東海大学大学院2011年度 修士論文

マルウェアの感染を判定する
発見的手法について

Heuristics for Detecting Malware Attacks

指導教員 菊池 浩明 教授

東海大学大学院 工学研究科 情報理工学専攻

0BDRM018 桑原 和也

目 次

第1章　序論

1.1　背景

インターネットが普及し利便性の高いサービスが提供される一方で，悪意のあるソフトウェアであるマルウェア (以下，MW) による被害が深刻化している．MWの攻撃手法は多様化・複雑化が進んでおり，感染時の振る舞いにも大きな変化がみられる．MWの攻撃手法には，数 10〜数 100 万台の PC を従えたボットネットによる不正行為や Web サイト経由でのマルウェア感染による攻撃である Drive-by-Download 攻撃などが存在する．

ボットネットとは，ボットと呼ばれるMWに感染した PC により構成され，指令者から遠隔操作によって命令を受け機能を実現する．指令者は，IRC[1)]サーバなどを介してボットネットに命令を送る．これにより，複数のボットを一斉に動作させる．このようにボットネットは容易に検出されない様に，複雑で高度な感染方式を用いる．ボットネットに対しての取り締まりでは，2011 年 11 月に 400 万台のボットネットが FBI，警察，トレンドマイクロなど様々な関係者の手により閉鎖された [1]．今日，ボットネットはスパムメールの大量送信，DDoS 攻撃，大規模な感染活動など様々なセキュリティインシデントの源泉となっている．

Drive-by-Download 攻撃とは，ユーザーが Web サイトを閲覧しただけで自動的にMWをダウンロードする攻撃である．Web サイトの中には日本の有名企業のサイトも改竄が行われており，2009 年に Gunblar という名称でメディアでも大きく取り上げられた．2009 年の Gumblar では，JR 東日本，ホンダ，ローソンなどの日本の企業サイトが改竄にあった．各ベンダーも Web サイト改竄に関する情報，危険なドメインに対する調査・対策が行われている [2][3][4].

Drive-by-Download 攻撃は，主に Web ブラウザや OS の脆弱性を狙い行われる．攻撃の中にはゼロデイの脆弱性を利用したものも存在し対策は困難である．

1.2　目的

そこで，本論文ではボットネットに関する検知手法の提案と Web 感染型マルウェアである Driveby-download 攻撃の分類を行う．検知精度を明らかにし，改善方法について考察する．

[1)] Internet Relay Chat システム．ボットネットの命令を送信するチャンネルとして多用されている．

1.3 論文構成

本論文の構成は次の通りである．第 2 章では，研究に関連する技術について述べる．第 3 章では，研究に使用した実験データについて述べる．第 4 章では，ボットネットの連携感染に関する検知手法について述べる．第 5 章では，Web 感染型マルウェアである Drive-by-download 攻撃の分類について述べる．第 6 章では，Web 感染型マルウェアである Drive-by-download 攻撃の検知手法について述べる．最後に第 7 章で結論と今後の課題について述べる．

第2章
研究に関連する技術

2.1 解析ツール (ボットネット)

2.1.1 BotHunter

BotHunter は，MW に感染した PC の通信パターンを認識する受動的なネットワーク監視ツールとして Guofei Gu 等によって設計された．Snort を用いて，通信データを分析し，既知のボットネット攻撃を検出することが出来るツールである [5]．図 2.1 は Bothunter の実行画面である．

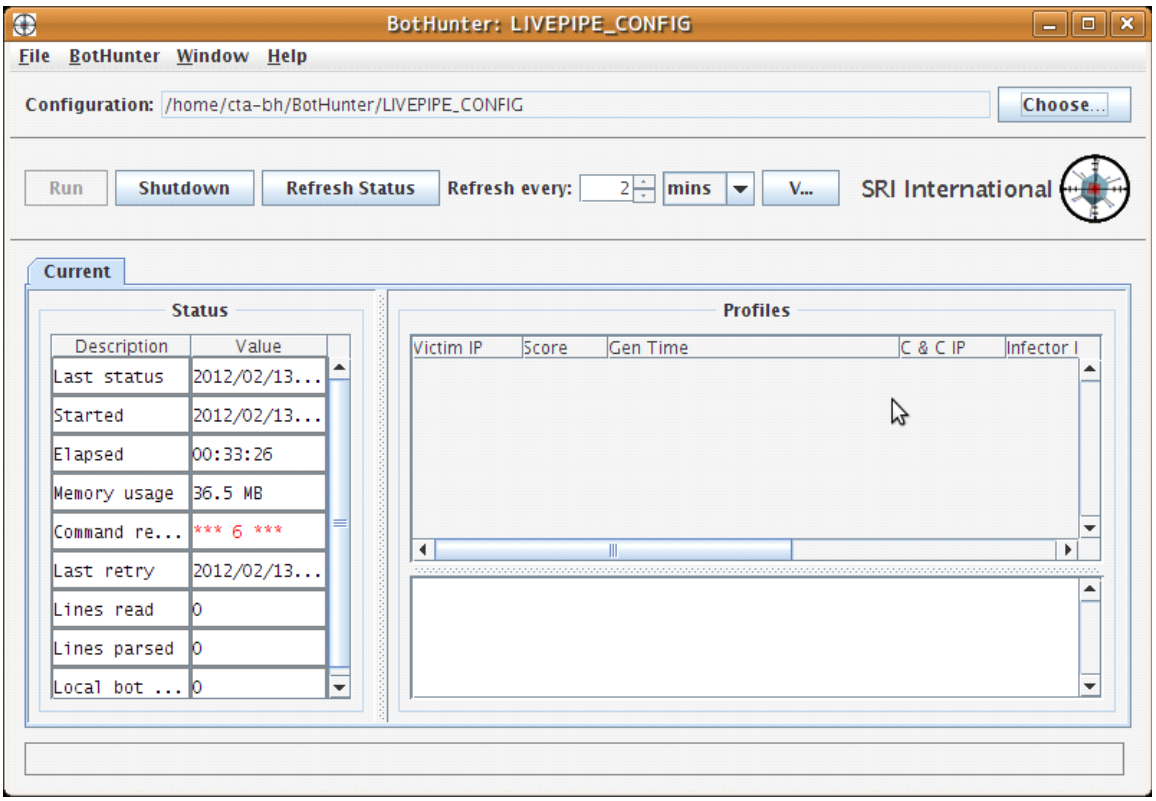


図 2.1: Bothunter の実行画面

2.2 解析ツール (Web 感染型マルウェア)

2.2.1 jsunpack-n

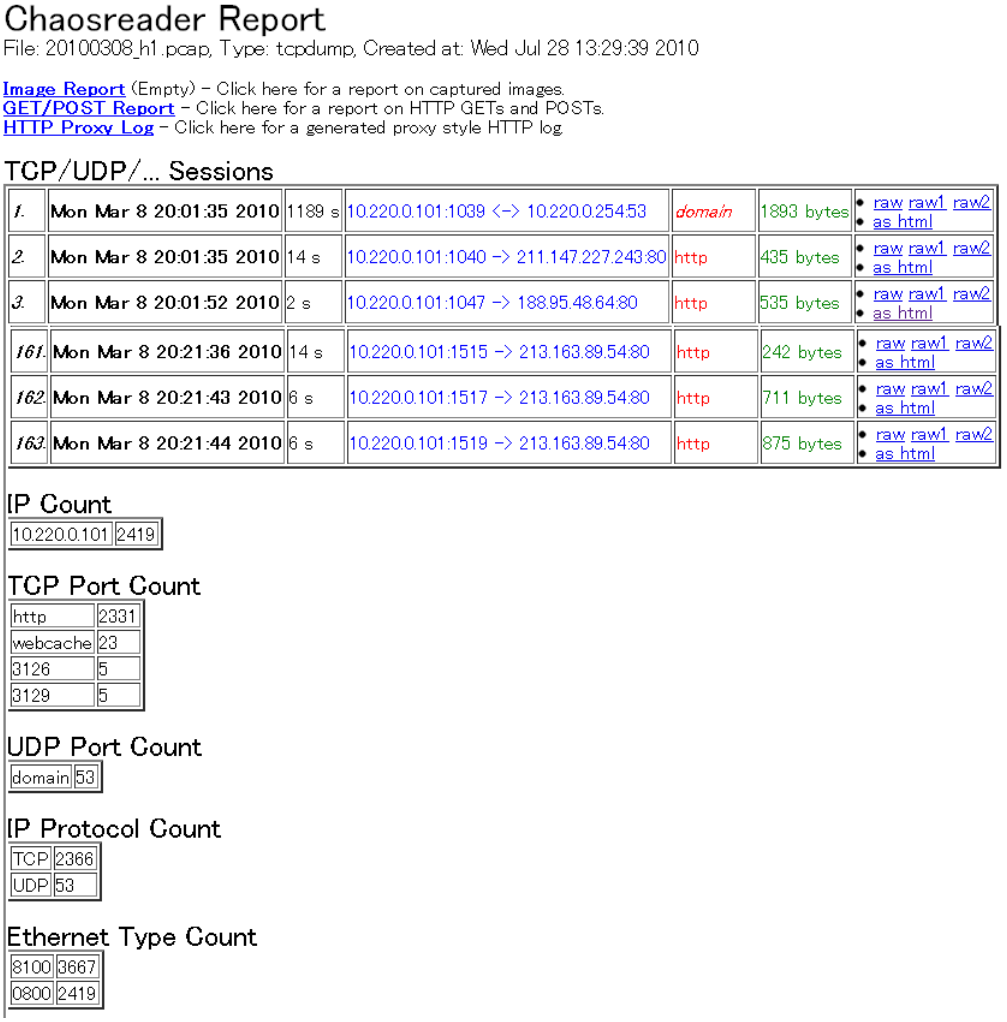
jsunpack-n は，Blake Hartstein によって開発されたマルウェアアンパックツールである．パケットキャプチャデータから JavaScript や shellcode 、exe を抽出することが出来る python のツールである．通信の遷移をテキストと図の両方で出力することが出来る．図 2.2 は，攻撃の流れをテキストで表示させたものである．テキストは，改行により同じ URL であるかどうかの区別を行っている．また，オンライン版も存在する．

```
[nothing detected:children=malicious:10] /home/hirt/jsunpack-n/D3M2010/D3M20100308/20100308_006_006.pcap
  [malicious:7] (ipaddr:109.196.143.31) GET securixp.com/s/
    info: [meta refresh] URL=securixp.com/s/index.php?s=1
    info: [decodingLevel=0] found JavaScript
    info: DecodedGenericCLSID detected F0E42D50-368C-11D0-AD81-00A0C90DC8D9 BD96C556-65A3-11D0-983A-00C04FC29E36
    malicious: MSOfficeSnapshotViewer CVE-2008-2463 detected F0E42D50-368C-11D0-AD81-00A0C90DC8D9
    info: [javascript variable] URL=securixp.com/s/exe.php?s=17
    info: [var mokeso] URL=securixp.com/s/
    info: [var newurl] URL=securixp.com/s/
    info: [decodingLevel=1] found JavaScript
    info: file: saved securixp.com/s/ to (./files/original_bd5cf1b218726a64f93ef6883fd8d16bf4b5d46a)
    file: stream_bd5cf1b218726a64f93ef6883fd8d16bf4b5d46a: 47067 bytes
    file: decoding_7d0e2ed34ec4426fe5009888218dde9551ece165: 58352 bytes
    [malicious:10] (ipaddr:109.196.143.31) [PDF] GET (metarefresh) securixp.com/s/index.php?s=1
      info: [decodingLevel=0] JavaScript in PDF 26843 bytes, with 231 bytes headers
      info: [decodingLevel=1] found JavaScript
      malicious: Utilprintf CVE-2008-2992 detected
      malicious: collectEmailInfo CVE-2007-5659 detected
      malicious: CollabgetIcon CVE-2009-0927 detected
      suspicious: Warning detected //warning CVE-NO-MATCH Shellcode Engine Binary Threshold //warning CVE-NO-MATCH
      malicious: shellcode of length 618/324
      malicious: shellcode URL=securixp.com/s/exe.php?s=24
      info: [decodingLevel=2] found JavaScript
      suspicious: Warning detected //warning CVE-NO-MATCH Shellcode NOP len 2096798
      info: file: saved securixp.com/s/index.php?s=1 to (./files/original_c7c44455a9725e0b5eb434ee6fd9a4bd98fdf9c0)
      file: stream_c7c44455a9725e0b5eb434ee6fd9a4bd98fdf9c0: 27954 bytes
      file: decoding_a799168a0949577db9591d774f3a0b715f315a6e: 27074 bytes
      file: decoding_35b09cce2a88ce261e6653e839b011c9ddc05d5e: 12630 bytes
      file: shellcode_8d2e9762e4e784ddbf0d8df535a947ada0700c4c: 618 bytes
      file: decoding_f12f164999cb8ae302a4b252c8aae245721f2deb: 4465 bytes
      [not analyzed] (shellcode) securixp.com/s/exe.php?s=24
    [nothing detected] [MZ] GET (jsvar) securixp.com/s/exe.php?s=17
      info: [0] executable file
      file: stream_f7098736606b7026de4fd65c48d933e752cf5f91: 186368 bytes
    [not analyzed] (var newurl) securixp.com/s/
    [nothing detected] GET securixp.com/s/%u9090%u9090%u9090%u9090%u9090%u9090%u9090%u9090
      file: stream_bd64e6c3b451e81bdaf616158e5f6b6fbfd9b7b3: 374 bytes
      file: stream_441027fd9dbf361cc822a11929b804b26b0b74d9: 375 bytes
```

図 2.2: jsunpack-n の出力結果 (テキスト)

2.2.2 Chaosreader

オーストラリアの Brendan Gregg によって開発された TCP セッションを HTML レポート化する Perl ツールである [8]．図 2.3 は Chaosreader の出力結果である．



Chaosreader Report
File: 20100308_h1.pcap, Type: tcpdump, Created at: Wed Jul 28 13:29:39 2010

Image Report (Empty) – Click here for a report on captured images.
GET/POST Report – Click here for a report on HTTP GETs and POSTs.
HTTP Proxy Log – Click here for a generated proxy style HTTP log.

TCP/UDP/... Sessions

1.	Mon Mar 8 20:01:35 2010	1189 s	10.220.0.101:1039 <-> 10.220.0.254:53	domain	1893 bytes	raw raw1 raw2 as html
2.	Mon Mar 8 20:01:35 2010	14 s	10.220.0.101:1040 -> 211.147.227.243:80	http	435 bytes	raw raw1 raw2 as html
3.	Mon Mar 8 20:01:52 2010	2 s	10.220.0.101:1047 -> 188.95.48.64:80	http	535 bytes	raw raw1 raw2 as html
161.	Mon Mar 8 20:21:36 2010	14 s	10.220.0.101:1515 -> 213.163.89.54:80	http	242 bytes	raw raw1 raw2 as html
162.	Mon Mar 8 20:21:43 2010	6 s	10.220.0.101:1517 -> 213.163.89.54:80	http	711 bytes	raw raw1 raw2 as html
163.	Mon Mar 8 20:21:44 2010	6 s	10.220.0.101:1519 -> 213.163.89.54:80	http	875 bytes	raw raw1 raw2 as html

IP Count

10.220.0.101	2419

TCP Port Count

http	2331
webcache	23
3126	5
3129	5

UDP Port Count

domain	53

IP Protocol Count

TCP	2366
UDP	53

Ethernet Type Count

8100	3667
0800	2419

図 2.3: Chaosreader の出力結果

2.3 セキュリティ情報データベース

2.3.1 Virus Total

Virus Total は，スペインのセキュリティベンダー Hispasec Sistemas が運営する無償の Web サービスである．複数のウイルス対策エンジンを用いて一括でウイルスチェックできる [6].

2.3.2 aguse

aguse は，調査したいサイトの URL や受信したメールのメールヘッダーを入力することにより、関連する情報を表示するサービスである [7].

2.4 解析ツール (その他)

2.4.1 clamav

Clam Antivirus は Tomasz Kojm 等によって開発・メンテナンスされている UNIX 系のシステムで動作するアンチウイルスのフリーソフトである [9].

2.4.2 tcpflow

tcpflow は TCP 通信のデータに対し，プロトコル解析やデバッグを行うことが出来るツールである．UNIX 上で動作する．

2.4.3 Wireshark

Wireshark は，ネットワークアナライザのソフトウェアである．GUI の他，コマンドラインでも実行可能で，パケットキャプチャと解析機能がある．

第3章
実験データ

3.1 研究用データセット CCC DATAset

研究用データセット CCC DATAset とは，サイバークリーンセンター [11] で収集しているボット観測データ群である．配布しているデータの名称とデータ内容はつぎのとおりである．マルウェアの解析技術の研究のための「マルウェア検体」，感染手法の検知ならびに解析技術の研究のための「攻撃通信データ」，ボットの活動傾向把握の技術のための「攻撃元データ」の三つから構成される．

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．

3.1.2 攻撃通信データ

攻撃通信データは，2 台のハニーポットを用いて観測したボットネットとの通信を tcpdump でパケットキャプチャーした libpcap 形式のファイルである．ハニーポットは 1 台のホスト OS 上で動作する Windows 2000 と XP の 2 台のゲスト OS により構成されている．それぞれインターネット接続されており，パケットキャプチャーはホスト OS 上で行われている．

3.1.3 攻撃元データ

攻撃元データは，ハニーポットで記録したマルウェア取得時のログデータで，csv 形式のファイルである．攻撃元データの基本情報には，マルウェア検体の取得時刻，送信元 IP アドレス，送信元ポート番号，宛先 IP アドレス，宛先ポート番号，TCP または UDP，マルウェア検体のハッシュ値（SHA1），マルウェア名称，ファイル名がレコードとして記録されている．

3.2 研究用データセット D3M

D3M は，NTT 情報流通プラットフォーム研究所の高対話型の Web クライアントハニーポット (Marionette[12]) で収集したマルウェア検体，攻撃通信データの 2 つを収録した Web 感染型マルウェアの観測データ群である．Marionette は脆弱性に対する攻撃を受けるがダウンロードされたマルウェアの実行を許可しない．そのため，CCC DATAset の攻撃通信データとは異なり，感染後のマルウェアの通信挙動は D3M の攻撃通信データには含まれない．

3.2.1 マルウェア検体

Web クライアントハニーポットで収集した Web 感染型マルウェアのハッシュ値をテキスト形式で記載したファイルである．

3.2.2 攻撃通信データ

Web クライアントハニーポット 10 台の通信を tcpdump でパケットキャプチャした libpcap 形式のファイルである．ハニーポットの OS は WindowsXP SP2，ブラウザは Internet Explorer 6.0，プラグインが Adobe Reader ， Flash Player ， WinZip ，QuickTime，JRE であり，何れもセキュリティパッチは未適用である．10 台それぞれがインターネット接続されており，パケットキャプチャは上流ネットワークにあるスイッチのミラーポートで行っている．巡回対象 URL は公開されているブラックリスト（malwaredomainlist.com[13]）に登録されている URL の中から，各データ収集日に攻撃を検知した URL を予め抽出したものを用いており，参考情報として D3M2011 とともに提供している．各収集日においてアクセスした URL は同一とは限らず，また，入力 URL から派生する URL（リダイレクト，スクリプト読み込み，画像読み込みなど）は記載されていない．

第4章
連携感染を判定する発見的手法について

4.1 概要

近年，マルウェア (以下，MW) に感染した数 10〜数 100 万台の PC を従えたボットネットによる不正行為が深刻になっている．攻撃者のボットネットによる攻撃イメージを図 4.1 に示す．ボットネットは容易に検出されない様に，複雑で高度な感染方式を用いる．まず感染に用いるマルウェアはポートスキャンやバックドア設置などの機能毎に分割され，数多くの亜種が合成される．MW の配布も，数多くのダウンロードサーバ (以下，DL サーバ) に分散され[1)]，様々なプロトコルが用いられている．加えて感染の方式も複雑で，IRC などを介して動的にパターンが変更されたりする．この複雑な攻撃パターンを解析する為に様々な研究が行われている．例えば，水谷らは，ボットネットにおける状態遷移モデルを提案し，独自のファイル転送プロトコルの性質を報告している [15]．その他にも，中継ホストの活動期間やダウンロード関係の分布の解析 [16]，マルウェアのライフサイクルに着目した攻撃解析手法 [17]，通信プロトコルの種類と分類の研究 [18]，攻撃と DNS のクエリの相関に注目した研究 [20]，2 台のハニーポット間の連携を検出する研究 [21] など多くの研究報告がされている．

MW の検出を困難にしている大きな原因は，複数の連携した DL サーバによる多種類の MW を用いた感染方式である．このボットネットに特有の感染方式を本稿では連携感染と呼ぶ．連携感染には様々な効果がある．まず，不正サーバが多いので，ボットネット全体の特定が難しい．加えて，感染させる MW を変えるだけで攻撃パターンの再構成が可能である．例えば，本稿で後述する `WORM_SWTYMLAI.CD(WO3)` は，同時に感染する他の MW に依ってポートスキャンや DoS 攻撃のパターンが変わる[2)]．従って，ボットネットからの攻撃を検出し，そのパターンを判別するためには，もはや単一の MW の解析だけでは不十分であり，MW に感染した PC と複数の DL サーバ間での通信などを総合的に解析する必要がある．

連携感染は，2008 年の松木らの論文 [19] で既にその存在が報告されている．松木らは，連

[1)] 例えば，竹森は，1 つの MW が平均 2 台，最大 69 台の DL サーバから配布されていたことを報告している [14].

[2)] 後述する表 4.8 に示す様に，WO3 は 3 種の連携パターンの全てに用いられていた．それゆえ，MW 名が分かっても，種類からその先に生じる不正行為はポートスキャン (s4)，DoS，SMTP のどれであるか予測がつかなかった．この失敗が連携感染の重要性を認識することにつながった．

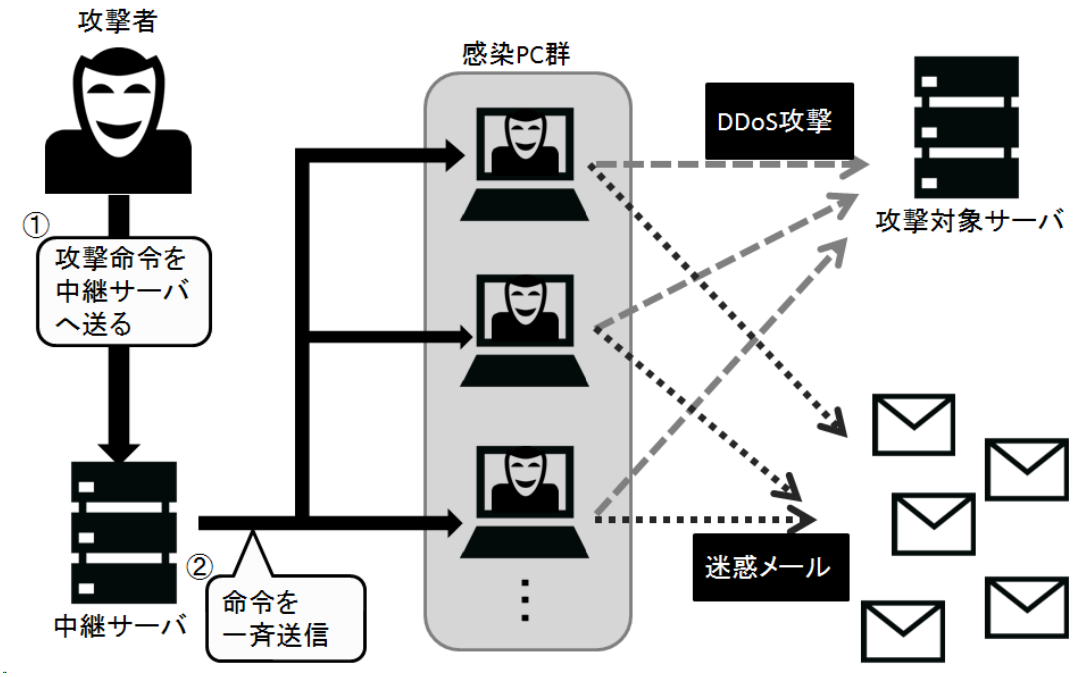


図 4.1: ボットネットによる攻撃のイメージ

携感染を定めるパラメータとして，1. 感染時間間隔，2. 攻撃元 IP アドレスの一致度，3. ソースポート番号の連続性，4. 検体名称，5. 検体ファイルサイズの 5 つを定義しているが，観測データ量の不足を理由に 1 の感染時間の間隔のみを時系列分析している．仮に十分な観測データが得られたとしても，5 つのパラメータの組み合わせは膨大で最適なパラメータを求めるのは困難であることが予測される．

そこで，本研究では，通信データから連携感染を検出する発見的手法を提案する．多くのパラメータの最適な組み合わせを求める代わりに，複数の検出ルールを組み合わせて，MW 名やソース IP アドレス，ポート番号，IRC 通信のメッセージなどの多くの情報を考慮した高精度の効率的な検出システムの実現を試みる．検出ルールを学習するデータとして，サイバークリーンセンター (以下，CCC) の 94 台のハニーポットで観測された通信データである CCC DATAset 2009 の攻撃通信データを用いる．キャプチャーされた攻撃通信データの中から，複数の DL サーバの連携により感染と攻撃が行われているパターンが存在すると仮定し，そのタイミング，ポート，MW の種類，通信先などの様々な連携感染固有の特徴を明らかにする．本論文はこの特徴に基づいて，(1) 連携感染パターンに関するルール，(2) 感染の有無に関するルール，(3) ポートスキャンなどの他の攻撃に関するルールから成る発見的手法を提案する．更に，特徴量の学習には用いなかった CCC DATAset 2009 の他の通信データを評価データとみなし，提案した発見的手法の精度を明らかにし，その有効性を検証する．

本研究と同様に，既知の通信パターンや不正ホストのアドレスリストを基にして感染を検出するシステムに，BotHunter，BotSniffer[22] などのシステムがあげられる．BotHunter

はMWに感染したPCの通信パターンを認識し，ボットの感染に用いられるトラフィックとMWに感染したPCの特定を試みる．BotSnifferはMWに感染したPCとC&C [3)]サーバのトラフィックの特徴を利用して疑わしいIRC通信の検出をする．これらの既存システムは不正ホストを列挙することを目的としているのに対して，本研究は複数ホストによる連携感染パターンを検出するところに新規性がある．

第2章の構成は次に示す通りである．まず，2.2節でCCC DATAset の統計量と概要を示す．2.3節では，連携感染，ポートスキャン，MWのダウンロードなどに関する特徴を報告する．これらの特徴に基づいて，2.4節では連携感染を検出するためのアルゴリズムを2種類提案し，その精度を評価する．2.5節でボットネットの連携感染を判定する発見的手法を結論づける．

4.2 解析データ

4.2.1 攻撃通信データ内のMWとハッシュ値

研究用データセットCCC DATA set 2009の攻撃通信データは，94台のハニーポットで観測されたボットネットとの通信をtcpdumpでパケットキャプチャーしたlibpcap形式のファイルである．文献[23]によると，ハニーポットは1台のホストOS上で動作するWindows 2000とXPの2台のゲストOSにより構成されている．それぞれインターネット接続されており，パケットキャプチャーはホストOS上で行われている．

ハニーポットは感染の有無に関わらず定期的にリセットされて運用されている．この期間を次のように定める．

定義 1 *1*台のハニーポットが起動して，*(*スケジュールに従って*)*リブートされるまでの観測期間を**スロット**という．

攻撃通信データ2日分はスロットについて145個に分割される[4)]．総MW数は200個あり，そのうちユニークハッシュ値は24種類，MWは表4.1に示す13種類であった．ここで，UH数はユニークハッシュ数を，DL数はダウンロード回数を示している．例えば，`PE_VIRUT.AV`と識別されるMWには，異なる8種類のハッシュ値があることを表わしている．プロトコルはMWをDLする際のトランスポート層の通信方式である．

[3)] C&C（Command and Control）サーバは感染したPCとボットネットの指令者を仲介する中継サーバである．これは指令者を見つかりにくくするためである．

[4)] CCC DATASet 攻撃通信データは，全スロットを単一のファイルに連結しているので，Windows XPが再起動する時にNTPサーバにアクセスするNTPパケットを利用して，スロット毎の通信データに分割して用いる．

表 4.1: 2 日間で観測された全 MW のリスト

MW 名	ラベル	UH 数	DL 数	スキャン数	プロトコル
PE_VIRUT.AV	PE1	8	91	18	TCP
PE_BOBAX.AK	PE2	1	4	4	TCP
PE_VIRUT.AT	PE3	1	1		TCP
BKDR_POEBOT.GN	BK1	1	30		TCP
BKDR_MYBOT.AH	BK2	1	1	6	UDP
BKDR_RBOT.ASA	BK3	4	5		UDP
TROJ_AGENT.ARWZ	TR1	1	6		TCP
TROJ_BUZUS.AGB	TR2	1	24		TCP
WORM_ALLAPLE.IK	WO1	1	1		TCP
WORM_POEBOT.AX	WO2	1	1		TCP
WORM_SWTYMLAI.CD	WO3	1	27		TCP
WORM_AUTORUN.CZU	WO4	1	3		TCP
WORM_IRCBOT.CHZ	WO5	1	1		TCP
UNKNOWN	UK	1	5		TCP

MW が引き起こす攻撃パターンの頻度を表 4.2 に示す．WORM_SWTYMLAI.CD のように，感染のたびに異なる攻撃をするものがあり，MW 名と攻撃の関係は一意ではない．しかし，後述する連携感染を考慮すれば，攻撃を一意に特定出来る．

表 4.2: 単一の MW と攻撃パターンの関係

MW	スキャン (s4)	スキャン (r2)	DoS	SMTP	計
PE_VIRUT.AV	18	1	0	0	91
PE_BOBAX.AK	4	0	3	3	4
BKDR_POEBOT.GN	6	0	0	0	30
WORM_SWTYMLAI.CD	24	1	3	3	27
TROJ_BUZUS.AGB	24	1	0	0	24

4.2.2 特徴量抽出

感染判定のために用いるスロットの特徴量を表 4.3 に示す．特徴量には，ハニーポットの入出力パケット数 P_I，P_O，パケット中に含まれる文字列に関するもの，ポートスキャンに関するもの，ダウンロードした MW に関するものの 4 種類がある．文字列検索には，Network Grep[24] を用いる．ポートスキャンのタイプの s_4 は，スキャンあて先アドレスの第 4 オク

テットが 1 づつ増加する形式である．r_3 はランダムに第 3 オクテットまでを変化させる．入出力パケット数はハニーポットが送受信したパケット数である．ポートスキャンタイプの判定はハニーポットのパケットのあて先を全て調査し，IP アドレスの変化によって明確に判定した．MW 名はその時点での最新パターンファイルを適用したウイルススキャナ (トレンドマイクロ社製) により判定されている [23]．判定できないものは UNKNOWN と表記される．MW の感染の有無は CCC DATAset2009 の攻撃元データとの参照により判定した．

表 4.3: 識別に用いる特徴量一覧

	特徴量	意味
統計量	$slot$	スロット ID$(0, \ldots, 145)$
	P_I, P_O	総入力 (出力) パケット数 [pkt]
文字列の出現の有無	MZ	“ MZ ”
	PE	“ PE ”
	DOS	“ `!This program cannot be run in DOS mode.` ”
	win	“ `!Windows Program` ”
	N, J	“ NICK ”かつ“ JOIN ”
	$ip1$	“ `#las6 * ipscan s.s.s.s dcom2 -s` ”
	$ip2$	“ `#last * ipscan s.s.s.s dcom2 -s` ”
スキャン	ST	ポートスキャンの種類 (s_2，s_3，s_4，r_3)
	DL	感染の有無
	MW	マルウェア名

4.3 解析結果

4.3.1 概要

表 4.3 の特徴量について解析した結果の一部を表 4.4，4.5 に示す．ここで，全スロットの総数を total，平均を ave の行に示す.「感染パターン」の列は，次節で詳細に述べる．

全 145 のスロットの中で MW をダウンロードしているスロットは 58 件であった．これらを詳細に解析した結果，表 4.6 に示されるいくつかのルールを発見した．ルールは連携感染に関する Rule 1〜5，ポートスキャンに関する Rule 6〜8，MW に関する Rule 9〜10 がある．これらのルールの発見過程に用いた関連データを表 4.6 の第 3 列に示し，以後詳細に述べる．

表 4.4: スロットと各種特徴量 (一部)1

スロット	P_I	P_O	MZ	PE	DOS	N, J	$ip1$, $ip2$	$ST(s_4)$	感染
0	276	17774	9	13	3	1		1	1
1	61	352	0	4	0				0
2	7488	178491	10	16	3	1	$ip2 \times 1$	1	1
3	350	240148	12	10	4	1	$ip2 \times 1$	1	1
4	2	55	0	0	0				0
5	5	59	0	0	0				0
14	354	135725	9	10	3	1	$ip1 \times 3$	1	1
55	822	179581	21	16	7	1	$ip1 \times 2$	1	1
46	379	791	0	0	0				1
83	571	74286	15	15	5	1		1	1
139	450	96211	13	18	3	1	$ip2 \times 1$	1	1
140	691	101877	21	24	5	1	$ip2 \times 1$	1	1
total	44452	3038276	691	966	219	60	33	28	58
ave	306.57	20953.63	4.77	6.66	1.51	0.41	0.23	0.19	0.4

表 4.5: スロットと各種特徴量 (一部)2

スロット	MW	感染パターン
0	$PE1$, $TR2$, $WO3$	1
1		
2	$WO1$, $PE1$, $TR2$, $WO3$	1
3	$PE1$, $TR2$, $WO3$, $PE1$	1
4		
5		
14	$BK1$, $TR2$, $WO3$	2
55	$BK1$, $WO3$, $TR2$, $BK1 \times 4$	2
46	$BK2$	
83	$PE1 \times 2$, $TR2$, $WO3$	1
139	$PE2$, $WO4$, $WO3$	3
140	$PE2$, $WO4$, $WO3$	3
total	200	
ave	1.38	

表 4.6: 連携感染の特徴を表わすルール一覧

NO.	ルール	関連 (データ)
Rule 1	PE_VIRUT.AV をダウンロードしたならば WORM_SWTYMLAI.CD と TROJ_BUZUS.AGB を同時刻にダウンロードを開始する.	図 4.2
Rule 2	WORM_SWTYMLAI.CD と TROJ_BUZUS.AGB のダウンロード直前に JOIN がある.	図 4.2
Rule 3	WORM_SWTYMLAI.CD と TROJ_BUZUS.AGB の DL サーバは常に一定.	表 4.7,4.9
Rule 4	PE_VIRUT.AV は 5 桁のポート番号使う.	表 4.7
Rule 5	WORM_SWTYMLAI.CD と TROJ_BUZUS.AGB はポート番号 80 番を使う	表 4.7
Rule 6	連携感染ならば，ポートスキャン先は PE_VIRUT.AV の DL サーバの第 1，2 オクテットと同じ.	表 4.10
Rule 7	IRC で“ JOIN ”を受信したならば約 5 秒後にポートスキャンを開始する.	図 4.4
Rule 8	連携感染したならば，1 秒間に 256 パケットのポートスキャンを連続して行う.	図 4.3
Rule 9	文字列“ MZ ”かつ“ PE ”を含むならば TCP による感染である.	表 4.4，4.5
Rule 10	UDP で win という文字列があれば，TFTP のダウンロードである.	なし

4.3.2 連携感染に関する特徴

定義 2 (連携感染) 単一のボットネットにより制御されている複数の *DL* サーバが連携して *1* つ以上の *MW* を単一ホストに多重に感染させる不正行為を**連携感染**と呼ぶ.

単一のハニーポットが複数の MW に感染しても，それが同一のボットネットによるものかどうかは厳密には分からない．しかし，連携感染は，通常スクリプトなどで機械的に引き起こされるので，感染間隔，MW の種類やポート番号に特定のパターンが生じやすい．利用される DL サーバ，ソース IP アドレスにも一定のパターンが生じる．そこで，多くのスロットを解析し，共通のパターンを抽出していく.

連携感染の基本パターンを図 4.2 のタイムチャートに示す．脆弱性のあるホスト (ハニーポット) は感染すると S_1，S_2，S_3, の 3 種類の中継/DL サーバから，PE を時刻 t_0 で，TROJ, WORM の異なる MW を t_2 のタイミングでダウンロードする (Rule 1)．また，TROJ と WORM をダウンロードする直前に C&C サーバー S_0 との間で IRC のセッションを確立し，NICK[5] と JOIN の命令を受ける (Rule 2)．時刻 t_4 で，指定されたあて先ネットワークにポートスキャンを試みる．ここで，最初の MW から次の MW をダウンロードする間隔と，IRC の JOIN からポートスキャンまでの間隔を各々,

$$\Delta T_1 = t_2 - t_1$$

$$\Delta T_2 = t_4 - t_2$$

と定義する.

[5] NICK は C&C サーバと最初に通信を行う際のコマンドである.

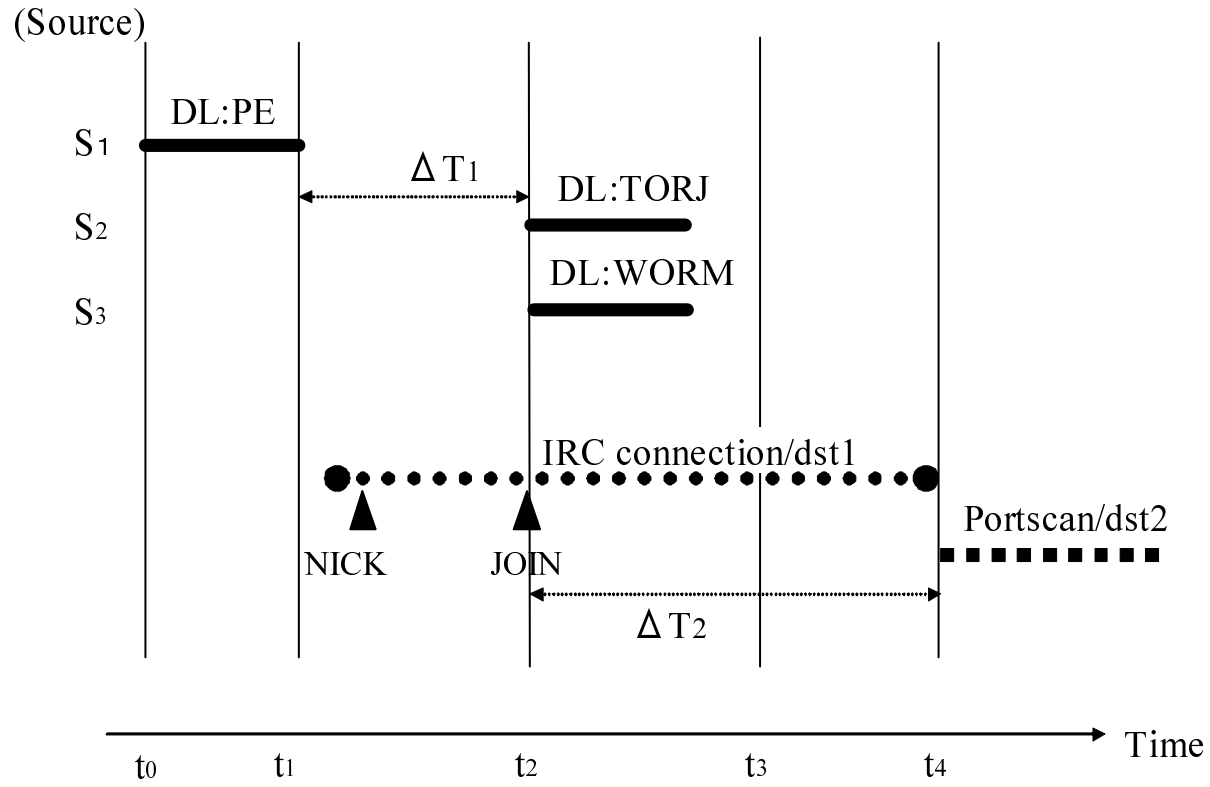


図 4.2: 連携感染の通信路のタイムチャート

連携感染する具体例を表 4.7 に示す．PE_VIRUT.AV をダウンロードさせる DL サーバの IP アドレスはまちまちだが，WORM_SWTYMLAI.CD と TROJ_BUZUS.AGB の DL サーバの IP アドレスは全てのスロットで同じであった (Rule 3)．どのスロットも，PE_VIRUT.AV は 5 桁のポート番号を用いている (Rule 4)．TROJ_BUZUS.AGB と WORM_SWTYMLAI.CD は 80 番である (Rule 5)．

MW をダウンロードしている 58 のスロットは表 4.8 に示される 3 つの連携パターンに分類される．MW 名は表 4.1 を元にしている．表 4.8 より，MW 感染が確認された 58 スロットの内 26 スロットが複数の DL サーバに渡る連携感染であることが分かる．中でも，連携パターン 1 は，頻度が高く，ポートスキャンなどの攻撃も伴うので重要である．ダウンロードする MW の種類やポート番号には共通の特徴が見られるが，時差 ΔT_1 の分散は大きく，感染の度に変化している．

MW と DL サーバの関係は 1 対 1 ではない．表 4.9 に示されるように，連携感染の最初の PE_VIRUT.AV は，感染の度に異なる (10 台の) サーバからダウンロードされているのに対し，後半の TROJ，WORM のダウンロードは一台のサーバに集中していた．

表 4.7: 連携感染パターン 1 の通信路

スロット	時間	srcIP	dstPort	MW 名
0	0:02:11	124.86.A1.B1	47556	PE_VIRUT.AV
0	0:03:48	67.215.C1.D1	80	TROJ_BUZUS.AGB
0	0:03:48	72.10.E1.F1	80	WORM_SWTYMLAI.CD
2	0:36:46	124.86.A2.B2	33258	PE_VIRUT.AV
2	0:36:52	72.10.E1.F1	80	WORM_SWTYMLAI.CD
2	0:36:52	67.215.C1.D1	80	TROJ_BUZUS.AGB
3	0:46:56	124.86.A2.B2	33258	PE_VIRUT.AV
3	0:48:52	67.215.C1.D1	80	TROJ_BUZUS.AGB
3	0:48:52	72.10.E1.F1	80	WORM_SWTYMLAI.CD
16	5:17:25	114.145.A3.B3	15224	PE_VIRUT.AV
16	5:18:37	67.215.C1.D1	80	TROJ_BUZUS.AGB
16	5:18:38	72.10.E1.F1	80	WORM_SWTYMLAI.CD

表 4.8: 連携感染パターンとその統計量

	パターン	スロット ID	スロット回数
連携 1	PE1 → TR2，WO3	0，2，3，16，29，30，50，60，63，69，70，71，83，94，100，130，132	17
連携 2	BK1 → TR2，WO3	14，55，56，124，125，126	6
連携 3	PE2 → WO4，WO3	139，140，141	3
4	WO1	2	1

4.3.3 ポートスキャンに関する特徴

表 4.10 は，連携感染してポートスキャンを引き起こしたスロットにおける，DL サーバ，ハニーポット (感染 PC)，ポートスキャンあて先の IP アドレスを示している．3 つの IP アドレスの第 1，2 オクテットは，全て等しく (Rule 6)，ハニーポットとスキャンのあて先 IP アドレスの第 3，4 オクテットは等しい．なお，このあて先 IP アドレスは，1 ずつインクリメントされる．

図 4.3 は，連携感染 1 における入出力パケットの通信速度の変化を表している．上がハニーポットへの入力，下が出力を表している．スロット内の相対時刻で 600[s] の時に連携感染が生じ，その直後にポートスキャンを外部に対して行っている．この送信は毎秒 256 パケットの一定の割合で行われる (Rule 8)．

ポートスキャンには，第 4 オクテットを 1 づつ増加させる s_4 と第 3 オクテットをランダムに変える r_3 の 2 種類が観測された．コマンド“ JOIN ”が送られてからポートスキャンが

表 4.9: MW ごとのユニーク DL サーバ

MW 名	ユニーク DL サーバ数
PE_VIRUT.AV	10
TROJ_BUZUS.AGB	1
WORM_SWTYMILAI.CD	1

表 4.10: DL サーバ，ハニーポット，スキャンの IP アドレス

slot	DL サーバ	ハニーポット	スキャンあて先
0	124.86.$C1$.$D1$	124.86.$E1$.$F1$	124.86.$E1$.$F1+1$
2	124.86.$C2$.$D2$	124.86.$E2$.$F2$	124.86.$E2$.$F2+1$
3	124.86.$C2$.$D2$	124.86.$E2$.$F2$	124.86.$E2$.$F2+1$
16	114.145.$C3$.$D3$	114.145.$E3$.$F3$	114.145.$E3$.$F3+1$
29	114.164.$C4$.$D4$	114.164.$E4$.$F4$	114.164.$E4$.$F4+1$
例	$A.B.C.D$	$A.B.E.F$	$A.B.E.F+1$

起きるまでの時間差 ΔT_2 の分布を図 4.4 に示す．X 軸は“ JOIN ”，Y 軸はポートスキャンの通信開始時刻を表している (ただし，時間と分の値を略して，秒だけで表したグラフに重ねてプロットしている)．直線と観測時刻との間が時差 ΔT_2 である．観測された 26 回の s_4 のポートスキャン全てで，JOIN に対しスキャン開始時間が正確に 5 秒遅延している事が分かる (Rule 7).

4.3.4 MW のダウンロードに関する特徴

連携感染を行う際には，特徴的なメッセージが送信されている．表 4.4 に示される様に，“ MZ ”と“ PE ”の両方が送信される時は感染をしている (Rule 9).

4.3.5 UDP の感染に関する特徴

UDP を使った tftp での感染は 6 スロットあった．そのうち MW 名は 5 スロットが BKDR_RBOT.ASA で，残り 1 スロットは BKDR_MYBOT.AH であった (Rule 10).

4.4 感染判別の手法

4.4.1 一般感染の検出アルゴリズム

表 4.6 のルールに基づき，図 4.5 に示す感染判定の決定木を提案する．ここでは連携感染と通常の感染の区別をせず，与えられたスロット内で (任意の) 感染があることを自動判別す

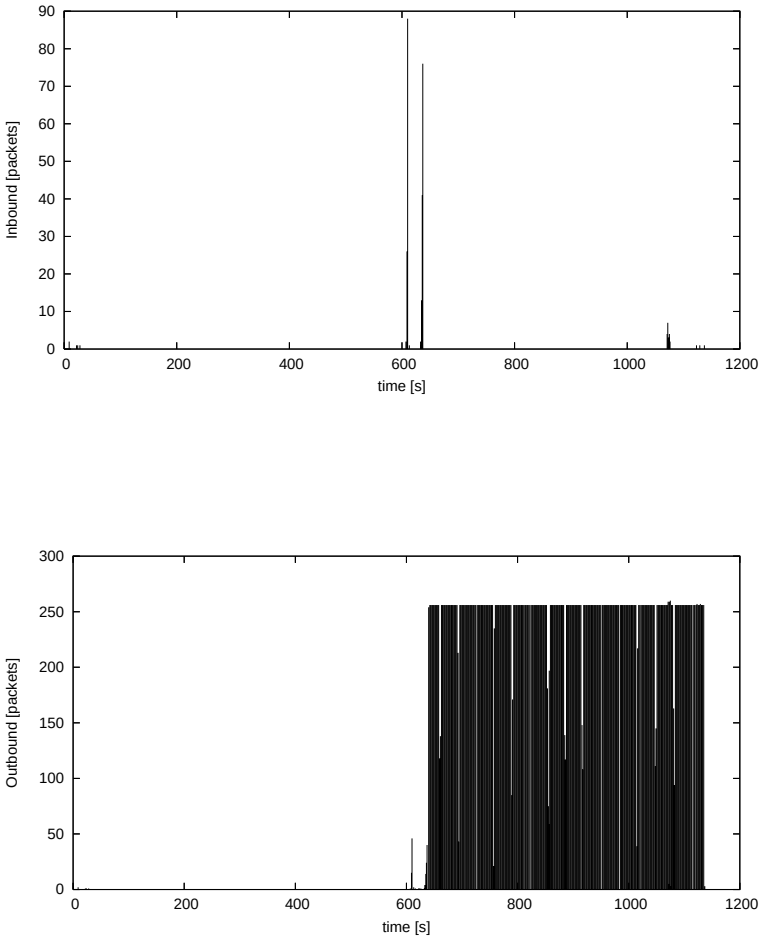


図 4.3: 単位時間当たりの入出力パケット数の変化

る．決定木のノードは，表 4.3 で定義した識別の特徴量を示し，木の枝に示される式は識別の閾値を与えている．例えば，木のルートは総入力パケット数 P_I が 85 パケット以上かどうかで分岐することを表わしている．“ DOS ”というノードは， exe ファイルがダウンロードされたときに文字列“ !This program cannot be run in DOS mode. ”が出現するか (Y) 否か (N) で識別する．感染判定の決定木は 2009 年の攻撃通信データのみを使い作成した．このアルゴリズムの精度を表 4.11 に示す．ルールを発見するための学習にはあるハニーポット (Windows XP) の攻撃通信データ，評価には別のハニーポット (Windows 2000) のデータを用いた．両データセット共，誤検出は生じなかった．

代表的な決定木アルゴリズム C4.5[25][26] を適用して，抽出した感染を判別する決定木を図 4.6 に示す．図 4.5 と同様，ノードは識別の特徴量を表す．葉の「1 (49/0)」は，その葉へ分類されるデータの数が 49 件あり，1(感染) という識別ラベルに対して誤識別が 0 であるこ

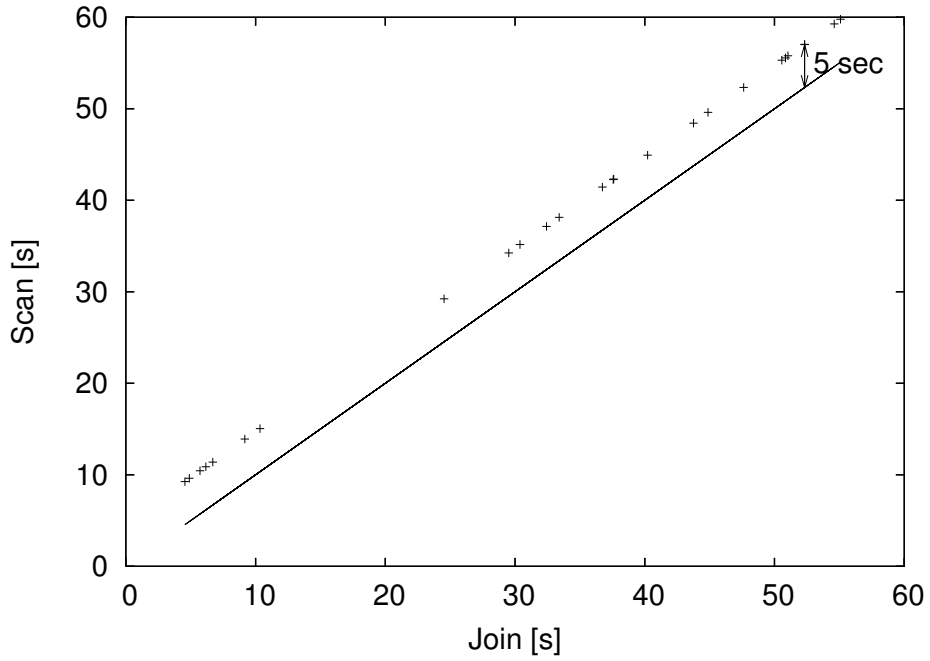


図 4.4: JOIN の送信時刻と Scan の開始時刻の差 ΔT_2 の分布

表 4.11: 感染 (連携感染を含む) を判定する決定木の精度

真値 \ 判定結果		感染あり	感染なし	total slot
学習	感染あり	58	0	58
データ	感染なし	0	87	87
評価	感染あり	6	0	6
データ	感染なし	0	14	14

とを表わしている．図 4.5 と比較して，ノードの数が 4 つと少なく，最適化が試みられているが，`Out_pkt < 338` に分類されている 7 スロット中，感染と誤判定 (False Positive) されるスロットが 1 つ生じている．

4.4.2 連携感染パターンの発見的手法

単一の感染は，4.1 節の決定木で判別が容易だが，連携感染は例外的振舞いが多く，確定的なアルゴリズムでの検出が困難である．そこで，3 章で述べた連携感染に関する規則に基づき，各ルールを並列に評価した合計スコアによる発見的手法を提案する．

i 番目のスロットにおける Rule j の成立を $x_{ij} = 1$ と定める．スロット i のスコアは，$S_i = \sum_j^9 x_{ij}$ と定義する．このスコアが閾値以上かどうかで判定を行う．学習データにはあるハニーポットの 2 日間の全スロットデータを使用した．表 4.12 は，学習データにおける

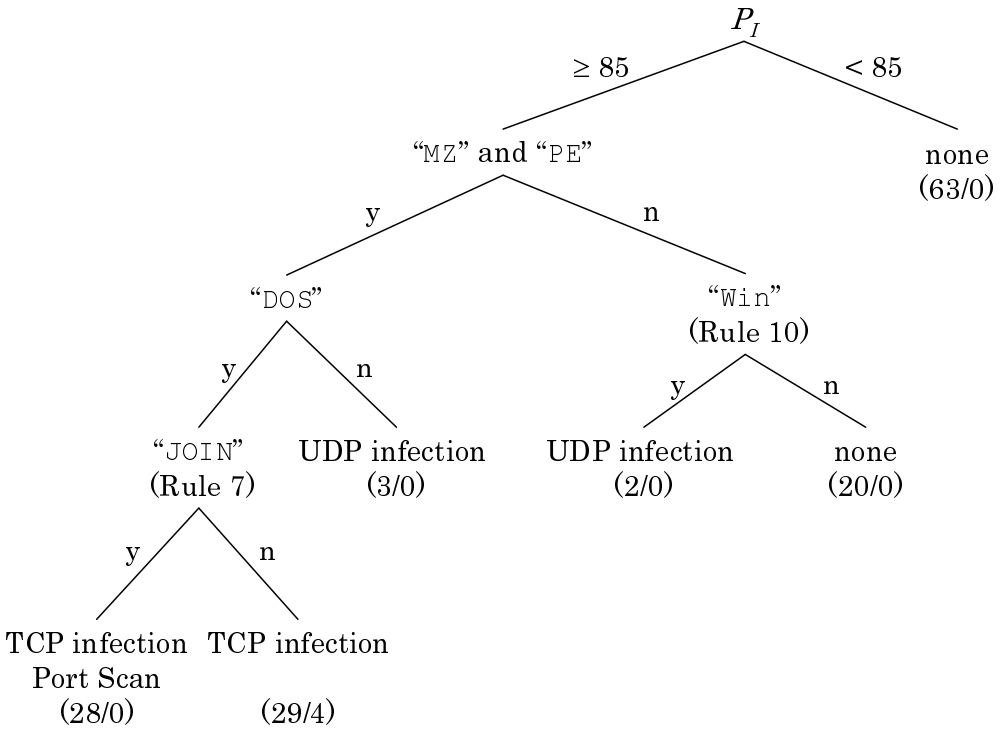


図 4.5: 感染 (連携感染を含む) を判定する決定木

各ルールの成立とスコアの一部である．この学習データでは，連携感染しているスロットの最小スコアが 3 であった．そこで，閾値を 3 と定める．このときのスコアの分布を図 4.7 に示す．表 4.13 の学習データにおいて，連携感染の誤検知が 2 スロット生じている．この内のひとつはスロット 66 であり，表 4.8 で分類した 3 種類の連携感染のどのパターンでもなく，4 番目の新たな連携感染パターン (PE2 → WO4，WO3) に分類される[6]．もうひとつは CCC DATAset 2009 攻撃元データの誤り[7]から混入したものであった．この提案による精度を表 4.13 に示す．

145 スロットの中で感染している 58 パターンの出現頻度とその精度 (ルールの成立割合) を表 4.14 に示す．例えば，Rule 1 は 145 スロット中 17 スロットが該当しており (頻度)，その精度は PE_VIRUT.AV をダウンロードした全 38 スロット中，WORM と TROJ をダウンロードしたものが 17 スロットあることを示している．145 スロット中 58 の感染スロットの中で連携感染を行っているスロットは 26 あり，約半分が連携感染である．

[6] これは，手作業で表 4.8 を作成した際に列挙から漏れてしまっていたパターンであり，本来ならば，表 4.8 に加えるべきものである．従って，提案方式の有効性を失わせるものではなく，むしろ，発見的手法が学習データの誤り検出に有効であったことを示している．

[7] 攻撃通信データには存在するが攻撃元データには記録のないスロットであった．

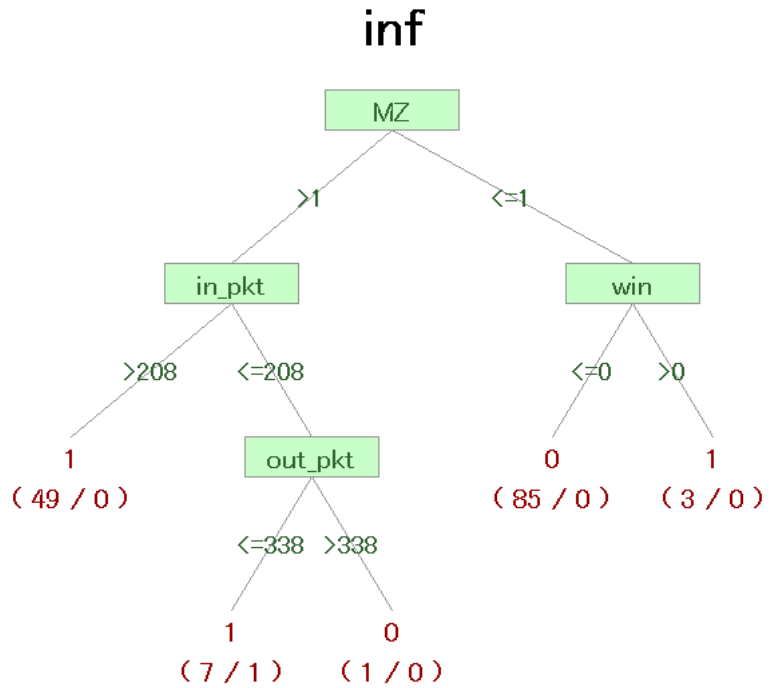


図 4.6: C4.5 による感染判定の決定木

表 4.12: 発見的手法のスコアと連携感染の有無の関係 (一部)

スロット	Rule									スコア	
i	1	2	3	4	5	6	7	8	9	S_i	連携感染
0	1	1	1	1	1	1	1	1	1	9	1
1	0	0	0	0	0	0	0	0	0	0	0
2	1	1	1	1	1	1	1	1	1	9	1
3	1	1	1	1	1	1	1	1	1	9	1
14	0	1	1	0	1	0	1	1	1	6	1
15	0	0	0	0	0	0	0	0	1	1	0
139	0	0	0	0	0	0	1	1	1	3	1
total	17	24	24	17	24	17	28	28	56	170	28

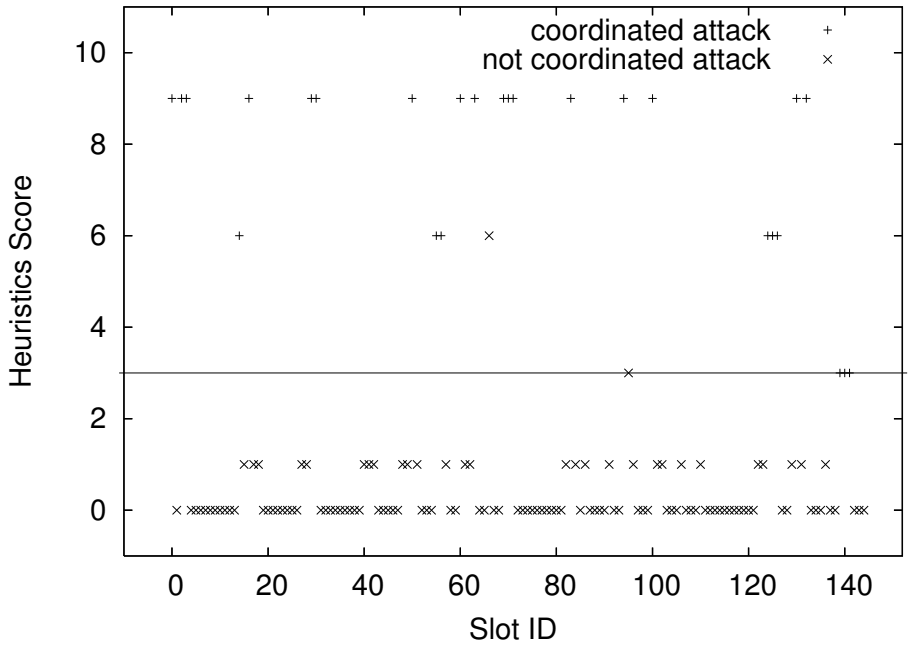


図 4.7: 発見的手法のスコアの分布

表 4.13: 連携感染を判定する発見的手法の精度

真値 \ 判定結果		連携感染と判定	連携感染でないと判定	FP	FN
学習	連携感染	26	0		
データ	連携感染でない	2	119	2/28	0/117
評価	連携感染	2	0		
データ	連携感染でない	1	7	1/3	0/7

表 4.14: Rule の出現頻度と成立割合

ルール	出現頻度 [スロット]	成立割合 [スロット](%)
Rule 1	17/145	17/38 (45%)
Rule 2	17/145	17/27 (89%)
Rule 3	22/145	22/27 (81%)
Rule 4	17/145	17/17 (100%)
Rule 5	17/145	17/17 (100%)
Rule 6	17/145	17/17 (100%)
Rule 7	28/145	28/28 (100%)
Rule 8	28/145	26/28 (93%)
Rule 9	55/145	55/63 (87%)
Rule 10	6/145	6/6 (100%)

4.5 付録

4.5.1 MW 名判別の発見的手法

キャプチャデータから，tcpflow[27] などのツールを用いて MW をダウンロードすれば，アンチウイルスソフトにより検出が可能である．ただし，全てのパケットから抽出出来るのではなく，表 4.15 に示される割合で成立する．MW の特定は HTTP，UDP ともファイル復元ができ，達成することができた．

表 4.15: MW 名の判定

ルール	ファイル復元	MW 名判定
TCP	192/194 スロット	192/192 スロット
UDP	6/6 スロット	6/6 スロット
	ファイル復元数 /攻撃元データ	MW 判定数 /復元ファイル数

第5章
Drive-by-download攻撃の分類

5.1 概要

近年，インターネット上の脅威の一つとして，Web ブラウザの脆弱性を利用して感染させる Drive-by-Download 攻撃 [30] によるマルウェアの感染被害が後を絶たない．中でも 2009 年 4 月に出現した Gumblar は，亜種の発生が非常に早く [31]，従来のパターンマッチングによるマルウェアの検知では対応が遅れてしまう．例えば，IP フィルタリングは，既知の IP からの攻撃を防ぐことは可能であるが，存続期間が短い Drive-by-Download などの Web を利用した攻撃には有効でない．次々と新たなホスト名を用いた攻撃サイトが作られる [32]．Drive-by-download 攻撃によるマルウェア感染のイメージを図 5.1 に示す．

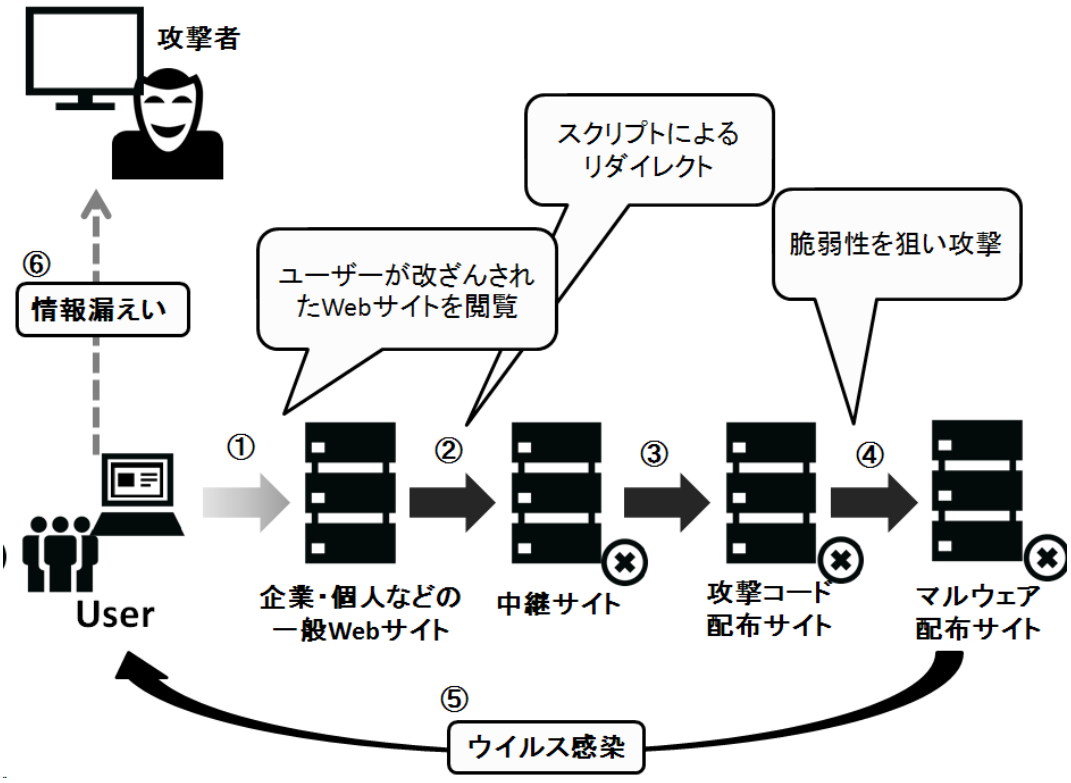


図 5.1: Drive-by-download 攻撃によるマルウェア感染のイメージ

そこで本研究では，マルウェア本体の挙動ではなく，マルウェアがダウンロードされるまでの通信と挙動の特徴に着目し，未知の攻撃に対しても有効な分類方法を提案する．サーバのアドレスが変わっても，引き起こされる一連の攻撃のパス列は変動しないことに着目し，攻撃検知や識別に利用することが出来るのではないかと考える．そこで，私たちは，Web 感染型マルウェアの通信を観測した MWS2010 研究用データセット D3M2010 攻撃通信データ [33] を用いて，本仮説を検証した．通常の通信から逸脱している通信の遷移を特定することにより，Drive-by-Download 攻撃固有のパスに注目し，次の 3 つの提案手法により攻撃の分類を試みる．

特徴量 A**.** Drive-by-Download 攻撃に用いられる通信の発信元 IP などの特徴.

特徴量 B**.** 一連の攻撃に用いられる URL のパス列.

特徴量 C**.** 攻撃に用いられる脆弱性の種類と数.

5.2 D3M 2010 攻撃通信データの解析

D3M 2010 攻撃通信データは，Web クライアントハニーポット 10 台で特定の URL を巡回した時の通信をキャプチャした pcap 形式のデータである．特定の URL は，公開ブラックリストの提供サイト [13] に登録されている URL の中から，Drive-by-Download 攻撃を検知した URL である (以下，この URL を巡回対象 URL と呼ぶ).

5.2.1 攻撃通信データの分割

基本となる攻撃の単位を次の様に定義する.

定義 3 (スロット) *1* つの巡回対象 *URL* へのアクセスにより生じる一連の通信，すなわち，*HTTP* 通信が行われ，リダイレクト，外部サイトのスクリプトや画像の読み込みなどの派生先 *URL* へのアクセスを含む複数の通信路を**スロット**と呼ぶ.（スロットは“ 観測日-識別番号 ”の形式の *ID* で識別する）.

スロットへの分割は次のように行う；(1)Referer ヘッダ (参照元 URL) を含まない GET リクエストを抽出，(2) 巡回 URL リストの URL を照会し，巡回対象 URL へのアクセスを行っている GET リクエストを抽出，(3)GET リクエストが複数出力された場合は，目視で調査し巡回対象 URL へのアクセスを特定する.

各観測日の巡回対象 URL へのアクセスデータは表 5.1 の通りである．1 件も候補が存在しない URL に対しては，DNS による名前未解決，サーバーエラーなどの原因が考えられる．失敗の原因を表 5.2 に示す．巡回対象 URL のリストに幾つか重複した URL が存在していた

が，抽出した GET リクエストのパケットログを調査した結果，URL が重複していてもアクセスは 1 度きりであると判明したため，重複 URL はユニークとした．

以上の処理により，D3M 攻撃通信データ 3 日間を 518 スロット分割した．

表 5.1: 各観測日の巡回対象 URL へのアクセスのデータ

観測日	2010/3/8	3/9	3/11
巡回対象 URL	205	180	172
該当しなかった URL	25	6	5
一回のみヒットした URL	163	158	158
複数回ヒットした URL	17	16	9
出力スロット数	180	174	164

表 5.2: 巡回対象 URL へのアクセスが確認できなかった URL に対しての調査

観測日	2010/3/8	3/9	3/11
該当しなかった URL	25	6	5
DNS 応答なし	4	0	0
DNS 応答あり	2	2	0
DNS 応答あり (3-way 未確立)	19	4	5

5.3 提案方式

5.3.1 既知攻撃の特徴

D3M2010 の攻撃通信データには，Gamblar の亜種であり，8080 ポートを使うことで有名な ru:8080[35] が 13 回，3129-3126 ポートを使うインジェクション攻撃 [36] が 10 回観測された．ru:8080 で引き起こされる一連の URL の関係を図 5.2 に示す．

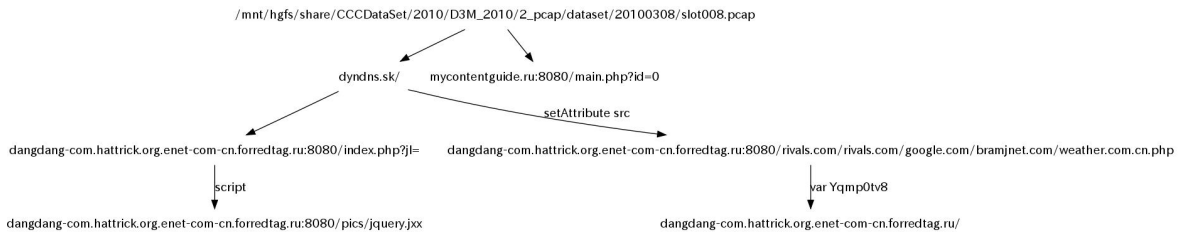


図 5.2: ru:8080 攻撃により誘起される URL の関係 (スロット ID，308-8)

これらの攻撃にはそれぞれの特徴的なポートを使う事に加え，表 5.3 に示す一連の通信で用いられるディレクトリ構造 (以下，パス) の特徴が見られた．このように初めの誘導サイトと中継サイトのホスト名はまちまちであったが，そこで用いられるシーケンスは全て同じであった．これは，ru:8080 に限らず他の攻撃にも観測される特徴であり，例えば表 5.4 に示される 9 スロットは最初の 4 つを除いてほぼ一定のパスシーケンスを持つ．これらは，後述する攻撃パターン $B1$ に分類されるパスシーケンスである．

表 5.3: ru:8080，3126-3129 攻撃のパスのシーケンス

No.	ru:8080 Gumblar	3126-3129 インジェクション攻撃
1	/index.htm またはなし	.html またはなし
2	.com.php または.com.cn.php	/in.php
3	/index.php?jl=	/js
4	/pics/jquery.jxx	/download/index.php
5	/mycontentguide.ru:8080 /main.php?id=0	/download/jabber.php
6	/pics/ChangeLog.pdf	/download/banner.php?spl=mdac
7	/pics/java.html	
8	/pics/JavaJopa.jar	
9	/pics/JavaJopa.jar	
10	/pics/JavaJopa.jar	
11	/pics/JavaJopa.jar	
12	/welcome.php?id=9&hey240	

5.3.2 特徴量 A

表 5.5 に，518 スロットから抽出した代表的な攻撃パターンの一覧 $A1, \ldots, A12$ を示す．各パターンは，URL のパスの特徴的な文字列によって識別されている．ここで DNS 数，IP 数は，各攻撃パターンの特徴を満たすのものの数である．表 5.5 の 1〜7 の攻撃パターンは全て，1 つのパスに対して，複数の IP が存在する．攻撃パターン 13 は 1 つの IP からの攻撃であるが，同じパスのシーケンスにもかかわらず，マルウェアの配布を成功した場合とそうでない場合が存在する．このことから，ある決まった条件の時のみ，マルウェアをダウンロードするように考えられる．

最も頻度が高かった攻撃は，pdf.php を含む攻撃パターン $A3$ である．

表 5.4: パスのゆらぎ (攻撃パターン $B-1$)

308-2	308-45	308-149	309-4	309-82	309-155	311-53	311-59	311-74
15	15	15	15	15	15	15	15	15
15	15	15	15	15	15	1	180	1
57	57	57	57	57	57	180	169	
1	1	1	1	1	1	169	170	
180	180	180	180	180	180	170	171	
169	169	169	169	169	169	171	176	
170	170	170	170	170	170	176	173	
171	171	171	171	171	171	173	174	
176	176	176	176	176	176	174	175	
173	173	173	173	173	173	175	181	
174	174	174	174	174	174	181	50	
175	175	175	175	175	175	50	183	
181	181	181	181	181	181	183	184	
50	50	50	50	50	50	184	179	
183	183	183	183	183	183	178	175	
184	184	184	184	184	184	172	168	
180	177	180	180	180	180	172	177	
174	172	173	177	177	174	182	178	
182	175	182	178	179	178	183	168	
178	180	176	173	178	175	173	173	
172	174	177	179	173	179	176	174	
185	182	172	175	176	173	168	176	
177	170	175	174	175	177	171	168	
172	179	178	176	170	176	181	172	
175	172	172	167	167	170	183	170	
179	176	185	167	167	171	179	170	
181	178	167	172	172	183	174	172	
	181	179	172	172	167	177		
	185	181	172	167	172	173		
	168	174	167	172	182	174		
	168	167	181	181	168	170		
	167	172	167	167	172	172		
	167	168	182	182	168	172		
	172	167	185	185	167	170		
	168	167	168	168	172	177		
	167	168	168	168	185	170		
	170	168	168	168	181	171		
	175	170	176	173	168	185		
	173	175	170	174	174	168		
	176	176	174	170	176	170		
		174	175	175	173	169		
					175	183		
						172		
						175		

5.3.3 特徴量 B

3.1 節の解析により，同一のマルウェアによる攻撃は，脆弱性コードを埋め込む Web サーバは変わっていても，そこから遷移するパスは共通であることが多い事が分かった．そこで，このパスのシーケンスを，攻撃を識別するための特徴量として用いることを提案する．URL から DNS 名を取り除いたものをフルパスと呼び，一意な識別番号で参照する．パスシーケンスの中の特徴的な部分パスを用いて，表 5.6 の 21 パターンの特徴量 $B1, \ldots, B21$ を定めた．これをフルパスインデックスと呼ぶ．

5.3.4 特徴量 C(脆弱性)

同一のマルウェアならば，用いる脆弱性コード CVE の組みにも共通の特徴が見られるはずである．そこで，Blake Hartstein によって開発されたマルウェアアンパックツール jsunpack-n[34] を用いて，各攻撃で用いられる脆弱性を抽出し，その組を特徴量とすることを考える．

jsunpack-n とは、通信に含まれる脆弱性を突いた攻撃の検出を行うツールで、通信の要約、受信データの解読、難読化の解除を行うことができる．D3M では，CVE2005-2127 か

表 5.5: 攻撃パターン特徴量分類 A

攻撃パターン	パスに含まれる特徴的な文字列/発信元 IP	スロット数
$A1$	“index.php?spl=2”	27
$A2$	“cache/PDF.php?st=Internet\%20Explorer\%206.0”	34
$A3$	“pdf.php[pdf]”	55
$A4$	“load.php?a=a\&e=6$”	15
$A5$	“/load.php?spl=mdac$”	8
$A6$	“/load.php?id=0$”	7
$A7$	“/load.php”	24
$A8$	“85.17.90.206	17
$A9$	“91.213.174.22	8
$A10$	“213.163.89.54	21
$A11$	“$/newload.php?ids=MDAC$”	7
$A12$	115.100.250.73 ”	8
	計	231

ら 2010-0249 までの 14 種類の脆弱性が用いられた．攻撃パターン 3 の脆弱性を表 5.7 に整理した.（図 5.3 は，誘導される URL 数の分布を示している．平均 7.29 である.）

5.4 攻撃分類の精度

5.4.1 既知の攻撃に対する精度

提案した分類方法の有効性を考える．既知の攻撃 G(ru:8080，インジェクション-3129 攻撃等) に対して，攻撃パターン A，フルパスインデックス B，脆弱性 C の 3 つの方式での識別結果を表 5.8，5.9，5.10 に各々示す．それぞれの表で 0 は各パターン以外のその他に分類した．$A1$，$A5$，$A7$ によって既知攻撃 ru:8080 を識別する時の再現率，適合率は，

$$\begin{aligned} R_{A8080} &= \frac{4}{13} = 0.31, \\ P_{A8080} &= \frac{4}{49} = 0.08 \end{aligned}$$

であり，同様に 3126 に対しては，$R_{A3126} = 2/13 = 0.15$，$P_{A3126} = 2/5 = 0.4$ である以上により，A の総合的な精度をこれらの平均再現率で,

$$R_A = \frac{0.31 + 0.08}{2} = 0.20$$

と定める.

表 5.6: パスによるスロットの攻撃分類 B

	フルパスシーケンス	第 1 フルパス（シーケンスの初めのパス）	スロット数
$B1$	180, 169, 170, 171, 176, 173,	/res/1/1/images/page_progressbar.gif	9
	174, 175, 181, 50, 183, 184		3
$B2$	60, 2	/java	19
$B3$	3, 4, 62	/cache/CSS.css	11
$B4$	199	/zcv.gif	20
$B5$	64, 66、67	/new/da.js	11
$B6$	71, 8	/pca3.crl	12
$B7$	56	/index.php	17
$B8$	53 or 63	/in.cgi?3	11
$B9$	4	/cache/PDF.php?st=Internet\%20Explorer%206.0	10
$B11$	(198)	/x/?html=1&id=992&hash=6339a5f067adeab2eb7cd0e942c81583	6
$B12$	197, 10, 9	/wp.js	6
$B13$	(7), 6	/cry217/xd.php	3
$B14$	(190), 193, 195	/webalizer/050709wareza/crack=17=keygen=serial.html	5
$B15$	58	/intl/ja/images/jawh_prodicons1.png	3
$B16$	(188), 52	/script/in.cgi?2	30
$B17$	72	/pdf.php	3
$B18$	12	/ga.js	7
$B19$	52	/in.cgi?2	3
$B20$	(54)	/in.cgi?4	6
$B21$	(187)	/s/	2

再現率が低い理由は，A を定める際に，既知の攻撃を除外したためである．$G \times B$，$G \times C$ についても同様に $B4$ と $C1$，$C2$ を ru:8080 の識別条件として，再現率を求めると，$R_{B8080} = 2/13 = 0.15$, $R_{B3126} = 8/13 = 0.61$, $R_{C8080} = 4/13 = 0.31$, $R_{C3126} = 10/13 = 0.91$ であった．再現率の平均値は，

$$R_B = 0.38, R_C = 0.54$$

，従って，既知攻撃に対しては，C が最も精度が高い．

5.4.2 分類方式間の相関

特徴量 A と脆弱性特徴量 C との相関をクロス集計で表 5.11 に示す．単一の攻撃パターンが数多くの脆弱性を使っている事が表されている．A と C は独立と言える．

表 5.7: 脆弱性

	脆弱性	観測回数
$C1$	CVE-2005-2127	15
$C2$	CVE-2006-0003	34
$C3$	CVE-2006-3730	10
$C4$	CVE-2006-5820	2
$C5$	CVE-2007-0024	15
$C6$	CVE-2007-5659	161
$C7$	CVE-2008-0015	50
$C8$	CVE-2008-2463	27
$C9$	CVE-2008-2992	102
$C10$	CVE-2009-0927	105
$C11$	CVE-2009-1136	35
$C12$	CVE-2009-1492	40
$C13$	CVE-2009-4324	27
$C14$	CVE-2010-0249	36
計		595

表 5.8: A と G の関係

G ＼ A	1	2	5	7	9	0	計
8080	2	0	1	1	0	9	13
3126-3129	0	1	0	0	1	11	13
その他	25	0	0	20	5	251	301

表 5.9: B と G の関係

G ＼ B	4	10	14	18	0	計
8080	2	0	0	0	11	13
3126-3129	0	1	6	1	5	13
その他	10	10	0	2	291	313

表 5.10: G と C の関係

G ＼ C	1	2	3	4	5	6	7	8	9	10	11	12	13	0	計
8080	2	2	1	0	0	0	0	0	1	0	0	0	0	0	6
3126	0	1	3	1	2	1	0	0	1	1	0	0	0	1	11
その他	34	156	90	9	105	22	15	34	33	35	15	1	2	27	

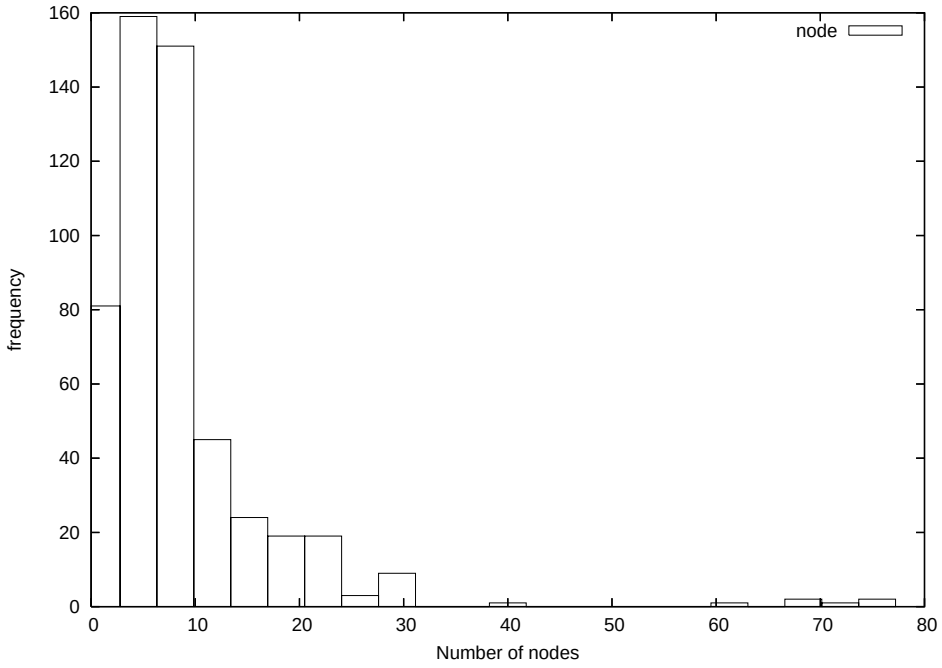


図 5.3: ノード数の分布

表 5.11: A と C の関係

A ＼ C	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	36	63	28	7	24	5	15	11	5	22	2	1	1	4
1	0	24	6	0	24	0	0	0	0	0	0	0	0	1
2	0	19	7	0	7	0	0	8	0	0	0	0	0	8
3	0	41	41	0	41	18	0	13	0	14	13	0	1	13
4	0	1	1	0	1	0	0	2	0	0	0	0	0	2
5	0	2	2	0	2	0	0	0	2	0	0	0	0	0
6	0	7	7	0	0	0	0	0	7	0	0	0	0	0
7	0	0	0	3	0	0	0	0	21	0	0	0	0	0
8	0	2	2	0	8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0

第6章
Drive-by-download攻撃の検知

3.4 の攻撃分割の精度の結果より，脆弱性の特徴量 C による精度が高かったことから，以降，脆弱性を用いた検知手法について考察する．

6.1 概要

D3M2010 攻撃通信データの解析から，1 回の攻撃（1 つの URL の巡回）に対し，複数の脆弱性 (CVE[1)]) が利用される傾向が確認された．この背景から攻撃に用いられる CVE の組み合わせを用いた検知手法が有効であると考えた．

6.2 解析データ

6.2.1 解析データ内のMWと脆弱性

D3M2010 攻撃通信データ [33] と D3M2011 攻撃通信データ [37] の 2 つのデータを使用した．D3M 2011 は，NTT 情報流通プラットフォーム研究所の高対話型の Web クライアントハニーポットで収集した攻撃通信データで Web 感染型マルウェアの観測データ群である．D3M2010 および D3M2011 攻撃通信データは，Web クライアントハニーポット 10 台の通信をパケットキャプチャしたファイルである．ハニーポットの OS は WindowsXP SP2，ブラウザは Internet Explorer 6.0，プラグインが Adobe Reader ， Flash Player ， WinZip ，QuickTime，JRE であり，何れもセキュリティパッチは未適用である．巡回対象 URL は公開されているブラックリスト (MDL[13]) に登録されている URL の中から，各データ収集日に攻撃を検知した URL を予め抽出したものを用いている．

脆弱性とファイルの復元は，jsunpack-n[34] を使用し行った．データを解析した結果抽出した MW を表 6.1 に示す．MW 名はその時点での最新パターンファイルを適用したウイルススキャナ (トレンドマイクロ社製) により判定されている．MW の種類は 203 種類あった．MW の DL 数の多かった上位 10 種類を表 6.2 に示す．Web 感染型 MW の特徴としては，第

[1)] CVE とは脆弱性の識別子であり，米国の Mitre 社が脆弱性に関する情報共有のため提案したもので Common Vulnerabilities and Exposures の略である [38].

2 章のボットネットで使われていた，PE や worm のようなウイルスは少なく，代わりにトロイの木馬型や脆弱性関係のウイルスを多く発見した．今回の解析結果からは，攻撃に利用される MW は総 MW 数に対するユニーク数の数が多いことから，攻撃ごとの特徴は見られなかった．

表 6.1: 各観測日から抽出した MW

観測日	2010/3/8	3/9	3/11	2011/3/8	3/14	3/16
総 MW 数	309	316	323	63	79	139
ユニーク MW 数	115	112	105	31	50	50

表 6.2: MW リスト (上位 10 件)

MW 名	DL 数
HEUR_PDFEXP.B	85
EXPL_EXECOD.A	83
Expl_ShellCodeSM	78
TROJ_PIDIEF.SMZB	46
JS_EXPLOIT.SMDX	32
TROJ_PIDIEF.SMAA	27
JS_ONLOAD.SMD	27
TROJ_KRAP.SMEP	24
JS_FPRAY.SMA	24
TROJ_PIDIEF.SML	20

次に各観測日毎の脆弱性の種類と数を表 6.3 に示す．脆弱性の種類は 2010 年 14 種類，2011 年は 11 種類であった．2011 年に新たに確認された脆弱性は CVE-2010-0806 の 1 種類のみであった．脆弱性は 2006 年，2007 年，2008 年のような古いものに関しても攻撃に利用されている．これら結果から，攻撃に利用される脆弱性の種類は大きく変わらないことが分かる．

6.3 解析結果

6.3.1 脆弱性の組み合わせ

表 6.3 の 2010 年の脆弱性について，スロット毎の脆弱性の攻撃のパターンを表 6.4 に示す．表 6.4 は出現頻度の高かった上位 10 パターンである．例えば，脆弱性パターン No.1 CVE-2008-2992 → CVE-2007-5659 → CVE-2009-0927 は，全て Adobe の脆弱性である．CVE-2008-2992 は Javascript 関数を呼び出す PDF ファイル処理に関するバッファオーバーフロー

表 6.3: 各観測日毎の脆弱性の種類と数

脆弱性	2010/03/08	3/09	3/11	2011/3/8	3/14	3/16
CVE-2005-2127	5	5	5	0	1	1
CVE-2006-0003	11	11	12	4	4	4
CVE-2006-3730	3	4	3	0	2	2
CVE-2006-5820	0	2	0	0	1	1
CVE-2007-0024	5	5	5	0	0	0
CVE-2007-5659	50	50	61	1	2	1
CVE-2008-0015	19	20	11	0	0	0
CVE-2008-2463	10	8	9	0	0	0
CVE-2008-2992	27	29	46	1	2	1
CVE-2009-0927	29	32	44	2	3	2
CVE-2009-1136	13	13	9	0	2	3
CVE-2009-1492	13	14	13	0	0	0
CVE-2009-4324	8	7	12	4	5	4
CVE-2010-0249	15	14	7	6	6	7
CVE-2010-0806	0	0	0	0	1	0
NO-MATCH	83	94	103	11	17	38
計	291	308	340	29	46	64

の脆弱性，CVE-2007-5659 は JavaScript メソッドにおけるバッファオーバーフローの脆弱性，CVE-2009-0927 は任意のコードを実行される脆弱性である．この 3 つの脆弱性を用いて，最後に MW の DL が行われる．

ユーザーが Web サイトにアクセスし，リダイレクトされた後の攻撃の流れについて調査した．その結果，攻撃の流れが 3 パターンが存在することを確認した．

攻撃パターン A**.** 脆弱性を複数突かれた後に MW を DL する．

攻撃パターン B**.** MW-A を DL 後に脆弱性を複数回突かれ，その後 MW-A を再び DL する．

攻撃パターン C**.** MW-A を DL 後に脆弱性を複数回突かれ，その後 MW-B を DL する．

表 6.4: 脆弱性の攻撃パターン (上位 10 件)

No.	脆弱性パターン	出現回数
1	CVE-2008-2992 → CVE-2007-5659 → CVE-2009-0927	68
2	CVE-2008-2992 → CVE-2009-4324 → CVE-2007-5659	23
3	CVE-2008-2992 → CVE-2009-4324 → CVE-2007-5659 → CVE-2009-0927	23
4	CVE-2008-2992 → CVE-2009-4324 → CVE-2009-0927	23
5	CVE-2009-4324 → CVE-2007-5659 → CVE-2009-0927	23
6	CVE-2008-2463 → CVE-2007-5659 → CVE-2009-0927	15
7	CVE-2008-2463 → CVE-2008-2992 → CVE-2007-5659	15
8	CVE-2008-2463 → CVE-2008-2992 → CVE-2007-5659 → CVE-2009-0927	15
9	CVE-2008-2463 → CVE-2008-2992 → CVE-2009-0927	15
10	CVE-2008-2992 → CVE-2009-0927 → CVE-2008-2992	14

第7章
結論と今後の課題

7.1 結論

7.1.1 連携感染を判定する発見的手法について

本論文では，CCC DATAset 2009 攻撃通信データにおける，感染種類を判定する発見的手法を報告した．その中で UDP 感染，連携感染などのいくつかの有益な特徴を発見した．MW のダウンロード方式にもいくつかの種類があり，それらを識別するルール，アルゴリズム (決定木) と発見的手法を提案し，評価データによる検出精度を明らかにした．学習データに対して，2/28(7%) の誤検知 (FP) があったが，未検知 (FN) はなく，十分な精度が得られる手法である．

7.1.2 Drive-by-download 攻撃の分類

第 3 章では D3M2010 攻撃通信データにおける，パスを用いた通信の振る舞いについて報告した．この攻撃パターンにはそれぞれ特徴があり，例えば，パスに pdf.php（$A3$）を含む攻撃では，多くの脆弱性が使われていた．従って，脆弱性の関連性から攻撃パターンを正しく識別することは困難である．

7.2 課題

7.2.1 ボットネットの検知について

この研究では MW のダウンロードに対する，2.2 節で述べた文字列検索に重点をおいて行ったが，文字列だけでは通常の通信を含めた場合の感染判定は難しい．また，特徴量を用いたシグネチャによる検知では，日々進化する攻撃に対応することが難しいため，他の検知手法を考える必要がある．

7.2.2 Drive-by-download 攻撃の検知について

解析に用いたデータ量が少ないため，解析結果が有効であるかどうか見極めることが難しい．今後の課題としては，クライアント型ハニーポットを作成し長期間のデータを取得した上で，apriori や prefixspan のようなデータマイニングを用いる必要がある．

参 考 文 献

[1] FBI Operation Ghost Click:
http://www.fbi.gov/news/stories/2011/november/malware_110911

[2] Gumblar と類似した Web サイト改ざんを利用する攻撃:
http://www-935.ibm.com/services/jp/ja/it-services/conspov/
jp-gr-iss-weekly-soc-report-20100204.html

[3] 危険な Web サイトの世界分布:
http://www.mcafee.com/japan/media/mcafeeb2b/international/japan/pdf/
threatreport/1010_MTMW_Report.pdf

[4] Internet Infrastructure Review vol.7 - IIJ:
http://www.iij.ad.jp/company/development/report/iir/pdf/iir_vol07.pdf

[5] Guofei Gu, Junjie Zhang and Wenke Lee: "Botsniffer: Detecting botnet command and control channel, " Internet Society, Proc. of Network and Distributed System Security Symposium(NDSS 2008), Feb. 2008.

[6] virustotal:
https://www.virustotal.com/

[7] aguse:
http://www.aguse.jp/

[8] chaosreader:
http://chaosreader.sourceforge.net/

[9] clamav:
http://www.clamav.net/lang/en/

[10] tcpflow:
http://www.circlemud.org/jelson/software/tcpflow/

[11] Cyber Clean Center:
https://www.ccc.go.jp/

[12] Mitsuaki Akiyama, et al: “Design and Implementation of High Interaction Client Honeypot for Drive-by-download Attacks, IEICE Transaction s on Communication”, Vol.E93-B No.5 pp.1131-1139 (2010.05)

[13] MALWARE DOMAIN LIST:
http://www.malwaredomainlist.com/

[14] 竹森，他: “ボットネットおよびボットコードセットの耐性解析”，マルウェア対策研究人材育成 ワークショップ 2008 (MWS2008)，pp. 49-54，2008.

[15] 水谷，他: “通信の状態遷移に着目したボット活動の調査”，マルウェア対策研究人材育成 ワークショップ 2008 (MWS2008)，pp. 25-30，2008.

[16] 石井，佐藤，田端，“ダウンロードホストに着目したマルウェアの活動傾向分析”，マルウェア対策研究人材育成 ワークショップ 2008 (MWS2008)，pp. 97-102，2008.

[17] 小櫻，津田，鳥居，“ウイルスのライフサイクルに着目した攻撃挙動の見える化”, マルウェア対策研究人材育成 ワークショップ 2008 (MWS2008), pp. 55-59，2008.

[18] 藤原，寺田，安部，菊池 “マルウェアの感染動作に基づく分類に関する検討”, 情報処理学会, pp. 177-182, 2008.

[19] 松木，他: “時系列分析による連鎖感染の可視化と検体種別の推測”，マルウェア対策研究人材育成 ワークショップ 2008 (MWS2008), pp. 37-42, 2008.

[20] 東角，鳥居，“DNS 通信の挙動からみたボット感染検知方式の検討”，マルウェア対策研究人材育成 ワークショップ 2008 (MWS2008), pp. 13-18, 2008.

[21] 仲小路，他: “パケット送受信における同調活動に着目した ボット感染ノードへの指令および反応活動の可視化”，マルウェア対策研究人材育成 ワークショップ 2008 (MWS2008), pp. 31-36, 2008.

[22] Guofei Gu, Phillip Porras, Vinod Yegneswaran, Martin Fong, and Wenke Lee: “BotHunter: Detecting Malware Infection Through IDS-Driven Dialog Correlation,” USENIX, Proc. of 16th USENIX Security Symposium, 2007.

[23] 畑田，中津留，寺田，篠田: “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), pp. 1-8, 2009.

[24] Network Grep，`http://ngrep.sourceforge.net/` (2009 年 10 月参照).

[25] Quinlan，J．R．：“C4.5 Progarams for Machine Learning”，Morgan Kaufmann，San Mateo, California.

[26] 並木，菊池:“ユーザビリティの高い GUI ベースの決定木学習ツール ID3E の開発”，情報処理学会第 67 回全国大会，vol．w-8，3，pp. 249-250．2005.

[27] tcpflow，`http://www.circlemud.org/~jelson/software/tcpflow/` (2009 年 11 月参照).

[28] 畑田，他:“複数観測データを用いたボットネットの活動分析に関する一考察”，マルウェア対策研究人材育成 ワークショップ 2008 (MWS2008)，pp. 87-92, 2008.

[29] 阿部義徳，田中英彦:“C&C セッション分類によるボットネットの検出手法の一検討”，FIT2007, L-033, pp. 77-78, 2007.

[30] Alexander Moshchuk, Tanya Bragin, Steven D. Gribble, and Henry M. Levy: “A Crawler-based Study of Spyware on the Web”

[31] 水谷 正慶，武田 圭史，村井 純:“Web 感染型悪性プログラムの分析と検知手法の提案”，電子情報通信学会論文誌. B, 通信 J92-B(10), 1631-1642, 2009-10-01

[32] 阪井 哲晴，寺田 真敏，土居 範久:“Web サイトに埋め込まれたインジェクション攻撃の追跡検知システムの提案”，情報処理学会研究報告，Vol.2010-CSEC-48 No.9, 2010-03-04

[33] 畑田 充弘，中津留 勇，秋山 満昭，三輪 信介:“マルウェア対策のための研究用データセット ～ MWS 2010 Datasets ～”，マルウェア対策研究人材育成ワークショップ 2010（MWS2010），2010.

[34] jsunpack-n:
https://code.google.com/p/jsunpack-n/

[35] Andrew Brandta:
“When admins attack: 30 hours in the life of a Gumblar victim”

[36] インジェクション - 3129:
http://jvnrss.ise.chuo-u.ac.jp/csn/

[37] 畑田 充弘，中津留 勇，秋山 満昭:“マルウェア対策のための研究用データセット ～ MWS 2011 Datasets ～”，マルウェア対策研究人材育成ワークショップ 2011（MWS2011），2011.

[38] CVE:
http://cve.mitre.org/

業 績 リ ス ト

[1] 桑原和也，藤原将志，菊池浩明，寺田真敏，“パケットキャプチャーから感染種類を判定する発見的手法について”，マルウェア対策研究人材育成ワークショップ 2009（MWS2009），pp.397-402，2009.

[2] K. Kazuya, Hiroaki Kikuchi, Masashi Fujiwara and Masato Terada, 4th International Workshop on Advances in Information Security (WAIS2010)，pp.603-607，2010.

[3] 桑原和也，藤原将志，菊池浩明，寺田真敏，“ボットネットの連携感染を判定する発見的手法について”，情報処理学会論文誌，Vol.51 No9，pp.1600-1609，2010.

[4] 桑原和也，安藤 槙悟，藤原将志，菊池浩明，寺田真敏，趙 晋輝，“パスシーケンスに基づく Drive-by-Download 攻撃の分類”，マルウェア対策研究人材育成ワークショップ 2010（MWS2010），pp.771-776，2010.

[5] 2009 年度マルウェア対策研究人材育成ワークショップ 学生論文発表賞，MWS 2009.

謝　辞

本論文を執筆するにあたり，多くの方から多大なる御指導，御鞭撻を賜りました．

特に，研究に関わらず私を導いて下さった東海大学情報通信学部通信ネットワーク工学科 菊池 浩明 教授に深甚なる感謝を申し上げます．

また，本研究を推進するにあたって，懇切なる御教示並びに御激励を賜りました東海大学情報理工学部情報科学科 中西 祥八郎 教授，東海大学情報理工学部情報科学科 内田 理 准教授に厚く御礼申し上げます．

さらに，東海大学・中央大学・株式会社日立製作所による合同研究プロジェクト Scanners の一員として，活発な議論及び技術的な御助言，御示唆を賜わった株式会社日立製作所 寺田 真敏 氏，藤原 将志 氏，仲小路 博史 氏，鬼頭 哲郎 氏，東海大学 松尾 俊治 氏，大類 将之 氏，Scanners OB である小堀 智弘 氏に深く御礼申し上げます．

また，マルウェアに関する共同研究を行い，有益な意見を下さった中央大学 安藤 槙悟 氏に深く感謝致します．

そして，2 年間共に楽しみ，苦しみ，励まし合い，時には研究に対して有益な意見を与えてくれた東海大学大学院工学研究科情報理工学専攻の皆様，先生がたに感謝致します．

最後に，家族に心から感謝の意を表すると共に，謝辞とさせて頂きます．