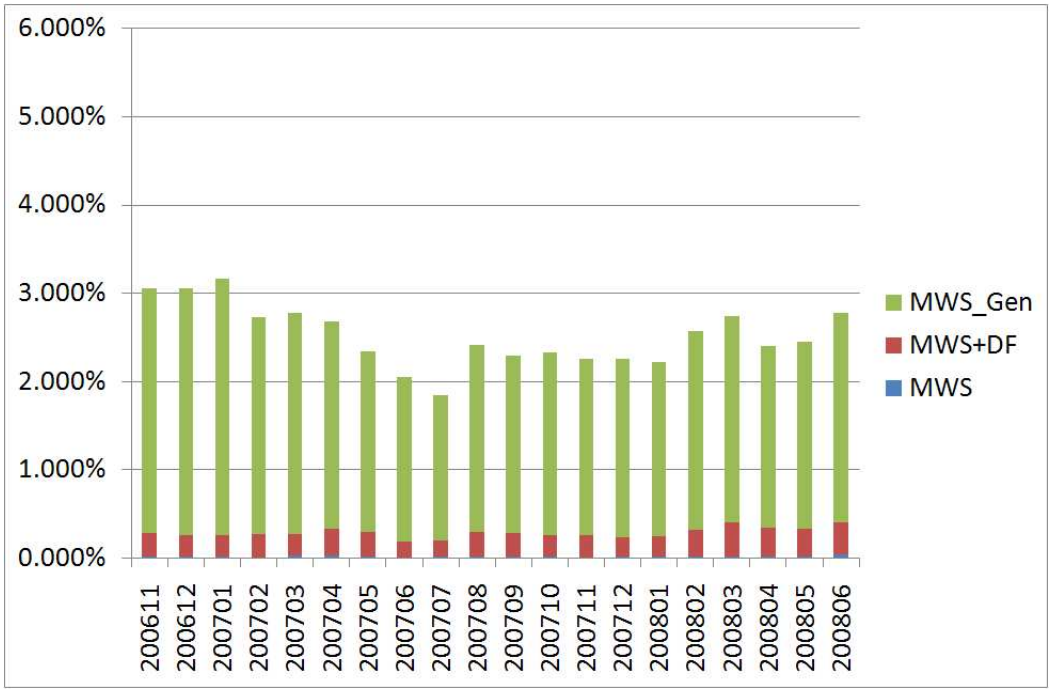

図 4.7: 各種 MWS シグネチャによる送信元 IP アドレス数の推移（MAWI） 1

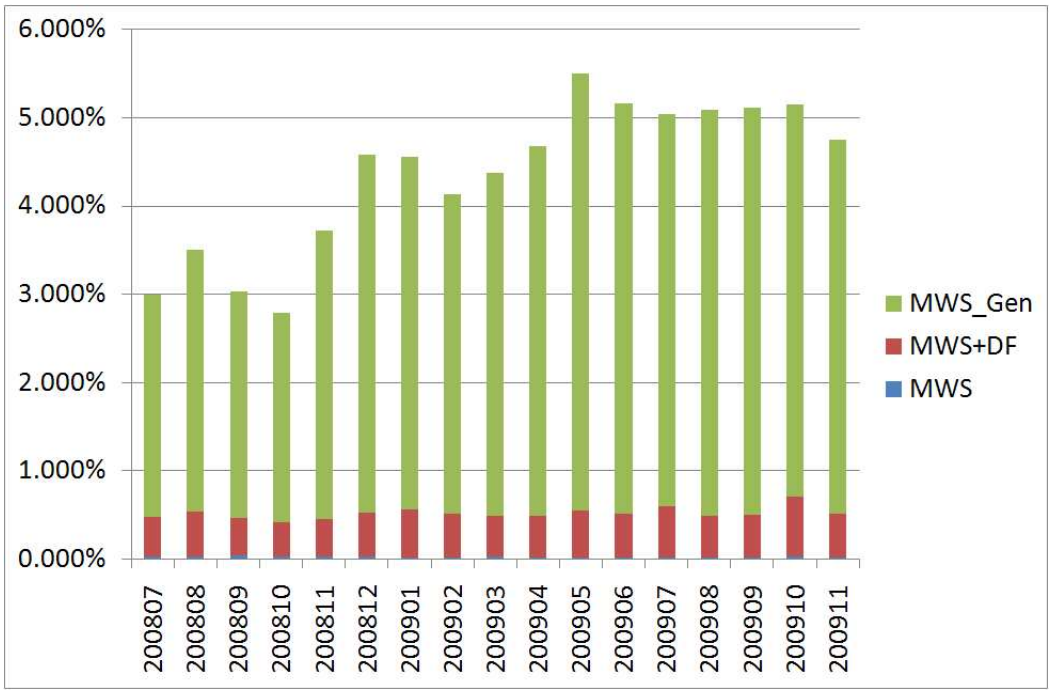

図 4.8: 各種 MWS シグネチャによる送信元 IP アドレス数の推移（MAWI） 2

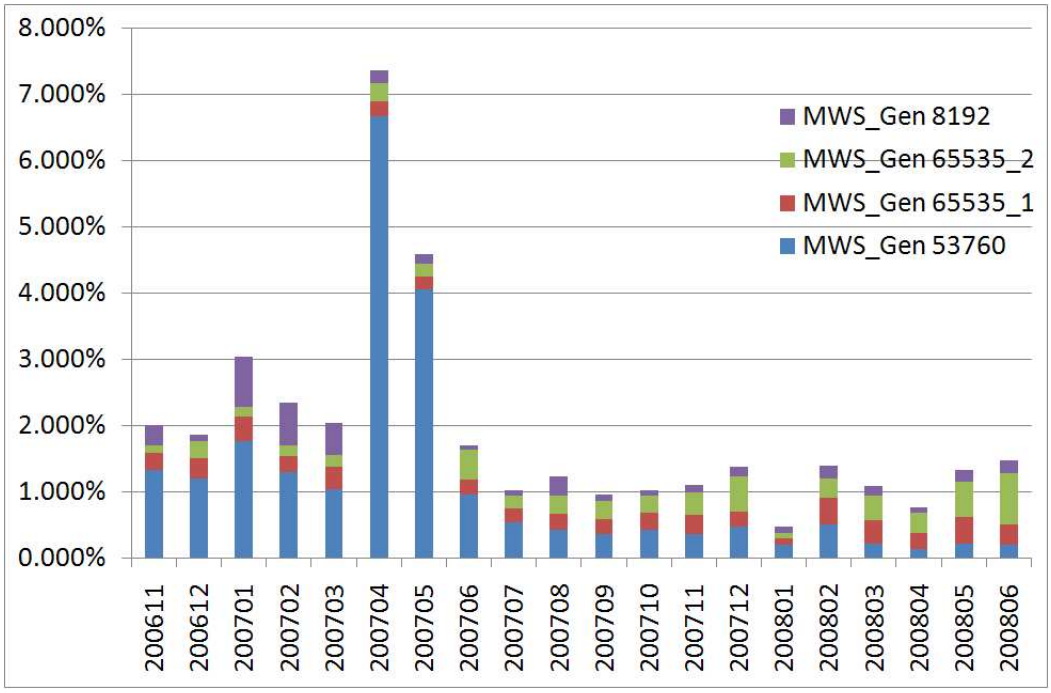

図 4.9: MWS_Gen シグネチャ 4 種による SYN パケット割合の推移（MAWI） 1

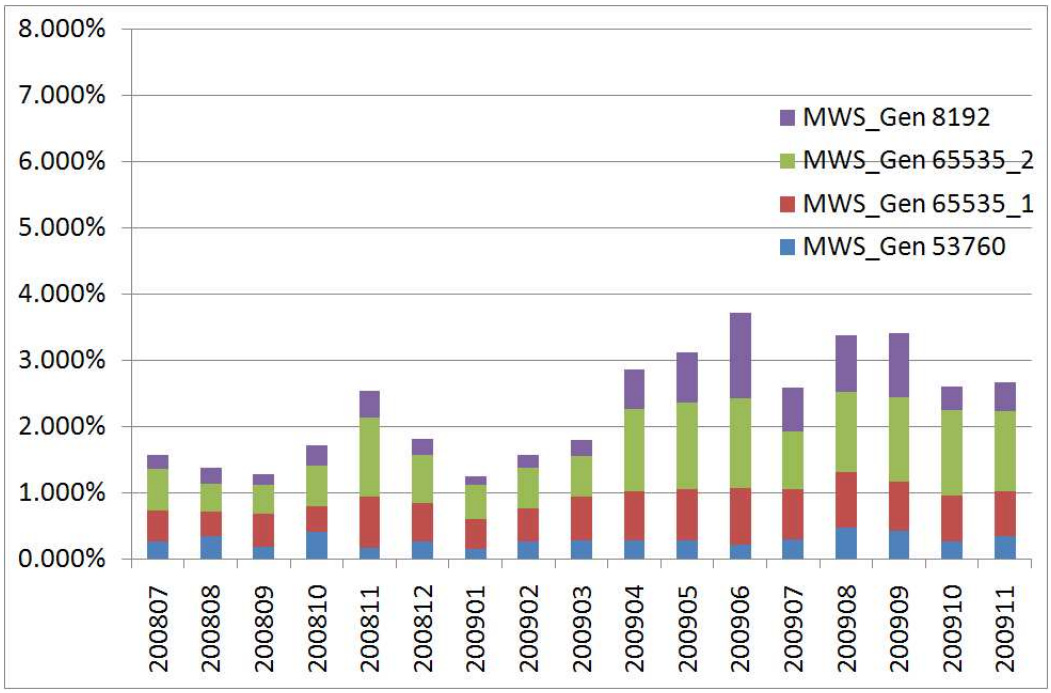

図 4.10: MWS_Gen シグネチャ 4 種による SYN パケット割合の推移（MAWI） 2

表 4.15: MWS 16384_1 シグネチャによる SYN パケットの送信先ポート番号（MAWI 2009 年 11 月）

ポート番号	SYN パケット数
1521	521,196
1433	502,537
2967	312,409
135	275,083
445	227,608
3306	70,647
8080	36,319
3128	32,218
その他合計	52,822

第 5 章

まとめ

本論文では、カーネルマルウェアの可能性がある通信を TCP フィンガープリントによって識別する手法を提案し、実ネットワーク上の通信データの分析を通して提案手法の有効性を検討した。ハニーポットに対する通信だけでなく、実ネットワーク上の通信データでも、作成したシグネチャによる悪意のある通信が行われている可能性が高いことを示すことができた。

本研究結果の実用途として、まずマルウェア感染ホストの特定に用いることが挙げられる。また、IDS などと組み合わせて、フィルタリングに役立てるということも考えられる。該当シグネチャによる通信である場合は、フィルタリングのためのスコアをプラスするという形である。

5.1 今後の課題

本論文では主に通信先ポート番号から悪意のある通信であることを示すという手法を取ったが、悪意のある通信であるかどうかを実際に判別するためにはパケットの内容を詳細にみる必要がある。既存の IDS などとの連携をすることで、よりパケットの中身に踏み込んだ分析を行い、今回抽出したシグネチャによってどのような通信が行われているのかをさらに検討する必要があると考えられる。

また、シグネチャの送信元ホストを特定し、そのホストが実際にマルウェアに感染しているかどうかの確認を行う必要がある。感染していた場合には、本論文の提案手法の有効性が高いことを証明できるからである。さらに、該当ホストの通信を分析することでシグネチャの正確性を増すことも可能である。

シグネチャそのものについても課題が残る。たとえば本研究を行うにあたって Windows 7 のシグネチャは用いなかったが、CCC DATAset 採取時に Windows 7 が未発売であったため、この点については問題はない。しかし、将来的には新しい OS による通信が増加していくことが予測される。これを更新しない限り、UNKNOWN のシグネチャがカーネルマルウェアによるものであるとは言えない可能性が高まる。したがって、既存 OS のシグネチャを随時アップデート

する必要がある。

謝辞

本修士論文の作成にあたり、日頃より御指導を頂いた早稲田大学理工学部の後藤滋樹教授に深く感謝致します。

本論文のもととなりました MWS2009 提出論文にも多大なる御協力をいただきました森達哉、下田晃弘各氏に深く感謝いたします。また、 MWS2009 に参加する機会を与えて頂いた日立製作所の寺田真敏先生に感謝いたします。

最後に、多大なる御協力を頂きました後藤研究室の諸氏に感謝致します。

参考文献

[1] Philip Miller, 苅田幸雄監訳, 『マスタリング TCP/IP 応用編』, オーム社, 1998.

[2] W. Richard Stevens, 橘康雄訳, 井上尚司監訳, 『詳解 TCP/IP Vol.1 [新装版]』, ピアソン・エデュケーション, 2000.

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

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

[5] H. Esquivel, T. Mori, and A. Akella, "Router-Level Spam Filtering Using TCP Fingerprints: Architecture and Measurement-Based Evaluation", http://ceas.cc/2009/papers/ceas2009-paper-10.pdf, 2009.

[6] 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.

[7] H. Stern, "The Rise and Fall of Reactor Mailer", http://projects.csail.mit.edu/spamconf/SC2009/Henry_Stern/, 2009.

[8] R. Riley, X. Jiang, and D. Xu, "Multi-aspect profiling of kernel rootkit behavior", Fourth ACM european conference on Computer systems, pp47-60, Apr. 2009.

[9] Michal Zalewski, "the new p0f: 2.0.8", http://lcamtuf.coredump.cx/p0f.shtml, 2006.

[10] 畑田充弘, 中津留勇, 寺田真敏, 篠田陽一, "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", マルウェア対策研究人材育成ワークショップ 2009 (MWS2009), A1-1, pp1-8, 2009.

[11] 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”,
マルウェア対策研究人材育成ワークショップ 2009 (MWS2009), A6-2, pp553-558, 2009.