情報処理学会論文誌 Vol. 50 No. 9 2360–2370 (Sep. 2009)

推薦論文

ボットおよび指令サーバのホスト型追跡

竹森敬祐[†1] 藤長昌彦[†1]
佐山俊哉[†1] 西垣正勝[†2]

昨今，指令サーバに制御される多数のボットを踏み台にした送信元 IP アドレス詐称パケットによる DDoS 攻撃が脅威となっている．ボットは未知のコードが多数あること，感染 PC に直接的な被害を及ぼさないことで，その感染が見過ごされてしまう傾向がある．これまで，攻撃の発信源を特定する技術として，インターネット上に専用のプローブを設置して，被害者 PC から加害者 PC を特定する IP 追跡方式が提案されている．しかしボットネットの場合，指令サーバからボットへの制御とボットからの攻撃は，異なる通信アプリケーションで非同期的に行われており，単一の通信プロトコルを想定した既存方式では，指令サーバやボットの追跡を行えないという問題がある．そこで本研究では，(i) 被害者 PC からボットへと，(ii) ボットから指令サーバへの両者に対応したホスト型の追跡方式を提案する．(i) は，被害者 PC からの申告情報を基に，自身が加害者であることを自己認識するクレームドリブンなボットの追跡方式である．(ii) は，各地のボットの通信履歴から共通する宛先情報を抽出して，活発な指令サーバを特定する連携的な追跡方式である．実際にホスト型追跡システムの実装を行い，感染 PC 上の異常な通信の特定と，インターネット上の活発な指令サーバの特定を行える様子を示す．

Host-based Traceback against Bot and C&C Server

Keisuke Takemori,[†1] Masahiko Fujinaga,[†1]
Toshiya Sayama[†1] and Masakatsu Nishigaki[†2]

Recently, DDoS attacks involving source IP spoofing have now become critical issues on the Internet. These attacks are considered to be sent from bots that are controlled by command and control (C&C) servers. As many types of unknown bots that only affects the PC slightly are released, users tend to leave them infected. There has been active research into IP traceback systems that probe packets on the Internet. However, efforts to determine traceback from victims' PCs to bots and from bots to C&C servers have not yet been achieved. Because control and attack packets are sent asynchronously, it is hard to grasp the relation between bots and C&C servers. In this research, we propose host-based traceback schemes that track (i) from a victim PC to a bot, and (ii) from the bot to a C&C server. In the case of (i), the victim PC notifies its IP address to another PCs in order to inspect their access records. The notification is considered to be a claim driven traceback scheme. In the case of (ii), bot access records are gathered and compared in order to extract the active IP address considered to be a significant C&C server. The comparison is considered to be a cooperative traceback scheme. We implement our proposed model and evaluate the tracking ability against the bot process on the infected PC and the active C&C servers on the Internet.

1. はじめに

昨今，多数の PC がボットに感染する被害が広がる中，ボットを踏み台にした送信元 IP アドレス詐称パケットによる DDoS 攻撃への脅威が高まっている．ボットの中には，正規の通信プロトコルを利用した攻撃を行うもの，ボット 1 台あたりの送信頻度が小さな攻撃を行うもの，自身のコードを頻繁に更新するものなどがあり，侵入検知システム（IDS）やアンチウイルスソフト（AV）で検知し難いボットが多い．また，感染 PC に直接的な被害を及ぼさないことで，長期間ユーザが放置してしまうという問題もある．

従来，ボットなどから発信される送信元 IP アドレス詐称パケットを追跡する手法として，IP 追跡方式が注目されている[1)–4)]．既存の IP 追跡方式は，通信経路上に設置した専用のプローブに，通過するパケット情報を保存しておくことで，攻撃を受けた時刻情報とパケット情報より，被害者 PC から加害者 PC への通信経路を遡るネットワーク型の追跡方式である．しかしボットネットの場合，ボットネットを管理するボットマスタ，指令やコードを配信する指令サーバ（本論文では，C&C サーバとダウンロードサーバを合わせて，指令サーバと呼ぶ），攻撃の踏み台となるボットから構成され，異なる通信アプリケーションで非同期的に制御や攻撃が行われることから，単一の通信プロトコルを想定した既存の追跡方式では，指令サーバまで遡って追跡できないという技術的な課題がある．また，DDoS 攻撃の発

†1 株式会社 KDDI 研究所
KDDI R&D Laboratories
†2 静岡大学創造科学大学院
Graduate School of Science and Technology, Shizuoka University
本論文の内容は 2008 年 7 月のマルチメディア，分散，協調とモバイル（DICOMO2008）シンポジウムにて報告され，同プログラム委員長により情報処理学会論文誌ジャーナルへの掲載が推薦された論文である．

信源である多数のボットを個々に経路をたどりながら追跡することになり，効率面での課題もある．さらに，ネットワーク上で通信をモニタして追跡するという行為が通信の秘密に関する法律に抵触する可能性もあり，導入には法的側面からも考慮すべき課題がある．

そこで本論文では，(i) 被害者 PC からボットの追跡，(ii) ボットから指令サーバの追跡を 1 度に実現するホスト型の追跡方式を提案する．これは，通信経路を順にたどる追跡ではなく，攻撃元となった複数の PC やサーバを直接探し出すことで，処理の一括性と，通信経路のホップ数に依存しない追跡を実現するものである．また，通信の当事者の同意によって通信履歴を共有することで，通信の秘密に抵触しない方式となっている．

(i) は，被害者 PC からの申告情報を基に，自身が加害者であることを自己認識するクレームドリブンな感染 PC の追跡方式である．申告情報として，IDS や AV でアラームを検知した被害者 PC のグローバ IP アドレス（以下，IP と略す）もしくは Fully Qualified Domain Name（FQDN）を，被害者 PC からセンタ局へ報告する．そして，各地の PC がセンタ局から被害者の IP/FQDN を受信して，自身の通信履歴を検査する．もし一致する宛先情報が発見された場合には，ボットに感染して踏み台にされていることを自己認識する．(ii) は，自己認識した各地のボット感染 PC のユーザが，自身の通信履歴をセンタ局に報告することで，センタ局側でボットネットを構成する指令サーバ群をあぶり出す連携的な追跡方式である．このとき，ボット感染 PC はボットに関わる不審な通信プロセスの通信履歴のみをセンタ局に報告することで，情報収集の効率化を図る．この不審な通信プロセスの見分け方は，ボットネットの通信パターンとして，ボットと指令サーバとの通信にみられる特徴，ならびに，ボットが攻撃の踏み台にされたときの通信にみられる特徴に注目する．

提案するホスト型追跡システムを Windows XP 端末へ実装して評価を行い，既存の IP 追跡方式よりも過去に遡って長期間の追跡を実現できることを示す．また，実際のボット通信パターンを用いて，インターネット上の指令サーバをあぶり出せる様子を確認する．

なお，本論文で扱うボットは，攻撃の踏み台になることを前提としており，被害者 PC へアクセスした時点で，ボット感染 PC は加害者 PC となる．以後の説明では，ボット感染 PC は必ず加害者 PC になることを想定し，攻撃のタイミングを考慮することなく，両者を同義語として扱うことに注意されたい．

以下，2 章で既存の追跡技術を概観し，3 章で提案するホスト型追跡モデルのアーキテクチャを説明する．4 章で被害者 PC からボットへの追跡手法を，5 章でボットから指令サーバの追跡手法を，それぞれ述べる．6 章で実装システムの評価を行い，提案方式の有効性を検証する．そして最後に 7 章でまとめる．

2. 既存の追跡手法

ここでは，既存の IP 追跡方式とアプリケーション追跡方式について概観する．

2.1 IP 追跡

IP 追跡方式とは，パケット中の IP レイヤの情報を用いて追跡する手法である．代表的な方式として，通過するパケットのハッシュ値を通信経路上の装置に保存しておき，被害者側に届いた攻撃パケットのハッシュ値を順に追跡するハッシュ方式がある[1),2)]．このほか，ルータを通過するパケットをサンプリングして，パケット情報とルータ情報を，ICMP パケットに載せて Destination IP へ送付する ICMP 方式[3)] や，通過するパケットをサンプリングして，ルーティングに影響のないヘッダ領域に，ルータ情報を書き込むパケットマーキング方式がある[4)]．

また，加害者 PC が攻撃の直前に DNS に問い合わせることによって被害者 PC の FQDN に該当する IP を取得することに注目し，DNS 検索のログからボットを特定する DNS ログ追跡方式[5)] や，インターネットとエンドネットワークの接続点において，パケットにネットワーク ID を付与して，受信者側で発信者を特定する出国印型追跡方式[6)] がある．この 2 つの方式は，通信経路をたどることなく直接送信元を特定できるため，効率的な手法といえる．

2.2 アプリケーション追跡

アプリケーション追跡方式とは，パケット中のアプリケーションレイヤの情報を用いて追跡する手法である．たとえば，DNS クエリレスポンスから予想される DNS クエリの固有情報に着目して，DNS リフレクション攻撃の送信元を追跡する[7)]．

また，複数の PC 間を渡り歩く踏み台攻撃の追跡手法も提案されている[8)]．これは，telnet などのターミナルアプリケーションでは，1 文字ごとにパケットが送受信されることに注目して，ある PC の入力パケットと出力パケットに同じ文字が含まれていた場合に，この PC を中継して攻撃が行われていると判定する．これにより同期性のある単一のターミナル通信に限れば，PC 間の追跡は可能である．

2.3 ボットネット追跡における課題

既存の追跡方式のほとんどが，送信者と受信者の通信ペアの追跡にとどまっており，指令と攻撃が非同期的に行われるボットネットの場合，被害者からボットを追跡できても，その先の指令サーバの追跡を行えない．DDoS 攻撃の発信源である多数のボットを追跡する場合，個々の攻撃パケットから 1 つずつボットを追跡することになり，効率面での課題もある．また，多数のプローブを通信経路上に設置する必要があり，プローブの設置コストを要

してしまうという課題がある．さらに，ネットワーク上で通信をモニタして追跡するという行為が通信の秘密に関する法律に抵触する可能性もあり，その導入には法的側面からも考慮すべき課題がある．

3. ホスト型追跡の構成

ボットは感染 PC に直接的な被害を与えないため，ユーザが感染に気付き難いことや，たとえ気づいたとしても加害者になっている意識が薄いことで，感染 PC を放置し続けてしまうという問題がある．その結果，次第に感染 PC の数が増え，指令サーバに制御されるボットネットが構成されていく．

そこで本論文では，(i) 被害者からの申告に基づき自身が加害者であることを自己認識するボット追跡の仕組み，(ii) ボット感染 PC の不審な通信プロセスの宛先 IP/FQDN に注目した指令サーバ追跡の仕組みを提案する．これらは，各ユーザが情報の提供と照会を自主的に行うことで通信の秘密を侵害しないホスト型追跡を実現している．

3.1 構　　成

被害者 PC からボット，ボットから指令サーバを追跡するフレームワークを**図 1** に示す．本論文では，IDS や AV などの攻撃検知ツールが各 PC にインストールされており，外部からの攻撃を検知できるものとする．その際，ボット感染 PC 上の IDS や AV ではボットを検知できない，もしくは検知できた場合でも，ユーザが放置してしまうことを想定している．たとえば DDoS 攻撃の場合，多量の攻撃パケットが届く被害者 PC の IDS やユーザは被害を把握できるものの，加害者 PC 1 台あたりの発信される攻撃パケットが少ないことで，加害者側の IDS やユーザが検知や状態把握を行えない．

被害者 PC は，IDS もしくは AV で攻撃を検知すると，被害者情報をセンタ局に報告する．センタ局は複数の報告をリストとしてとりまとめ，公開する．すべての PC は AV の定義ファイルの更新と同じ要領で，センタ局から被害報告リストを定期的にダウンロードする．そして，自身の通信履歴と照合し，リスト中のいずれかの被害報告の原因に該当する記録が見つかれば，自身がボットに感染した加害者であることを自己認識する．そして加害者 PC は，自身の通信履歴の中から，攻撃記録に該当する時間帯前後の不審な通信プロセスに関する通信記録を抽出し，センタ局に報告する．センタ局では，複数の加害者 PC の不審な通信履歴から，ボットネットの構成を描き出すとともに，多くの加害者 PC と通信を行っている活発な指令サーバを特定する．

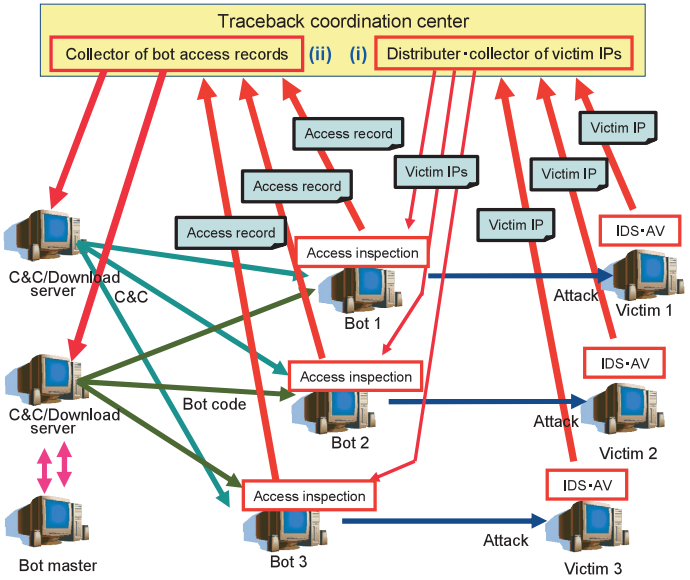


図 1　ボットと指令サーバを追跡するフレームワーク
Fig. 1　Traceback framework against bots and C&C/download servers.

3.2 ホスト型追跡モジュール

図 2 に，ボットと指令サーバの追跡に関わる処理構成を示しておく．処理は，PC 側とセンタ局から構成される．ここで，各 PC は被害者にも加害者にもなりうることから，被害者 PC とボット感染 PC を同じ処理構成とする．各 PC は，自身の送受信パケットを通信履歴として数日間保存しておき，これをボット感染の判定と通信履歴の報告に用いる．

図 2 に示したホスト型追跡モジュール（Host-based tracking module）の処理フローについて，被害者 PC とボット感染 PC の場合に分けて，**図 3** にまとめる．被害者 PC（図 3 右）は，送受信パケットをモニタして，通信履歴として保存するとともに，IDS や AV で自身に対する攻撃を監視する．もし IDS や AV によって攻撃を検知した場合には，その時刻と自身の IP アドレスをセンタ局へ報告する．ボット感染 PC（図 3 左）は，定期的にセンタ局から被害報告リストをダウンロードして，自身の通信履歴を検査する．もし該当する時刻に被害者 PC へのアクセスが確認された場合には，自身が加害者であることを認識し，図 2 に示す疑陽性ボットプロセス検知モジュール（False-positive process inspector）

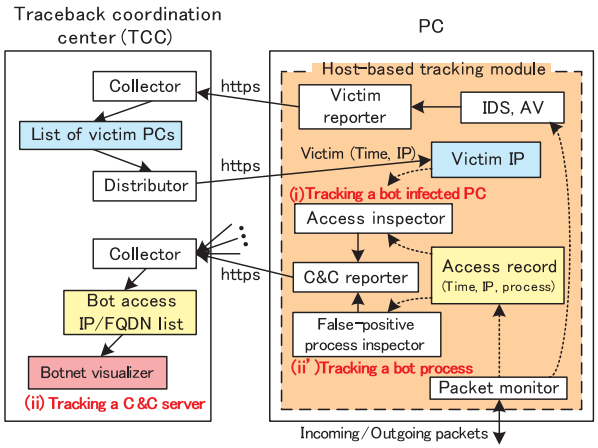


図 2　ホスト型追跡システムの機能構成
Fig. 2　Functions of the host-based traceback system.

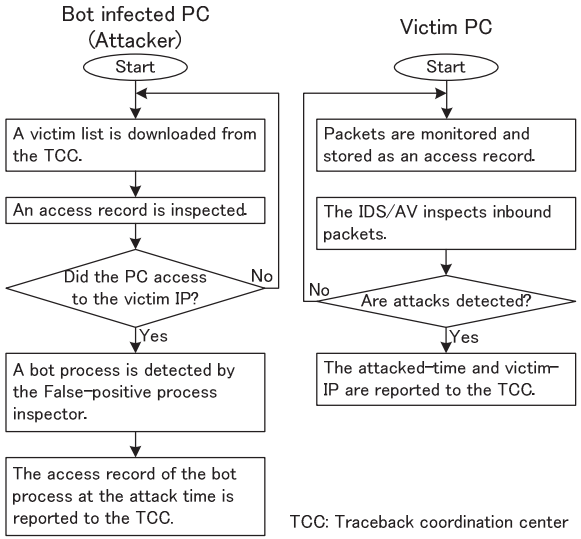


図 3　ホスト型追跡モジュールの処理フロー
Fig. 3　Flow model of host-based tracking module.

によってボットらしき不審なプロセスを特定する．特定手順の詳細は 5.1 節で説明する．そして，該当時刻の不審なプロセスの通信履歴をセンタ局へ報告する．

本方式におけるボット感染 PC の追跡は，加害者側の PC が被害報告リストに基づいて自身の通信履歴を検査する方式であるため，送信元 IP が詐称されていない攻撃だけでなく，詐称された攻撃についても追跡できる．また，ボット感染 PC 側のアドレス情報を用いないため，ブロードバンドルータなどの背後にあるプライベートアドレスを持つボット感染 PC の追跡も行える．なお，ボットネットの多くはボットが指令サーバにアクセスする構成であり，ボット感染 PC から指令サーバの追跡において，指令サーバの IP/FQDN が詐称されていることや，プライベートアドレスであることを想定する必要はない．

4. ボットの追跡

ここでは，被害者 PC がセンタ局に被害を報告し，センタ局が被害報告リストを作成，各 PC が自身の通信履歴と被害報告リストを照合してボットに感染していることを自己認識する，ボット追跡の仕組みを説明する．

4.1 被害者 PC からの報告

被害者 PC は，IDS や AV によって攻撃を受けたことを検知する．報告処理は以下のシナリオに分かれる．

[基本報告シナリオ]

被害者 PC は，攻撃受信時刻と自身の IP/FQDN などをセンタ局に申告する．このとき，攻撃パケットの送信元 IP は報告しない．これは，送信元 IP が詐称されている可能性があること，ブロードバンドルータの背後にいるボットの場合，送信元のローカルアドレスが変化してしまうことが理由である．また，多数のボットから 1 つの被害者 PC への DDoS 攻撃については，被害者 IP/FQDN のみを用いてすべての加害者 PC を一括追跡することで効率化が図れることも理由である．

[DNS リフレクションと Smurf 攻撃の報告シナリオ]

DNS リフレクションや Smurf 攻撃の場合，増幅の踏み台に利用された DNS サーバやネットワークが，攻撃パケットの送信元 IP として現れる．いい換えると，ボットから発信される DNS リフレクションや Smurf 攻撃パケットの宛先 IP が，踏み台に利用された DNS サーバやネットワークであり，被害者 PC に届いた攻撃パケットの送信元 IP である．そこで，被害者 PC の IDS や AV が，DNS リフレクションや Smurf 攻撃を検知した場合には，攻撃パケットの送信元 IP をセンタ局に報告する．ちなみに IDS や AV は，DNS クエリをと

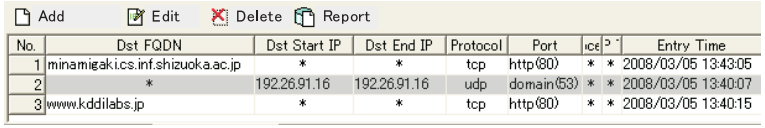


図 4　被害報告の編集用インタフェース
Fig. 4　Interface of victim report editor.

もなわない DNS クエリレスポンスを受けた場合を DNS リフレクション，ICMP Echo をともなわない ICMP Echo Reply を受けた場合を Smurf 攻撃と判定している．

図 4 に，実装した被害者 PC の被害申告インタフェース（IF）を示す．被害者側は，被害報告の申告のための情報を，追加・編集・削除の機能を使って作成する．このとき，時刻情報と IP/FQDN に加えて，プロトコル，Port もあわせて記載することで，ボット感染 PC の通信履歴をより厳密に特定できる．送信の際には，被害報告の送信についての同意を得るためのポップアップを表示する．この送信が追跡要求の発行となる．センタ局との通信には，Firewall によるフィルタリングを回避すること，ならびに，盗聴による情報漏洩を防ぐために，HTTPs を利用した．クライアント側となる被害者 PC は，センタ局の IP を本 IF に設定するのみで，Web クライアントとして動作する．

4.2　被害報告リストの作成

センタ局は，各地の被害者 PC から報告される IP/FQDN を被害報告リストとしてとりまとめ，Web サーバを通じて各地の PC に配信する．

4.3　ボット感染 PC 上での自己監査

センタ局から配信される被害報告リストを定期的にダウンロードして，これを自身の通信履歴と照合し，被害報告の原因となった通信記録が検知された場合には，これを警告する検知モジュールを実装した．その IF を**図 5** に示す．上部に被害報告リストに一致するパケットの一覧を，下部に各パケットの詳細情報を表示している．被害者 PC からの申告情報が，自身の通信履歴の中から見つかることで，加害者は自分の PC が攻撃に加担したという事実を認識する．このように，ボット感染検知をクレームドリブンに実現することで，加害者にボットの駆除を促す効果が期待される．

図 5 では，センタ局から配信される被害報告リストに “192.26.91.14” が含まれており，これを加害者 PC の通信履歴と照合した結果，加害者 PC が “192.26.91.14” へアクセスしていたことが判明し，その通信履歴がアラームとして表示された様子を示している．

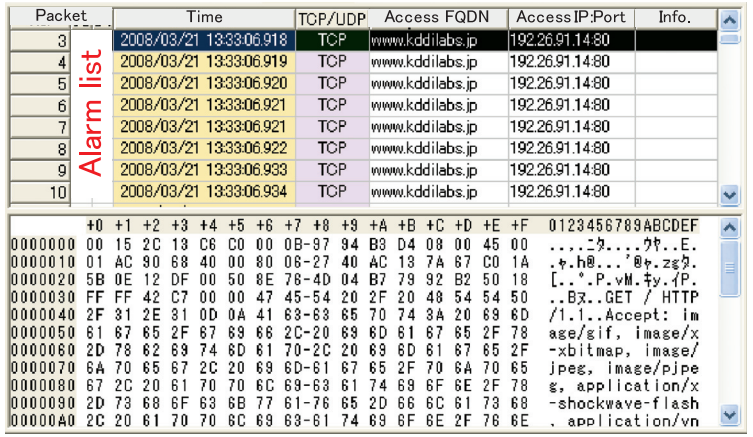


図 5　通信履歴との照合による警告 IF
Fig. 5　Communication inspection and alarm IF.

5.　指令サーバの追跡

ここでは，加害者 PC いわゆるボット感染 PC 上の不審な通信プロセスを特定して，この通信履歴をセンタ局に報告することで，インターネット上の活発な指令サーバを追跡する手法について説明する．

5.1　ボットプロセスの特定

ボット感染 PC は，攻撃の前後に指令サーバと通信している可能性が高い．そこで，ボット感染 PC の通信履歴をセンタ局において収集し，複数のボット感染 PC に共通する通信先を抽出することによって，活発な指令サーバを追跡することができる．このとき，ボット感染 PC はボットに関わる不審な通信プロセスの通信履歴のみをセンタ局に報告することで，情報収集の効率化を図る．

ボット感染 PC は，センタ局に報告する通信履歴を絞り込むために，起動中の通信プロセスの中からボットプロセスを特定する必要がある．PC 上で起動中のボットプロセスを特定することは，ボットの駆除にも寄与する．**表 1** に，ボットプロセスを特定するために注目すべき特徴をまとめる．表 1 の各項目はボットとしての通信挙動に現れる特徴を示しており，ボットの多くがこのうちの 1 つ以上の特徴を持っている．以下にそれぞれの特徴を概説する．

表 1　ボット通信パケットの特徴
Table 1　Packet features of bot communication.

	Protocol	Packet features	Information
1	HTTP (80/TCP)	GET *.exe POST *.exe	Bot code downloading
2	IRC (High port)	User, Nick, Ping, Pong, Mode, Join	Control messages
3	Attack ports (135-139,445,1433-1434)	–	Infection activities
4	DNS (53/UDP)	No such name Server failure	C&C address retrieval
5	SMTP (25/TCP)	Too many SMTP sessions Recipientaddress rejected User unknown Message refused	Spam e-mail sending

1. ボットは，指令サーバから新たなボットコードを取得する．この通信として HTTP サービスを利用する場合が多く，HTTP ヘッダに “GET *.exe” や “POST *.exe” が現れる．ここで “*” には，様々な文字列が入る．
2. 指令サーバは IRC サービスを利用して，ボットを制御する．IRC サービスを見分けるために，Port 1024 以上の High Port が利用されていること，そのパケットに “User”，“Nick”，“Ping”，“Pong”，“Mode”，“Join” のコマンドが現れることに注目する．なお，本論文では，ボット感染 PC のユーザは，自身による IRC 通信と見知らぬ IRC 通信を区別できるものとし，指令サーバとの IRC 通信を判別する方法については本論文の対象外とする．
3. ボットは次の感染先を探すために，脆弱性を持つ Port へのスキャンを行う．感染に悪用されがちな Port として，“135–139”，“445”，“1433–1434” があげられる．なお，正規の通信の中でこれらの Port へ向かうパケットは，LAN 上では頻繁に送受信されるものの，インターネット上に送信されることはほとんどない．よって，この特徴を考える場合には，インターネット上の IP と Port の組合せに注目する．
4. 指令サーバは，ユーザによる駆除や，電源が落とされるたびに頻繁に入れ替わる．このため，ボットが指令サーバの名前解決を行うと，たびたび DNS エラーレスポンス “No such name” が返信される．また，ISP から一時的な名前を割り当てられた指令サーバも多く，FQDN に各国の ISP が割り当てる特殊な文字列を含むことで，たびたび DNS エラーレスポンス “Server failure” が返信される．
5. スパムメール送信型ボットの場合，ボットとメールサーバの間に，SMTP 通信セッションが確立されるが，この通信セッション数が大きな場合，“Too many SMTP sessions” エラーが返信される．また，メールの宛先アドレスをでたらめに決める場合も多く，“Recipient address rejected” エラーもしくは “User unknown” エラーが返信される．さらに，送信元ドメイン認証を行っているメールサーバに接続した場合，“Message refused” エラーが返信される．

ここで，ボットの活動は通常潜在的・散発的なものが多く，正規のプロセスの挙動の中にボットの挙動が埋もれてしまい，表 1 にあげた特徴だけを利用して精度良くボットを検出することはできないことに注意されたい．本論文における不審な通信プロセスの特定は，4.3 節の自己監査の結果，自身がボットに感染していることが明白になった後に，被害報告リストに記されている攻撃時刻から自身の通信履歴を絞り込み，さらに表 1 の特徴に注目することで，既存の IDS や AV で達成できなかったボット検知を実現する技術である．

我々は，これらの特徴を利用してボットであると疑われるプロセスのみを抽出する疑陽性ボットプロセス検知モジュールを Windows XP に実装した．これは，以下に示す API で通信プロセスをフックし，プロセス名，宛先 IP:Port，パケット情報を取得して，表 1 とのマッチングを行う．

- TCP:AllocateAndGetTcpExTableFromStack()
- UDP:AllocateAndGetUdpExTableFromStack()

擬陽性プロセスは，表 1 の特徴を 1 つでも有するプロセスである．判定基準が広く設定されているために一部の正規プロセスも誤ってボットプロセスとして抽出される False Positive を含むことに注意されたい．

宛先 FQDN の取得は，DNS サーバによる名前解決に関わる DNS クエリレスポンスパケットに注目する．この様子を**図 6** に示す．図 6 では，DNS クエリレスポンス中に，宛先 IP と FQDN が含まれていることから，これを管理しておく．その後，発信されるパケットの宛先 IP に該当する FQDN を対応付ける．

図 7 に，本検知モジュールの IF の様子を示す．図 7 は，実際にボットに感染した PC 上で通信パケットをモニタリングしており，表 1 に該当した通信パケットを抽出して，時刻，プロトコル，宛先 IP:Port，宛先 FQDN，プロセス名が表示されている．“*exploere.exe*” は，初期のボットコードであり，HTTP port 80/TCP を通じて新たなコード “*winIogon.exe*” の取得と，IRC port 7000/TCP を通じて指令サーバからの制御を受けている．“*winIogon.exe*” は，

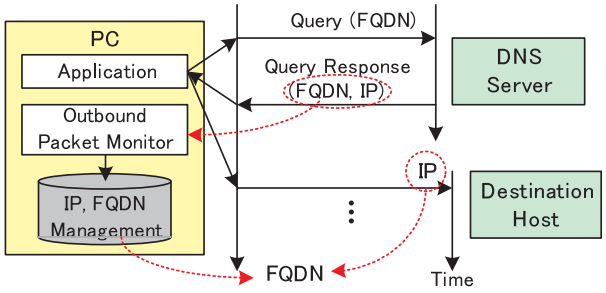


図 6　宛先 FQDN の取得
Fig. 6　Management of the destination FQDN.

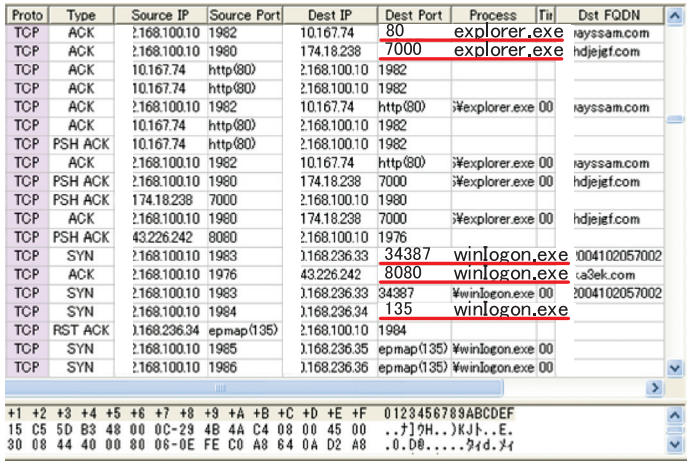


図 7　実装した疑陽性ボットプロセス検知モジュール
Fig. 7　Implementation of the bot process detector.

IRC port 34387/TCP，IRC port 8080/TCP を通じて指令を受け，脆弱 port 135/TCP に向けて攻撃パケットを発信している．

なお，ボットの通信履歴には概して表 1 のような特徴が含まれるが，その逆は真ではなく，このような特徴を有する通信のすべてがボットによるものではない．よって，本検知モジュールを利用して，各 PC で平時から通信履歴を残しておき，被害者からの報告によって自身が加害者になっていることが判明した場合に，その攻撃時刻の前後に，ボットであると

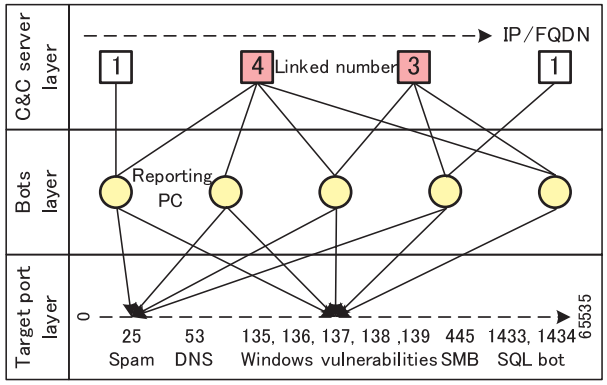


図 8　ボットネットの視覚化
Fig. 8　Botnet visualizer.

疑われるプロセスによる通信履歴が見つかれば，これをボットによるものであると確定し，該当する通信履歴のみをセンタ局に報告する．こうした実装によって，加害者判定をより確実なものにできること，センタ局への通信履歴の送信を効率的に行えることを実現している．

5.2　ボットネットの視覚化と指令サーバの追跡

センタ局は，それぞれのボット感染 PC から報告された不審な通信履歴を集約し，指令サーバの IP/FQDN，ボットの IP，攻撃先 Port を視覚化する．この作業によって，ボットネットの構成を把握するとともに，活発な指令サーバを特定できる．これは，共通する指令サーバをキーに，指令サーバとボットと攻撃先 Port の間をリンク付けることで，ボットネットをグループ化してあぶり出す手法である．この様子を**図 8** に示す．攻撃先 Port 層には Ports 25，53，135–139，445，1433–1434 TCP/UDP に該当する Port 情報を，ボット層には報告のあったボット感染 PC の IP アドレスを，指令サーバ層には攻撃パケット以外の宛先 IP/FQDN を表示する．

視覚化された指令サーバ層の IP/FQDN には，*.exe ファイルを配布する OS のアップデートサイトや正規の IRC サーバの IP/FQDN も含まれる可能性がある．そこで，センタ局の運用者は，Whois 検索など通じてよく知られた IP/FQDN の排除を行い，残った IP/FQDN を指令サーバと見なす．

ここで指令サーバを表す四角印の内部に示されている数値は，その指令サーバにアクセスしてきたボットの数を表しており，ボットネットにおける指令サーバとしての重要度を意味

する．インターネット上には，多数の指令サーバが存在するが，この重要度の高い指令サーバから順に駆除することが望まれる．

6. 評価および考察

ここでは，PC 内部で管理する通信履歴のサイズ，センタ局に報告する通信履歴のサイズ，ボットプロセスの検知特性について評価する．

6.1 PC 上で管理する通信履歴のサイズ

提案したホスト型追跡システムの場合，各 PC が自身の通信履歴を HDD に保存しておく必要がある．そこで，研究者，システム管理者，事務担当者の 3 名の通常業務において，本システムで収集される通信履歴のデータサイズについて調査した．**表 2** に 5 営業日観測したときの，1 日あたりの Pcap ファイルとして保存される平均的な通信履歴のサイズを示す．10 Gbps のバックボーンをモニタする既存のネットワーク型追跡システムの場合，記憶容量の問題からたかだか 100 秒程度のデータしか保存できないのに対して，本ホスト型追跡システムの場合は各 PC が保存すべき最大のデータ量は 1 日あたり 20〜600 MByte 程度であり，昨今の PC であれば 2，3 日程度の通信履歴であれば問題なく保存できる．攻撃の判定，センタ局への申告，センタ局からの被害報告リストのダウンロード，通信履歴の照合などの処理は，通信履歴が消えるまでに行えばよく，たとえば 1 日に 1 度，自身がボットに感染して加害者になっていることを検証するツールとして利用できる．

6.2 報告する通信履歴のサイズ

ボット感染 PC が自身の通信履歴をセンタ局に報告するとき，そのデータサイズがネットワーク負荷およびセンタ局の HDD 容量に影響を及ぼす．ここでは，ボット感染 PC からセンタ局に報告するデータ種別とサイズの関係を評価する．

そこで，ハニーポット[9] を用いて事前に 50 種類のボットコードを収集した．実装したホスト型追跡システムを仮想マシンモニタ上の PC にインストールして，収集された 50 種類のボットコードを個々に感染させたときの通信に関して調べた．なお，この 50 種類のボットコードには，仮想マシン上で活動しないものは含まれていない．また，外部への攻撃パケットについては途中の Firewall で棄却している．

表 2　1 日の通信で取得される平均 Pcap サイズ
Table 2　Mean sizes of the Pcap access record.

	Researcher	System manager	Secretary
Size/day	21.7 Mbyte	594.6 Mbyte	121.9 Mbyte

表 3 に，ボットプロセスが 10 分間に送受信した Pcap ファイルサイズ，パケット数，指令サーバの IP 数，攻撃先 IP 数を示す．ここで攻撃先 IP 数については，攻撃先 Port 情報も含まれている．ボット感染 PC は，初めの 10 分間で平均 0.56 MByte，2,450 パケットを送受信することが分かった．また，指令サーバと被害者 PC を合わせた宛先 IP 数は平均 60 種類となり，報告する通信履歴として指令サーバと被害者 PC の IP:Port 情報に絞ることで，ネットワーク負荷を与えない報告サイズとなることが分かった．

6.3 ボットプロセスの特定

ここでは，表 1 に示したパケットの特徴に注目した PC 上での疑陽性ボットプロセスの特定成功率について評価を行う．そこで，前節の評価の際に用いた 50 種類のボットコードを仮想マシンモニタ上の PC にて個々に感染させたときに，表 1 の特徴が検知される確率を調査した．その結果を**表 4** に示す．表 4 より，ボットの 78%が HTTP サービスを利用して新たなコードを取得し，72%が IRC 通信サービスを経由して指令サーバから制御を受け，66%が攻撃パケットを発信し，32%が DNS サーバやメールサーバからエラーレスポンスを受信していた．

図 9 に，各ボットが持ち合わせる表 1 の特徴の重複度の分布を示す．1 つの特徴にも該当しないボットはなく，1 つのみに該当するものが 3 種類，2 つ以上の特徴を持つものが 47 種類あることが分かる．以上より，表 1 の特徴を用いて PC 上で起動している疑陽性のボットプロセスを抽出することによって，ボットプロセスに関与する通信履歴を漏らすことなく報告できることが確認された．

6.4 ボットネットの視覚化についての考察

5 章で設計したボットネットの構成を視覚化するツールを**図 10** のとおり実装した．視覚化の効果を把握するために，*Cyber Clean Center* (*CCC*) *DATAset 2008* の攻撃通信データ[10),11)] を，ボット感染 PC から報告される通信履歴と見なして，本ツールを稼動させた．

表 3　通信履歴の報告サイズ
Table 3　Report sizes of the access record.

10 minutes bot access record	Mean size
Pcap file (Attack and C&C packets)	0.56 Mbyte 2,450 packets
IP addresses of C&C servers	5.8 kinds
IP addresses of victim PCs	54.4 kinds

表 4　疑陽性プロセス中のボットプロセスの特定率
Table 4　Bot process tracking ratio.

	Packet features	Tracking
1	HTTP (GET *.exe, POST *.exe)	78% (39/50)
2	IRC (User, Nick, Ping, Pong, Mode, Join)	72% (36/50)
3	Attack ports (Port 135-139,445,1433-1434)	66% (33/50)
4	DNS (DNS error responses)	32% (16/50)
5	SMTP (SMTP error responses)	32% (16/50)

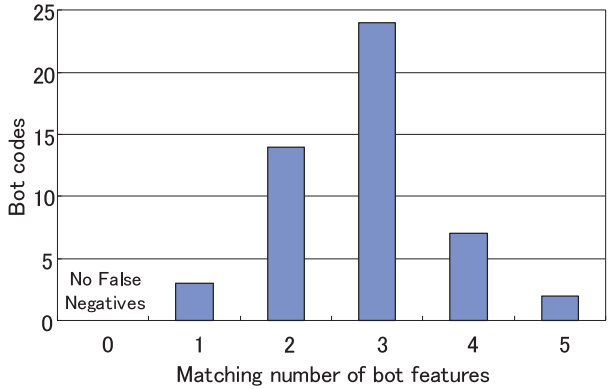


図 9　ボット通信の特徴重複度
Fig. 9　Matching number of bot features.

図 10 は，2008 年 4 月に観測されたある 1 つのボットネットを描画した様子を示している．図 10 より，4 つのボットが 7 つの共通する指令サーバへアクセスしていることが分かる．この 7 つの指令サーバは，ボットコードの配信サーバであり，同じコードを取得したボットは，同じ攻撃先 Port 25，53，135 に向けた通信を行っている．センタ局は，こうしたボットコードの配信に関わる活発な指令サーバの駆除を，該当するドメインのサーバ管理者に依頼することになる．

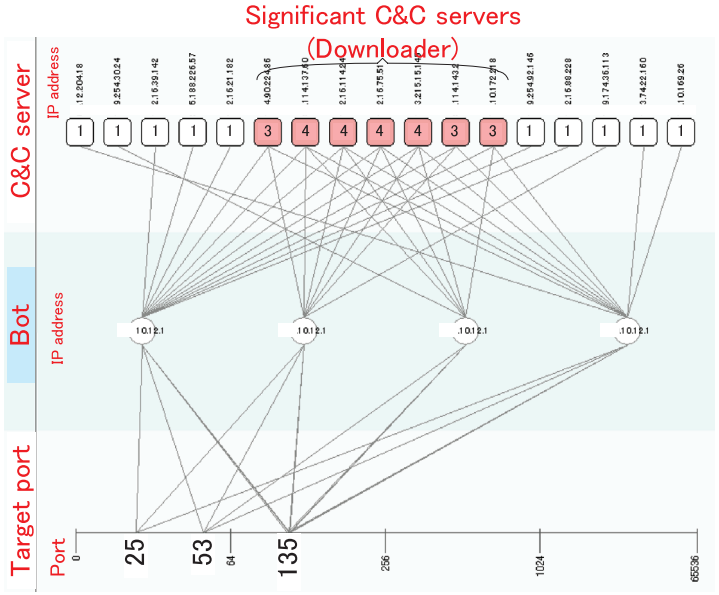


図 10　ボットネット構成の視覚化——CCC Dataset 2008 を用いたボットネットの構成
Fig. 10　Implementation of Botnet visualizer – Botnet topology using CCC Dataset 2008.

6.5　運用についての考察および今後の課題

本システムは，被害を受けた PC 側で攻撃種別を考慮しながら適宜 IP/FQDN をセンタ局に報告する必要がある．このため，一般ユーザに代わりネットワーク管理者が本システムの運用を代表して，被害報告を行うこともある．

本システムにおける被害報告リストは，IDS や AV に該当するパターンファイルの役割を果たしている．しかし，提案方式は被害者を信頼することを前提としているため，偽りの被害情報を報告された場合に，センタ局側で見抜くことは難しい．このため IDS や AV の攻撃検知ログと合わせて報告を受け付けるなど，報告情報の信頼度を高める手法について，検討の必要がある．

本方式の応用として，指令サーバの通信履歴も収集することで，指令サーバ間のつながりや，ボットマスタまでの追跡も期待される．ただし，実際の追跡には，複数の指令サーバの通信履歴が必要なことから，この追跡の工夫については，今後の課題とする．

7. おわりに

本論文では，(i) 被害者 PC からボット感染 PC への追跡，(ii) ボット感染 PC からインターネット上の指令サーバへの追跡を 1 度に実現するホスト型の追跡方式を提案した．ボット感染 PC の追跡は，被害者からの報告に基づき自身がボットに感染していることに気づく，クレームドリブンな追跡方式であり，ボットの駆除に向けた対処や，指令サーバ追跡のための情報提供を促進する効果が期待される．また，指令サーバの追跡は，ボット感染 PC の通信履歴を統合することで，活発な指令サーバを特定できる連携型の追跡方式である．これらは，異なる通信アプリケーションを用いて制御するボットネットを追跡できる手法であり，アプリケーション間追跡を実現している．こうしたエンド PC が自主的に連携する仕組みの場合，追跡のための専用装置をネットワーク上に設置する必要もなく，普及までのコスト的な敷居が低い．

なお，提案した追跡技術にはボットや指令サーバを検知する効果もあるが，追跡は各 PC が相互連携するフレームワークを指しているのに対して，検知技術は PC が個別に対処する機能ととらえて説明した．従来からの検知技術との差異として，既存のネットワーク型 IDS [12] による検知の場合，中継回線を個々の判断でモニタするという行為が通信の秘密に抵触する可能性があるのに対して，本論文の追跡は，加害者 PC が連携する枠組みで自主的に通信履歴を開示するために通信の秘密に抵触しないという特徴を持つ．また，既存のホスト型 AV [13] による検知の場合，悪意のコードを PC 単位で検知するにとどまるのに対して，本論文の追跡は，被害者 PC と加害者 PC が連携することで，多数のボットや指令サーバを 1 度に列挙できる技術である．これらにより，ボット感染 PC が放置され続ける問題の解決と，指令サーバの追跡によりボットネットの不活性化が期待される．

謝辞　本研究は，独立行政法人情報通信研究機構（NICT）の委託研究「インターネットにおけるトレースバック技術に関する研究開発」の一環として行われた．ここに深謝する．

参考文献

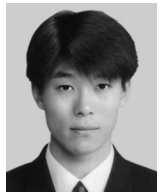
1) Strayer, W.T., Jones, C.E., Tchakountio, F., Snoeren, A.C., Schwartz, B., Clements, R.C., Condell, M. and Partridge, C.: Traceback of single IP packets using SPIE, *Proc. DARPA Information Survivability Conference and Exposition*, Vol.2, pp.266–270 (Apr. 2003).
2) Snoeren, A.C., Partridge, C., Sanchez, L.A., Jones, C.E., Tchakountio, F., Kent, S.T. and Strayer, W.T.: Hash-Based IP Traceback, *Proc. SIGCOMM '01* (Aug. 2001).
3) Bellovin, S., Leech, M. and Taylor, T.: ICMP Traceback Messages, IETF, Internet Draft, draft-ietf-itrace-04.txt (Aug. 2003).
4) Song, D. and Perrig, A.: Advanced and Authenticated Marking Schemes for IP Traceback, *Proc. IEEE Inforcom*, Vol.2, pp.878–886 (Apr. 2001).
5) 竹森敬祐，藤長昌彦，西垣正勝：DNS 記録に注目した詐称 IP 追跡，情報処理学会研究報告，CSEC-40, pp.61–66 (2008).
6) 潘　博文，佐々木良一：IP トレースバックの為の出国印方式の提案，情報処理，CSS 2006, pp.155–160 (2006).
7) 井澤志充，大島龍之介，国峯泰裕：アプリケーションレイヤ由来の情報を用いたトレースバック手法に関する研究，情報処理学会研究報告，CSEC-40, pp.225–260 (2008).
8) 竹尾大輔，渡辺　晃：渡り歩き検出方式の検討，情報処理，CSS 2004, pp.103–108 (2004).
9) Nepenthes. http://nepenthes.mwcollect.org/
10) Cyber Clean Center (CCC). https://www.ccc.go.jp/en_index.html
11) 竹森敬祐，磯原隆将，三宅　優，西垣正勝：ボットおよびボットコードセットの耐性解析，IPSJ, MWS 2008, M2-5 (2008).
12) Goebel, J. and Holz, T.: Rishi: Identify Bot Contaminated Hosts by IRC Nickname Evaluation, *USENIX, HotBots'07, Detection, Response and Analysis* (Apr. 2007).
13) Chiang, K. and Lloyd, L.: A Case Study of the Rustock Rootkit and Spam Bot, *USENIX, HotBots'07, Case Study* (Apr. 2007).

(平成 20 年 12 月 24 日受付)
(平成 21 年 6 月 4 日採録)

推薦文

本論文は，ボットの被害者 PC からの自己申告を基に，被害者 PC 自身が加害者 PC であることを自己申告するクレームドリブンな感染 PC 探索方式と，各加害者 PC の通信履歴をセンターに集め，共通する情報から指令サーバを抽出する方式とを提案している．これにより，これまで単一プロトコルを想定した追跡では困難だった指令者 PC/指令者サーバを特定できるようになった．本提案方式を実装し，その有効性を示している．提案方式は，新たなボット対策として有用性が高いものであり，論文誌の推薦論文としてふさわしい．

（マルチメディア，分散，協調とモバイル（DICOMO2008）シンポジウム
プログラム委員長　串間和彦）



竹森　敬祐（正会員）

1994 年慶應義塾大学理工学部電気工学科卒業．1996 年同大学院修士課程修了．同年 KDD 入社．2004 年慶應義塾大学大学院博士課程修了．現在，KDDI 研究所に勤務．ネットワークセキュリティ，通信ネットワークに関する研究に従事．博士（工学）．2002 年度電子情報通信学会学術奨励賞，2006 年 Interop Tokyo グランプリ，2007 年 DICOMO 最優秀論文賞および最優秀プレゼンテーション賞各受賞．IEEE，電子情報通信学会各会員．

藤長　昌彦（正会員）

1982 年東京大学工学部電気工学科卒業．1984 年同大学院修士課程修了．同年 KDD 入社．現在，KDDI 研究所に勤務．UNIX 通信，G4FAX，分散処理，インターネット経路制御，トレースバックシステム等の研究・開発に従事．IEEE 会員．

佐山　俊哉（正会員）

1981 年青山学院大学理工学部電気電子工学科卒業，同年 KDD 入社．現在，KDDI 研究所に勤務．ネットワークセキュリティ，携帯電話セキュリティに関する開発に従事．

西垣　正勝（正会員）

1990 年静岡大学工学部光電機械工学科卒業．1992 年同大学院修士課程修了．1995 年同博士課程修了．日本学術振興会特別研究員（PD）を経て，1996 年静岡大学情報学部助手．1999 年同講師，2001 年同助教授．2006 年より同創造科学技術大学院助教授．2007 年より准教授．博士（工学）．情報セキュリティ，ニューラルネットワーク，回路シミュレーション等に関する研究に従事．