情報処理学会論文誌　Vol. 49　No. 4　1680–1691 (Apr. 2008)

定点観測によるボットネットの観測とMalwareの動的挙動解析システムの提案

堀合啓一[†1,†2]　今泉隆文[†1]　田中英彦[†2]

近年，ボットネットによるスパムメールの大量送信やDDoS攻撃，情報の奪取などの不正行為が問題となっている．ボットネットは，従来のワームやウィルスのように自動的に感染を拡大せず，Herderと呼ばれる攻撃者からの指令を受けて活動するため，その実態の把握が難しいといわれている．本論文は，ハニーポットをセンサとした定点観測の手法でボットに利用されるMalwareを捕獲し，捕獲したMalwareを安全な環境で実行することによって，その挙動を自動的に解析し，利用者に使いやすい形で表示するシステムの提案である．本システムでは，関連するログ情報を視覚化することで，複数の観測サイトの情報を統合した全般的な傾向の把握に利用できるだけでなく，個々のイベントや個別のMalwareの挙動に関する詳細な情報まで掘り下げることができるように工夫している．

A Development of the Malware Dynamic Behavior Analyzing System and BOTNET Monitoring

Keiichi Horiai,[†1,†2] Takafumi Imaizumi[†1] and Hidehiko Tanaka[†2]

Today, BOTNET activities such as massive spam E-mail spreading, DDoS attacks and stealing information from exploited PCs are major concern regarding to the internet security. Since, spreading the infection of the bots is controlled by so called HERDER via command and control mechanism, and not spreading autonomously like conventional worm, it is not easy task to understand the behavior of the malwares of the botnets. We developed a Malware dynamic behavior analyzing system which is coupled to internet monitoring sites and provide useful information to understand the BOTNET characteristics. This system is intended to provide not only overall trend information but also drilling down to detailed information of each event including the result of the Malware behavior analysis.

1. はじめに

近年，ボットネットによるスパムメールの大量送信やDDoS攻撃，情報の奪取などの不正行為が問題となっている．ボットネットとは，一種のバックドアを埋め込まれた多数のPCで構成されるネットワークの総称であり，現在では，多くの場合IRC（Internet Relay Chat）の仕組みを利用して指令を受け制御されている．「IPアドレスの2%〜2.5%程度がボットに感染している」との調査結果[1)]もあり，ボットネットが大規模なDDoS攻撃に利用された場合には，インターネットの利用に対して甚大な被害が発生する可能性がある．一方，ボットネットは，従来のワームやウィルスのように自動的に感染を拡大せず，Herderと呼ばれる攻撃者からの指令を受けて活動するため，その挙動の把握が難しいといわれている．

また，ボットネットを構成するMalwareは，開発ツールなどの流通，パッキングや暗号化アルゴリズムの適用によって，次々に新種や亜種が登場し，パターンに頼る検出だけでは困難となりつつある．さらに，Malwareの作成や配布の意図が，従来の愉快犯的な動機から犯罪組織と結び付いた営利目的へと変貌し，これによって感染活動が目立たないように工夫されたものや，頻繁にバージョンアップされるものも出現している．また，特定の組織を狙い撃ちするいわゆるスピア型の出現などもあって，ウィルス対策ソフトウェアのパターンファイル更新で対応できないケースが増えているといわれている．このため，従来のようにMalware対策をセキュリティ・ベンダに全面的に依存することができ難い状況となりつつあり，今後は，自分の組織をターゲットとしたMalwareについては，解析の一部を自ら実施せざるをえなくなる可能性がある．

一方，Malwareの解析には，専門的な知識・技術と相当な労力が必要であり，従来はセキュリティ・ベンダや，この分野に興味を持つ一部の研究者の手に委ねられていた．本研究では，ボットネットに利用されるMalwareの収集から解析までの一連の流れを自動化するとともに解析結果を利用しやすい形式で表示し，組織の管理者がMalwareの対策を立案する際に必要な情報を提供するシステムを提案するものである．

†1 防衛省技術研究本部電子装備研究所
Electronic Systems Research Center, Technical Research & Development Institute, Ministry of Defense
†2 情報セキュリティ大学院大学
INSTITUTE of INFORMATION SECURITY

以下，次章では関連研究の状況，3 章では実装を行ったシステムについて述べ，4 章では提案するシステムを使用した解析例の紹介と考察を行い，5 章でまとめる．

2. 関 連 研 究

ボットネットの挙動解析については文献 1)–3) などが知られている．文献 2) は，総務省，JPCERT/CC の支援を受け，ISP，セキュリティ・ベンダ，研究機関から構成されるボットネット研究チームによるフィールド調査の結果の報告である．

文献 3) では，ハニーポットで捕獲した Malware をグレー BOX 上で実行し，発生するトラフィックの観測を行い，特に IRC サーバとの通信に注目した解析を行っている．さらに，この結果を利用して動作する IRC クライアントを使ってボットネットを追跡し，ネットワークの地域的な広がりや規模を推定するなど，ボットネットを多面的に解析した結果を報告している．

Malware の挙動を検証するシステムの提案としては，文献 4)–7) がある．文献 4) は Windows に対してネットワークワームが感染した際の通信パケットの状態を自動的に解析するシステムを提案している．文献 5) では，通信パケットの解析に加えて，API CALL などの情報をもとにして感染したシステム内のレジストリの変化などを動的に解析しレポートする．文献 6) は，ウィルスの実行時にメモリ上に展開されたコードを解析するシステムの提案であり，コードが暗号化されていても，ウィルスが DLL をロード/アンロードするタイミングでメモリダンプを取得することで，復号化された状態で逆アセンブルを行うことができる点に注目して解析している．文献 7) では，インターネット Worm の検出・防護のため，感染，被害，拡散などを詳細に観察できる安全で便利な仮想環境を提案し，この環境で観測した Lion/Slapper Worm について，感染ターゲットのアドレス空間上の広がり，感染率，感染ホストのアドレス空間上の広がりなどの解析結果を提示している．

一方，本提案のシステムは，Malware の収集から，通信パケットおよび感染 PC 内の挙動まで自動的に解析できる点に特徴があり，Malware の指令サーバとの通信の状況（FQDN，ポート番号，ログイン名など）や感染 PC 内の変化（Malware の実行にともなって顕在化するレジストリや hosts ファイルの改ざん，プロセスの起動など）を解析し，ネットワークの管理者などが使いやすい形式で表示できる点で，従来の研究とは異なっている．また，感染 PC 内の解析項目については，必要に応じて柔軟に拡張できるように工夫している．

ログ情報の視覚化としては文献 8)–10) などが知られている．文献 8) は複数のログに対して事象の出現頻度を基に時系列に視覚化し，不正侵入などの調査支援を行うシステムを提案し，ハニーポットのログの調査に適用した例を紹介している．文献 9) は，サイバー攻撃のうちワームに焦点を絞り，IP アドレスの 2 次元マトリックス表示を利用したサイバー攻撃視覚化手法の提案と実装であり，「IP アドレスの近接関係が自然に表現できる」ことや「インターネットレベルの大局的情報とサイトレベルの詳細情報が同時に表示できる」などの特徴がある．

本提案のシステムでは，定点観測における全般状況の時系列表示から，Malware の挙動解析結果やイベントの履歴などの詳細な情報の表示までドリルダウンでき，全般状況の把握と個々のイベントの詳細な調査を簡単な操作で実行できる点に特徴がある．

3. システムの概要

3.1 システムの要件

ボットネットの挙動を観測するための定点観測システムに必要な要件を整理する．最初に定点観測のセンサを設置するネットワーク上の位置が問題となるが，本研究では，インターネットの組織単位の利用者の立場で対策立案に必要な情報を得ることを目的としていることから，一般的な法人または個人のインターネット利用者として利用できる IP アドレスへ複数個のセンサの設置することを前提とした．

次に，インターネットの定点観測システムでは，情報を収集するセンサと，収集した情報を集約して表示する機能が最低限必要な機能となる．センサとしては IDS（不正侵入監視装置），FireWall，ハニーポットなどの利用が考えられる．本研究では，ボットネットの挙動観測を主目的としていることから，ボットのバイナリを捕獲可能なことが必須の機能要件となる．

センサで収集した情報を集約し表示する機能としては，イベント発生状況の全般的な傾向を把握するために@police のインターネット定点観測システム[11] や JPCERT/CC の ISDA（Internet Scan Data Acquisition System）[12] などと同様に横軸を時間軸とし，縦軸を何らかのイベント件数としてグラフ化した時系列の表示が必要である．また，ボットネットの挙動観測という観点からは，全般的な傾向の把握に加えて，さらに詳細な情報が必要である．たとえば，(1) いつ活動したか？（発信元 IP アドレスや，Malware バイナリの種類ごとの期間・時間帯などの観測履歴），(2) どこから来るか？（発信元 IP アドレスの利用者などに関する情報，IP アドレスの分布，センサの IP アドレスとの関係），(3) どうやって感染を広めるか？（狙われるポート番号），(4) どんな種類の Malware か？（Malware の名称），(5) どこからどんな指令を受けるか？（指令サーバのアドレスと指令内容），(6) どんな種類の攻

撃か？（スキャンのアドレス範囲，プロトコル，ポート番号）などの情報が必要と考えられる．ただし (1)〜(4) については，定点観測で何らかの情報を得ることができるものの，(5) および (6) については直接的な情報の取得は困難である．

そこで，本研究では定点観測で捕獲した Malware を Windows XP 上で実行して PC を感染させ，その挙動を観測することによって (5) および (6) の情報を得ることとした．nimda[13] や blaster[14] のように自律的に拡散する従来型のワームでは，単に Malware を実行することで感染を広めるためのスキャンの範囲やパターンを確認できるが，ボットの場合には，指令サーバと交信可能なネットワーク環境が必要となる．ボットの Malware は，バイナリの中に指令サーバなどのアドレスが FQDN の形式となっている場合が多い．この場合は，指令サーバとの交信に先立って DNS サーバをアクセスし，FQDN から IP アドレスの名前解決を行うため，Malware の実行環境の中に DNS サーバが必要となる．指令サーバとして IRC サーバが利用されることが多いが，IRC 以外に HTTP プロトコルを利用するボットの存在が報告されており[15]，また，ボットの機能としてスパムメールなどの発信も知られていることから，IRC とともに HTTP，SMTP についても模擬環境の中に含める必要がある．

模擬環境の中で Malware を実行する際に，Malware の実行にともなう感染拡大などの行為が，模擬環境からインターネットなどへ波及しないような仕組みが必須である．このため，Malware を実行する第 1 の段階としては，実ネットワークには接続しない，クローズドな環境における挙動の検証が必要となる．クローズドな環境で Malware を実行し，そのときの通信パケットを解析することによって，指令サーバの FQDN および IP アドレスと指令サーバへログインするための情報を得ることができると考えられる．ただし，模擬環境で Malware を実行しても，ボットに対する指令の内容や指令に対応した挙動の解析は不可能である．この点については，すでに知られている指令の文字列を利用する方法が考えられる．現在流通しているボットの 8 割が，agobot の亜種との報告もあるので[29]，模擬環境の中で観測できる IRC サーバへのログインの経過を観測し，そのシーケンスを文献 2) で報告されている指令文字列などから類推して解析できる可能性がある．

次に，Malware を実行して取得が必要な情報として，ボットに感染した PC 内のリソースの変化がある．その代表例として，たとえば感染した PC が再起動した際に，Malware が自動的に活動を開始するための仕組みとして行う Windows レジストリの特定の領域の書き換えなどがこれにあたる．

このほかに，Malware の実行で起動するプロセスの名称，パス，プロセス ID，リスンポートや主要なシステム関連のファイルの改ざん・削除・生成などが，感染 PC の発見や特定を

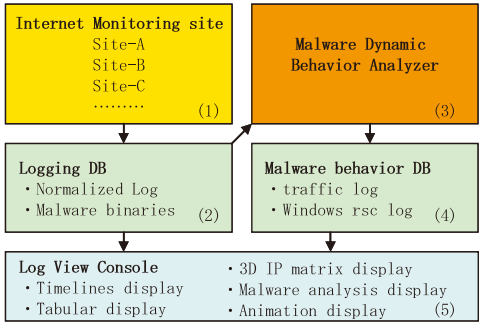

図 1　システムの全体構成
Fig. 1　An over view of proposed system.

行ううえで必要な情報である．これらの項目は，既知の Malware については，ウィルス対策製品のベンダなどから提供される情報でもあり，組織のネットワークやシステムの管理者が対策立案を行ううえで必要不可欠の情報と考えられる．

3.2　システムの全体構成

本システムは，(1) インターネット上に設置した複数のセンサ（ハニーポット），(2) センサで取得したログ情報および捕獲した Malware のバイナリ・ファイルなどから正規化ログを生成して DB へ格納する機能，(3) Malware を実行して挙動を解析する機能，(4) 挙動を解析した結果を格納した DB および，(5) システム全体の制御および視覚化されたログの閲覧機能で構成されている．最小構成では仮想マシン VMWare[16] の活用によって一般的な PC 2 台に実装可能である（観測データや解析結果の閲覧用 PC の台数は含まない）(**図 1**)．

3.3　センサの構成とログの収集

センサ用の OS として Linux を利用し，複数の IP アドレスを利用できる環境では，VMWare を利用して 1 台の物理マシンで複数個のセンサを実現した．センサには，ミディアム・インタラクション型のハニーポットの一種である Nepenthes[17] と，IDS として広く利用されている Snort[18] を使用した．Nepenthes は Windows の脆弱性を模擬し，攻撃を受けると受け側の応答をエミュレートして攻撃のパケットを記録する．この攻撃が，あらかじめ用意されている応答手順に含まれる場合には，さらに Malware の本体（ペイロード）をダウンロードする．Nepenthes では作動状況を示すログの記録に加えて，ダウンロードしたペイロードがバイナリ・データとして取得できる．複数の観測サイトで収集したログと Malware の本体を転送する手段として，UNIX の標準的なツールである RSYNC および

SSH を利用した．

3.4　ログ情報の正規化と集約処理

ログ情報は，出力する OS やアプリケーションによって各種各様であり，生のログ情報を直接処理すると，ログの種類ごとに視覚化などのソフトウェアを開発する必要があり，非効率である．このため，ログ情報の中からキーとなる項目を取捨選択し，必要に応じてデータの形式を変換する必要がある．ここで，選択した項目をログの要素と呼ぶ．正規化ログの基本要素として { タイムスタンプ，発信元 IP，あて先 IP，発信元ポート，あて先ポート，Malware の種類 } を基本とした．

正規化処理に続いてログ情報の集約処理を実行する．これは，複数サイトから集めた複数種類のログの情報からトップ 10 の算出および各種の関連付けを行う．たとえば IP アドレスと国別コードの関連付け，ダウンロードした Malware のハッシュ値とウィルススキャンした結果の関連付けなどの処理を行う．

3.4.1　出現回数のトップ 10 算出

正規化したログ情報の要素の中の注目すべき項目ごとに，一定期間内の出現回数の算出を行った．たとえば，要素の 1 つである発信元 IP アドレスに注目すると，特定の期間（たとえば 7 日間）内にログに記録されたすべてのユニークな IP アドレスごとに，それぞれ何件記録されているかを算出する．次に，算出した結果から，出現回数の多いものから順にトップ N を算出する．N の値は，図表化した際の見やすさなどの観点から，用途に応じて適当な整数（たとえば 5，10 など）を用いるが，表記上は代表してトップ 10 としている．この処理は，必要に応じて正規化ログの各要素に適用し，たとえば検出した Malware の種類トップ 10，発信元 IP アドレスの国別トップ 10 などを算出した．算出した結果は，表形式表示や時系列グラフ表示などの基本データとなる．

3.4.2　履歴の算出

イベントの調査を行ううえで過去の履歴に関する情報が必要となることが多い．このため，発信元 IP アドレスおよび Malware の種類について時間帯ごとの観測履歴の表を自動的に生成することとした．これにより，イベントが継続する期間や発生する時間帯もしくは曜日などとの相関を知ることが可能となる．

3.4.3　Malware の検出結果との融合

ハニーポットで捕獲した Malware をオープンソースのウィルス・スキャン・ソフトウエア ClamAV [19] を利用してスキャンし，ファイル名と Malware 名の対応表を生成した．スキャンの結果，Malware が検出されなかった場合は，新種または亜種である可能性が高いことから，*unknown*と表示した．

3.5　ログ情報の視覚化

ログ情報の視覚化処理は，閲覧のために専用のソフトウェアの配布が不要な Web ブラウザを利用することを前提として開発した．このため開発言語としては HTML との親和性が高い PHP [20] を利用し，グラフィックスの描画には，PHP のライブラリである JpGraph [21] を利用した．視覚化処理では，要素ごとのトップ 10 の表形式表示，時系列グラフ表示，Malware のアップデート履歴および 3.6 節で述べる Malware の動的挙動結果のレポートを作成した．

視覚化処理では 3.4 節で生成したデータから，時系列のグラフを含んだ HTML ファイルを生成する．これは，定点観測の全般的な状況の把握を目的としているが，個々のイベントなどの詳細な情報を検索する場合の操作上の入り口を兼ねている．

定点観測情報の表示画面の構成を示す**図 2** において，① は定点観測で得たイベント数の全般的な傾向を表すグラフで，3.4 節で述べた「要素」の中から 1 項目を選択して表示する．② は 3.4.1 項で算出したトップ 10 を表形式で表示するエリアであり，当日の状況と ③ で設定した期間中の累積の 2 種類を同時に表示し，さらに 2 種類のそれぞれについて攻撃元 IP アドレスと ① で選択したカテゴリ項目を表示する．このトップ 10 を示す表の HTML ファイルの中に，検索に必要な識別タグを埋め込むことで，各要素のトップ 10 が表示されてい

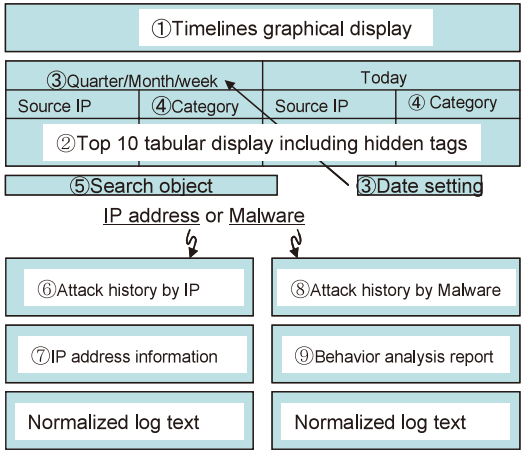

図 2　定点観測情報の表示
Fig. 2　An overview of the overall monitoring display.

る状態から，対話形式で個別のイベントなどの詳細情報の検索ができるよう工夫している．

② に表示されている IP アドレスのフィールドをクリックすることで，⑤ の検索対象として当該 IP アドレスがセットされ，⑥ に当該 IP アドレスからの攻撃の履歴，⑦ に NSLOOKUP と WHOIS で得られる IP アドレスに関する情報などの当該 IP アドレスに関連する情報を表示できる．カテゴリ項目が Malware の種類の場合には，② に表示されている Malware のフィールドをクリックすることで，⑤ の検索対象として当該 Malware がセットされ，⑧ へ当該 Malware の観測履歴と ⑨ には 3.6 節で述べる挙動解析結果のレポートを含む Malware に関する詳細情報を表示する．

3.6　Malware の実行環境

捕獲した Malware を Windows XP 上で実行させ，ネットワークアクセスと Windows システム内のリソースの状態を記録した．Malware の実行は，感染後の復旧の容易さから仮想マシンの一種である VMWare 上の Windows XP で行った．この際，ネットワーク環境の模擬は，Linux のカーネルパケットフィルタに使用されている iptables の機能を利用し，DNS，IRC，SMTP サーバの模擬は Truman [22] の一部の機能を利用した．ボットの活動状況を分かりやすく示すために，Malware を実行した際の通信を tcpdump でキャプチャし，FQDN，IP アドレス，ポート番号，IRC サーバへのログイン，パスワード，既知の指令文字列などを抽出した．また，Malware 実行前後の Windows システムにおけるレジストリや主要なシステムファイルの状態を自動的に記録し，Malware の実行にともなうこれらのファイルの改ざんなどの情報を取得した．さらに，Malware の実行にともなう起動プロセスの ID，プロセス名およびリッスンするポート番号などの情報を取得した．Malware の実行のつど，Windows XP を感染前の状態に戻す必要があるが，これには VMWare の Snapshot 機能を利用した．Malware 実行環境の構成を**図 3** に示す．

3.6.1　Malware の実行制御

Malware の挙動解析は，実行制御用のホスト（以下ホスト）OS と，このホストにインストールした仮想マシン（VMWare Server）上で作動し Malware を実行する Windows XP（以下 VictimPC）で構成され，物理的には 1 台の PC に実装している．ホスト内のスクリプトが設定したディレクトを監視し，そこにファイルが置かれていれば，VMWare の機能を利用して VictimPC を自動的に起動して解析を開始する仕組みとしている．

VictimPC の OS が起動すると続いて VictimPC 内のローダ（scLoader）が起動し，ホストからスクリプト（scVictim）をダウンロードする．次に VictimPC は scVictim を実行し，VictimPC 内の情報取得に必要な software tool をホストから受信し，Malware 実行前のログを取得する．続いて解析対象の Malware をホストから受信してこの Malware を実行し，指定した時間経過後に Malware 実行後のログを取得する．実行制御のホストと VictimPC 間は，host-only 接続の TCP/IP 通信（ftp，http）を利用しているが，Malware の実行によって VictimPC が予期できない状態となり，ホストからの制御が不安定となる場合も発生する．このような状態の検出と復旧のため，VMWare のホスト OS ⇔ ゲスト OS 間の直接 I/O ポートを監視してゲスト OS 側の起動状態を取得し，かつ一定期間にわたって状態を取得できない場合には，強制的にゲスト OS 側をリセットする機能を実装した．

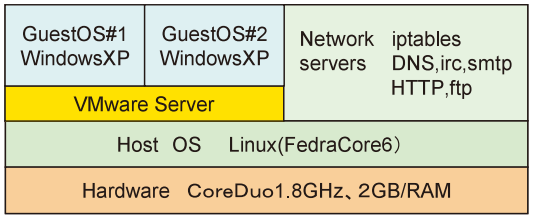

図 3　Malware 実行環境の構成
Fig. 3　Structure of the Malware execution environment.

最後に取得したログは VictimPC からホストへ転送され，転送されたログは DB へ蓄積される．結果を閲覧する際に，ホスト上のスクリプト（scRscChg）で Malware 実行前後の変化部分のみを解析結果として HTML 形式で出力し Web ブラウザで検索・閲覧可能とした．Malware 実行のフローを**図 4** に示す．

定点観測のシステムで，新規のハッシュ値を持った Malware を捕獲した場合には，このファイルが自動的に解析のキューへ転送され，Malware の解析が実行される仕組みとしている．1 個の Malware の解析に必要な時間は数分程度である．

3.6.2　Windows システム内の情報取得と利用ツール

Malware が感染した際の情報を取得するには，このためのソフトウェア・ツールが必要である．また，一連の解析処理を制御する仕組みが必要となるが，**表 1** に本システムで開発・利用したツールを示す．

表 1 で * つきの Function は VictimPC 上で実行される機能であり，これらのツールをあらかじめ解析対象の VictimPC へインストールしておくことも考えられるが，OS の種類やバージョンの違いなどが，感染 PC の挙動に及ぼす影響を調査する場合には，個々の VictimPC へツールをインストールする必要がある．また，挙動解析の項目を追加する場合

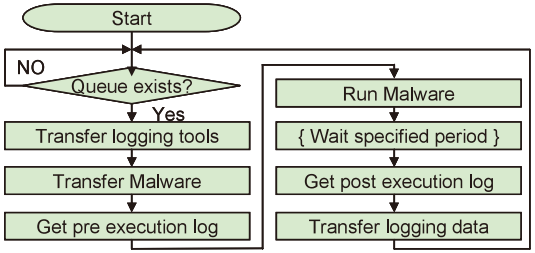

図 4　Malware 実行のフロー
Fig. 4　Malware Execution macro flow chart.

表 1　利用したソフトウェア・ツールの例
Table 1　Software tools utilized.

Function	Tool	Source
Overall Execution control	scExec	in house scripts
Find resource changes	scRscChg	in house scripts
*Loader	scLoader	in house scripts
*VictimPC Execution Control	scVictim	in house scripts
*File integrity	scMd5sum	in house scripts
*Registry	autorunsc	Microsoft[23]
*Process	pslist	Microsoft[23]
*Listen ports	Fport	Fundstone[24]

には，これに必要なツールを個々の VictimPC へインストールすることが必要となり非効率である．このような作業の効率化を図るため VictimPC 内へ 3.6.1 項で述べた scLoader を実装した．scLoader がホストからダウンロードして実行する scVictim は，**図 5** に一例を示す定義ファイルを参照して動的に生成する仕組みとしている．これによって，解析対象の VictimPC が複数種類存在する場合でも，scLoader だけが個別のインストールの対象となり，準備のための時間を短縮できる．

図 5 の例では ⟨tool_d⟩ のセクションで，取得するログ情報と利用するツールの対応を定義している．次に ⟨inspection⟩ のセクションで Malware の実行前に起動プロセスとリスンポートに関するログを取得し，Malware の実行後に 180 秒間待ってからレジストリ，起動プロセス，リスンポート，ホストファイルおよび別に定義した主要なシステム関連ファイルのハッシュ値（md5sum）に関するログを取得することを意味している．

```
<!--  # Tool definition -->
<tool_d>
              <tool>process, pslist.exe</tool>
              <tool>listenport, fport.exe</tool>
              <tool>registry, autorunsc.exe</tool>
              <tool>hosts,   script_host1</tool>
              <tool>md5sum,  script_md5</tool>
</tool_d>

<!-- # inspection items definition →
<inspect>
              <prior>
                            <check>process</check>
                            <check>listenport</check>
              </prior>
              <delay>180sec</delay>
              <following>
                            <check>registry</check>
                            <check>process</check>
                            <check>listenport</check>
                            <check>hosts</check>
                            <check>md5sum</check>
              </following>
</inspect>
```

図 5　取得する情報とツールの定義例
Fig. 5　An example of tools and checking items definition.

4. システムの使用例と考察

4.1 システムの使用例 1（定点観測）

定点観測システムの複数センサを含む全般状況の表示例を**図 6** に示す．この図は WEB ブラウザで閲覧するが，画面の内容は，設定した時間ごとに自動的に更新される（以下 IP アドレスの一部は伏せ字としている）．

図 6 の最上部（Site:all）は，各センサからの情報を総合したグラフであり，以降各センサのグラフが続いて表示される．この図では，捕獲した Malware の種類の 1 週間の推移を示している．

図 6 の特定のセンサのグラフをクリックすることで，**図 7** に示すように，当該センサの時系列グラフとトップ 10 の表示にドリルダウンできる．図 7 の表示から IP アドレスまたは Malware をクリックすることで，さらに詳細な情報の表示画面へドリルダウンできる．

たとえば，IP アドレスを選択した検索では，**図 8** に示すように，当該 IP アドレスの観測履歴と IP アドレスの利用者に関する情報を表示できる．履歴の表は，センサのハニーポッ

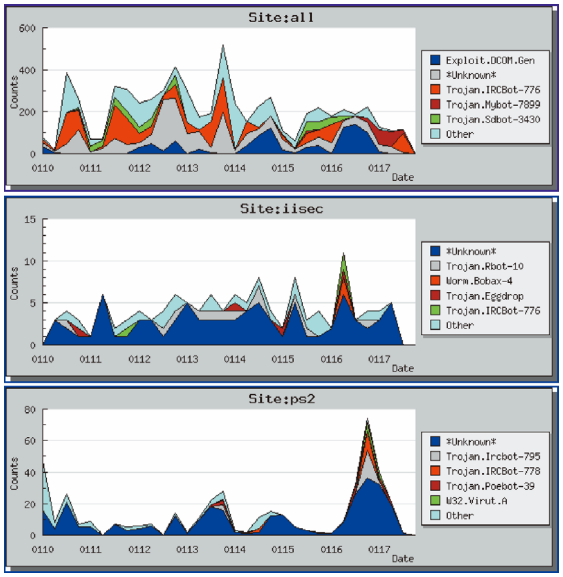

図 6　定点観測の全般状況表示例
Fig. 6　A screenshot of overall situational display.

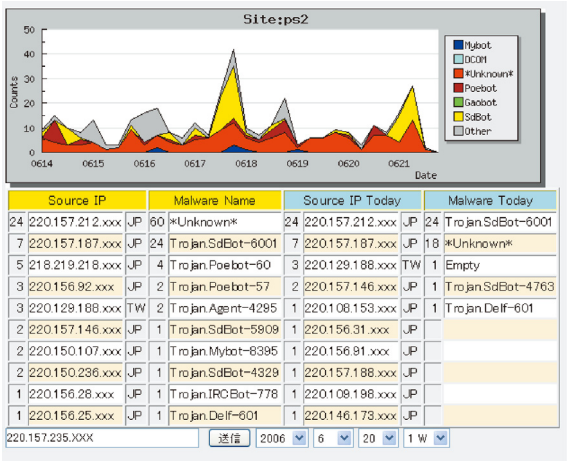

図 7　時系列表示とトップ 10 表示例
Fig. 7　A screenshot of the timelines and Top-10 display.

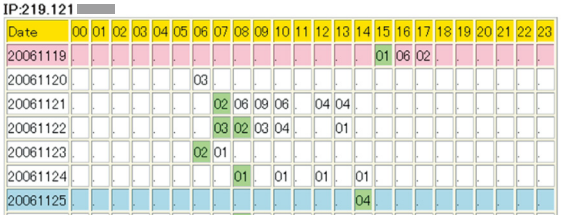

図 8　IP アドレスの観測履歴表示例
Fig. 8　A screenshot of the attacking history from specified IP address.

```
{Whois information}
a. [Network Number]        219.121.107.0/24
g. [Organization]          YYY CORPORATION
{truncated, nslookup information}
Non-authoritative answer:
xxx.xxx.121.219.in-addr.arpa name = usrXXX.bb011-
03.udc.im.wakwak.ne.jp.
{truncated...}
```

図 9　IP アドレスの利用者情報の表示例
Fig. 9　An example of the IP address's user information.

トが当該 IP アドレスからの攻撃を観測した年月日，1 時間ごとの攻撃回数および捕獲した Malware のハッシュ値の変化を示す．この例では，2006 年 11 月 19 日の 15 時台に最初の攻撃があり，同 11 月 21 日以降ほぼ毎日 Malware のハッシュ値が変化（着色部分が変化を示す）している．また，この IP アドレスからは，昼間帯の活動が活発であることが分かる．

履歴に続いて，当該 IP アドレスの利用者に関する情報を whois および nslookup で検索した結果が**図 9** のように表示される（一部，省略および伏せ字としている）．

以上のような，履歴や IP アドレスの利用者に関する情報は，あるイベントに関する調査を行ううえで頻繁に必要となる行為であることから，本システムではこれらの処理を自動化している．

4.2　システムの使用例 2（Malware 解析レポート）

Malware の解析は，基本的には定点観測で捕獲したバイナリが本システムで未解析の場合に，自動的に解析の待ち行列へ追加されて逐次解析が行われる．

ここでは，図 7 で示した定点観測のトップ 10 表示から，指定した Malware に関する解析結果のレポートを出力する例を説明する．トップ 10 の表示画面でたとえば Trojan.Sdbot-3012

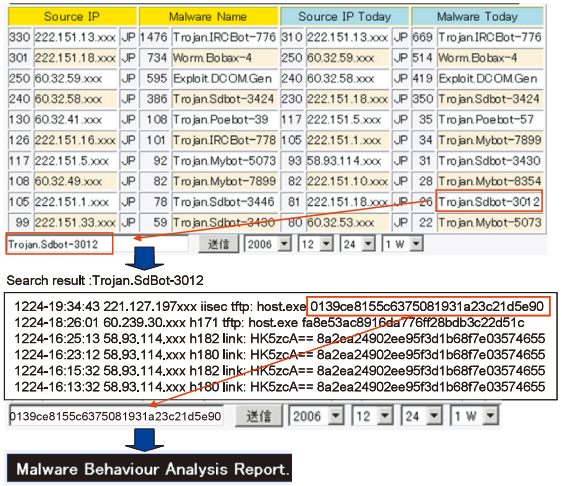

図 10　Malware の挙動解析結果の検索例
Fig. 10　A screenshot of the searching behavioral analysis report by Malware hash value.

をクリックすると，当該 Malware の観測履歴と Trojan.Sdbot-3012 を含む正規化ログが表示される．この様子を**図 10** に示す．

この例では Malware の名称 Trojan.Sdbot-3012 に対し，正規化ログの中にファイルのハッシュ値が異なる 3 種類のファイルが含まれていることが分かる．そこで，この中の 1 個を再度検索の対象として操作を繰り返すことによって，Malware の挙動の解析結果を表示する仕組みとしている．

以下，非常に数多くの亜種の存在が知られている AGOBOT（ハッシュ値 1f0e458c6852dc7e031cc1e005daf6 b4:PE_AGOBOT.AQM）および定点観測で捕獲数の多かったハッシュ値 f7f1e7c55fe828dcf27ef04f9adaedbf（PE_BOBAX.AH）の 2 種類の Malware を例として解析結果のレポートを示す．

4.2.1　PE_AGOBOT.AQM

PE_AGOBOT.AQM（別名：アゴボット，Back door.Sdbot，W32/Mytob, Win32.Agobot)[25] はウィルス対策ベンダの報告では，MS03-026（RPC/DCOM)[26]，MS03-001（RPC/Locator)[27]，WebDAV（MS03-007)[28] などの脆弱性をターゲットとして感染する．AGOBOT は Gaobot，Phatbot や Polybot などと呼ばれる亜種も含めて千

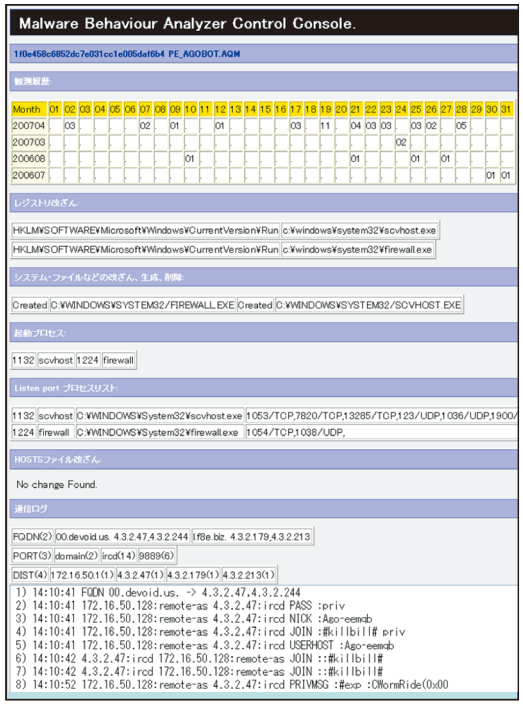

図 11　PE_AGOBOT の解析レポート表示例
Fig. 11　A screenshot of the analysis report: PE_AGOBOT.

を超える多種類のバージョンが検出されていて，最近のボットの 8 割は Sdbot と Gaobot の亜種とのレポートもある[29]．

図 11 に解析結果のレポート出力画面を示す．観測履歴は，定点観測で捕獲した日々の件数を示している．

この図から，2 項目がレジストリ HKLM¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run へ追加され，C:¥windows¥system32 の中に 2 個のファイル（firewall.exe，svchost.exe）を生成していることが分かる．また，起動プロセスの ID，プロセス名称，当該プロセスの PATH とともに，ネットワークをリッスンするポート番号と TCP/UDP の種別が示されている．一部の Malware は，感染した PC のウィルス対策ソフトウェアのパターンファイルの更新や，OS のセキュリティパッチなどを妨害するために，

Windows の hosts ファイルを改ざんするが，本実行例では，hosts ファイルの改ざんは検出されなかった．続いて通信パケットの解析結果が通信ログの欄に表示されている．感染した PC が，C&C サーバと通信するために FQDN の名前解決を行い，C&C サーバにログインする様子が記録されている（図 11 および図 12 における 172.16.50.128 は感染 PC の IP アドレスであり，4.3.2.XXX は，模擬環境の DNS サーバが割り当てた仮想の IP アドレスである）．

4.2.2　PE_BOBAX.AH

PE_BOBAX.AH は 2005/09/07 に発見され，MS04-011（LSASS）[30] の脆弱性に関連して拡散する Malware で，メールを大量に発信する特徴を有している．また，レジストリ値を追加し hosts ファイルを改ざんすることが知られている[31]．**図 12** に解析結果のレポート出力画面を示す．

この図から 2 項目がレジストリ HKLM¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run へ追加され，C:¥windows¥system32 の中に 2 個のファイル（WINAMP.EXE，OIOEKZZECLPY.EXE）が追加されていることが分かる．さらに，hosts ファイルが改ざんされ，ウィルス対策ソフトウェアの更新や，OS のセキュリティパッチなどに関連したアドレスが 255.255.255.255 へリダイレクトされていることが分かる．

続いて通信パケットの解析結果が通信ログの欄に表示されている．図 12 における通信ログの最後の表示領域において，1)〜3) で FQDN の名前解決を行い，4) では http のポートで Malware バイナリの更新と推定される通信が発生している．次に 5) で新たな FQDN の名前解決を行い，6)〜8) で http プロトコルを使って C&C サーバと通信を行い，11) でメールを送信する．この Malware は，その後も FQDN 名前解決，http 通信，メール送信のパターンを繰り返す．

4.3　定点観測と Malware の動的挙動解析のまとめ

本提案のシステムを使用し，捕獲した 5,000 個体以上の Malware を解析した結果のまとめを以下に述べる．収集した Malware を ClamAV でスキャンしたが，**表 2** に示すように，ほとんどがボット関連の Malware として分類され，全体の約 15%が未知の Malware と判定された．また，併設した snort で観測した結果，感染を広げるための攻撃に利用されたポートは**表 3** のとおり 135，139 および 445 の 3 種類で大半を占めている．必要のない限り，これらのポートは閉じておくことで，新たな感染の可能性を低減させることができる．

次に Malware の実行にともなって顕在化した Windows システム内の改ざんなどについて示す．**表 4** は，生成されたファイルのトップ 5 を示し，**表 5** は改ざんされたレジスト

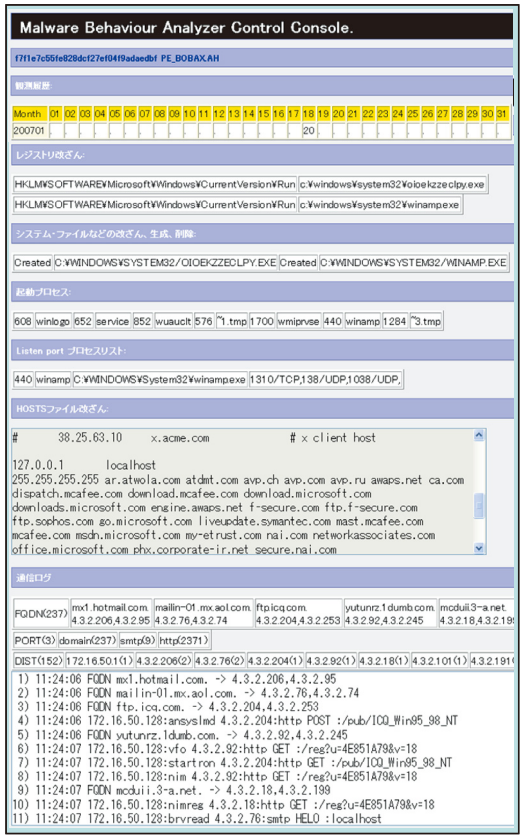

図 12　PE_BOBAX.AH の解析レポート表示例
Fig. 12　A screenshot of the analysis report: PE_BOBAX.AH.

表 2　Malware の分類
Table 2　Malware classification by clamav.

Name	SdBot	Unknown	Mybot	Poebot	Bobax	Other
ratio	22%	15%	12%	10%	8%	33%

表 3　感染に利用されたポート
Table 3　Port used for attacks.

Port	445	139	135	1433	Other
ratio	37.4%	31.1%	18.7%	2.3%	10.5%

表 4　生成されたファイル名トップ 5
Table 4　Top 5 of files that is created.

ratio	Created files
12.4%	C:¥WINDOWS¥SYSTEM32¥VCMGCD32.DLL
4.4%	C:¥WINDOWS¥SYSTEM32¥SPOOLSVC.EXE
4.4%	C:¥WINDOWS¥SYSTEM32¥LSSAS.EXE
4.2%	C:¥WINDOWS¥SYSTEM32¥WINIOGON.EXE
4.2%	C:¥WINDOWS¥SYSTEM32¥ALGS.EXE

表 5　改ざんされたレジストリ KEY トップ 5
Table 5　Top 5 of Windows Registry KEY that is modified.

ratio	Registry KEYs
14.0%	HKLM¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run@kqtpqeqgvtixahnd
7.7%	HKLM¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run@Spooler
7.5%	HKLM¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run@Windows
7.4%	HKLM¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run@Local
6.6%	HKLM¥SOFTWARE¥Microsoft¥Windows

リ KEY のトップ 5 を示している．これらの情報は，感染が疑われる PC を調査する際に，Windows システム・フォルダ内のファイルの存在，レジストリ KEY の値の確認などの方法で，感染の有無の判断の一要素として利用できる．

また**図 13** に，Malware の中で Windows の hosts ファイルおよびレジストリを改ざんする Malware の割合の時間的変化を示す（太線は移動平均）．この図は，個々の Malware の動的挙動解析結果と定点観測の履歴ログから生成したものであり，上側のプロットはレジストリを改ざんした Malware の割合を示し，全体の 70%前後で推移している．一方，下側のプロットは hosts ファイルを改ざんした Malware の割合を示し，ウィルス対策ソフトのパターン更新や WindowsOS のパッチを妨害する Malware の割合が増加の傾向にあることが読み取れる．

次に Malware の実行にともなって発生したトラフィックの解析結果と定点観測の履歴ログから生成した，ボットが利用するポートのトップ 5 の変化を**図 14** に示す．ここでポート 65520，8585，6667 は IRC サーバとの通信に利用され，80（HTTP）は Malware の更新

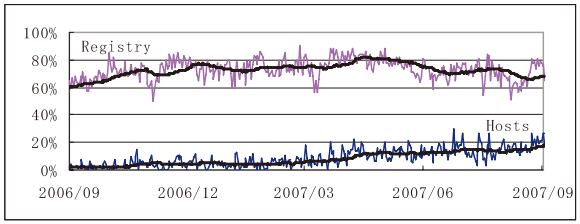

図 13　ファイルを改ざんする Malware の割合の変化
Fig. 13　Changes of Malware ratio that modifies system files.

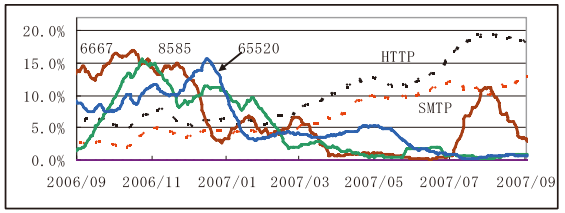

図 14　ボットが利用するポートの推移
Fig. 14　Changes of ports that are used by bot.

と C&C 機能の一部としても利用されている．25（SMTP）については，Malware の起動時に特定のメールアドレスへメールを送信するシーケンスがたびたび観測されたが，Malware の起動条件のチェックの 1 つと考えられる[32),33)]．図 14 から http とともに，smtp を利用する Malware が増加の傾向にあることが確認できる．

IRC サーバとの通信に利用されるポートについては，文献 2) とはやや異なったものとなった．原因としては，捕獲の環境（ハニーポットの種類，センサの IP アドレス）の違いなどの影響と，特に観測した期間が違うことで，インターネット上で実際に活動しているボットの種類が変化していることを反映しているものと推定される．ボットに感染した PC は，これら C&C に利用されるポートを使って herder から指令を受けるので，これらのポートの利用を制限することで，ボットネットの影響を低減させることができるが，C&C のポートに着目した対策を実施する場合には，その変化に対応できるよう観測の継続や情報の更新が必要と考えられる．

4.4　模擬環境における Malware の挙動解析の限界

近年の Malware の中には，解析を回避するための機能を備えている種類の存在が知られ

表 6　PC 内のデータを取得できない Malware トップ 5
Table 6　Top 5 of Malware that could not get data from PC.

ratio A/B	Malware	NoData(A)	Sample(B)
100.0%	WORM_RBOT.EIC	14	14
91.7%	WORM_RBOT.DSU	11	12
83.9%	PE_PARITE.A	78	93
66.7%	PE_TENGA.A	4	6
52.9%	PE_VIRUT.D	9	17

ている．たとえば仮想マシン上の環境では動かないものや，実際のインターネットと接続されているかどうか確認し，その挙動を変化させることが考えられる．したがって，本システムのようにクローズドで安全な環境で Malware を実行する方法のみでは，挙動の解析に限界がある．本システムで捕獲したハッシュ値の異なる 5,000 個体以上の Malware を実行したが，VictimPC のフリーズやリブートによって，一部の Malware については，PC 内からログを取得できなかった．本システムを利用して，PC 内のデータを取得できなかった既知の Malware（名称はトレンドマイクロ社製のウィルスバスター 2007 Trend Flex Security による）のトップ 5 を**表 6** に示す．

この表における (A) は，PC 内からデータを取得できなかった Malware のサンプル数，(B) はサンプルの総数であり，WORM_RBOT や PE_PARITE などの亜種の中に耐解析機能をする Malware が存在する可能性を示唆している．

このように，自動的に解析できない場合もあるが，捕獲した Malware 5,158 固体中，4,964 個についてはウィルス対策製品のベンダなどから得られる情報と同等または補完するデータの取得が可能であり，得られた情報はネットワーク上の対策立案や感染 PC の発見などに活用できるものと考えられる．

5. おわりに

本研究では，ハニーポットを利用した定点観測によって，ボットネットを構成する Malware を捕獲し，捕獲した Malware を実行させて，挙動の解析を自動化するシステムの構築を行った．正規化したログ情報から，時系列表示，履歴表示などによる視覚化を行うとともに，捕獲した Malware の実行結果のログを解析し，C&C サーバの FQDN，利用するポート番号，ログイン名などを抽出した．また，Windows システムの hosts ファイル，レジストリやシステム関連ファイルの改ざんなどの状況，起動プロセスに関する情報や待ち受ける通信ポートの状況など，ボットネットに利用される Malware の特徴およびその変化に関する知見を得た．また，このシステムを利用して得られる Malware の特徴を示す情報は，ウィルス対策製品のベンダなどから提供される内容を補完するものとして，ネットワークやシステムの管理者が対策を立案するうえで，有効と考えられる．

本システムを利用することにより，Malware の実行にともなう通信パケット（通信先，プロトコル，パケットサイズなど）や Windows システム内のリソース（レジストリ，起動プロセスなど）の変化など，Malware の挙動に関する多次元の情報を得ることができる．これらの情報を利用して行う Malware 自体の詳細な分析や，その分析結果を用いた Malware の検出と種類の自動判定の手法の検討が今後の課題である．また模擬環境では実行できない Malware の解析手法，ボットネットの制御の仕組みとして IRC 以外のプロトコルの解明などが今後の課題である．

参　考　文　献

1) Honeynet project: Know your Enemy: Tracking Botnets. http://www.honeynet.org/papers/bots/
2) 高橋正和，村上純一，須藤年章，平原伸昭，佐々木良一：フィールド調査によるボットネットの挙動解析，情報処理学会論文誌，Vol.47, No.8 (2006).
3) Rajab, M.A., Zarfoss, J., Monrose, F. and Terzis, A.: A Multifaceted Approach to Understanding the Botnet Phenomenon, *IMC'06*, October 25–27 (2006).
4) 寺田真敏，高田眞吾，土居範久：ネットワークワーム自動検証システムの提案，情報処理学会論文誌，Vol.46, No.8 (2005).
5) 吉岡克成，衛藤将史，井上大介，中尾康二：Macro-Micro Correlation Analysis for Binding Darknet Traffic and Malwares, *SCIS2007* (2007).
6) 井上大介，衛藤将史，吉岡克成，星澤裕二，伊沢亮一，森井昌克，中尾康二：Micro Analysis for Analyzing Malware Code and its Behavior on NICTER, *SCIS2007* (2007).
7) Jiang, X., Xu, D., Wang, H.J. and Spafford, E.H.: *Virtual Playgrounds For Worm Behavior Investigation*, Microsoft Research.
8) 江端真行，小池英樹：不正侵入調査を目的とした複数ログの時系列視覚化システム，情報処理学会論文誌，Vol.47, No.4 (2006).
9) 大野一広，小池英樹，小泉　芳：IP Matrix：広域ネットワーク監視のための視覚化手法，情報処理学会論文誌，Vol.47, No.4 (2006).
10) Ball, R., Fink, G.A. and North, C.: *Home-Centric Visualization of Network Traffic for Security Administration*, ACM (2004).
11) http://www.cyberpolice.go.jp/detect/observation.html

12) JPCERT/CC. http://www.jpcert.or.jp/isdas
13) Nimda. http://www.microsoft.com/japan/technet/security/alerts/nimda.mspx
14) Blaster. http://www.microsoft.com/japan/technet/security/alerts/blaster.mspx
15) http://www.cyberpolice.go.jp/detect/pdf/H170127_botnet.pdf
16) VMWare. http://www.VMWare.com/
17) Nepenthes. http://nepenthes.mwcollect.org/
18) snort. http://www.snort.org/
19) ClamAV AntiVirus. http://www.clamav.net/
20) PHP. http://www.php.net/
21) JpGraph. http://www.aditus.nu/jpgraph
22) Stewar, T.J.: Truman, The Reusable Unknown Malware Analysis Net. http://www.lurhq.com/truman/
23) http://www.microsoft.com/technet/sysinternals/default.mspx
24) http://www.foundstone.com/us/resources-free-tools.asp
25) http://www.trendmicro.co.jp/vinfo/virusencyclo/default5.asp?VName=PE%5FAGOBOT%2EAQM
26) http://support.microsoft.com/kb/823980/ja
27) http://support.microsoft.com/kb/810833/ja
28) http://support.microsoft.com/kb/815021/ja
29) http://www.itmedia.co.jp/enterprise/articles/0704/13/news020.html
30) http://support.microsoft.com/kb/835732/ja
31) http://www.trendmicro.co.jp/vinfo/virusencyclo/default5.asp?VName=PE%5FBOBAX%2EAH&VSect=P
32) http://www.ipsj.or.jp/10jigyo/seminar/2007/2007-1.html
33) http://www.jpcert.or.jp/research/2007/P2P_bot_analysis_report.pdf

(平成 19 年 8 月 13 日受付)
(平成 20 年 1 月 8 日採録)

堀合　啓一（正会員）

1973 年防衛大学校電気工学科卒業．同年航空自衛隊入隊．1979 年防衛大学校理工学研究科電子工学専攻修了．2005 年から防衛省技術研究本部電子装備研究所勤務．現在，情報セキュリティ大学院大学博士課程在学中．

今泉　隆文

2004 年東京大学工学部計数工学科卒業．2006 年同大学大学院情報理工学系研究科博士前期課程修了．同年防衛省技術研究本部入省．現在，同省電子装備研究所勤務．

田中　英彦（フェロー）

1970 年東京大学大学院博士課程修了，工学博士．東京大学大学院情報理工学系研究科教授・研究科長を経て，2004 年情報セキュリティ大学院大学教授．計算機アーキテクチャ，分散処理，知識処理，デペンダブル情報システム等に興味を持つ．著書に『非ノイマンコンピュータ』『計算機アーキテクチャ』『Parallel Inference Engine』等がある．日本学術会議会員，電子情報通信学会，情報処理学会，人工知能学会，IEEE 各フェロー．