ボットネットおよびボットコードセットの耐性解析

竹森 敬祐† 磯原 隆将† 三宅 優† 西垣 正勝‡

†株式会社 KDDI 研究所 〒356-8502 埼玉県ふじみ野市大原 2-1-15

‡静岡大学創造科学技術大学院 〒432-8011 静岡県浜松市中区城北 3-5-1

あらまし ボットネットを制御する指令者は，ボットコード（以下，コードと呼ぶ）および指令の効率的な配布とボットネットの安定運用を図るために，中継サーバの並列化を図っている．また，Anti Virus (AV)の普及が進む中，多数のコードを PC に送り込むこと，一つのコードが指令の受信から攻撃の発信までの機能を持つことで単独で一連の処理を完結できること，駆除を逃れて残ったコードが新たなコードを取得すること，AV 処理を妨害することなど，様々な工夫でボットの確保に努めている．本稿では，ネットワーク上での中継サーバの駆除に対するボットネットの耐性について，(i)中継サーバの並列化と(ii)コードの分散管理に着目して紹介する．また，PC 内のボット駆除に対する耐性について，(iii)PC に送り込まれるコード数，(iv)コードの多機能性，(v) AV 駆除後の残存コードの挙動，(vi) AV 処理の妨害の仕組みについて紹介する．

キーワード ボットネット，中継サーバ，ボットコードセット，視覚化，通信プロセスモニタ

Analysis of Robust Mechanisms into Botnet and Code Set

Keisuke TAKEMORI†, Takamasa ISOHARA†, Yutaka MIYAKE†, and Masakatsu NISHIGAKI‡

†KDDI R&D Laboratories Inc. 2-1-15 Ohara, Fujimino-shi, Saitama, 356-8502, JAPAN

‡Shizuoka University 3-5-1 Jyohoku, Naka-ku, Hamamatsu-shi, Shizuoka, 432-8011, JAPAN

Abstract A botnet is composed of multiple command and control (C&C) servers to distribute bot codes and control commands efficiently. A scheme of parallel C&C servers persists against an antivirus (AV) software. It is hard to remove the all bot codes from an infected PC because of the multiple infection, the multiple functions, the update to new code, and the invalidation of AV functions. In this paper, we investigate into the robust mechanisms of the botnet: (i) the parallel structure of the C&C severs and (ii) the parallel management for the bot code on the Internet. Also, we investigate into the robust mechanisms of the bot code: (iii) the number of infected bot codes, (iv) the variety of bot functions, (v) the update function, and (vi) the anti AV technique.

Keywords Botnet, C&C Server, Bot Code Set, Visualizer, Communication Process Monitor

1. はじめに

ボットネットは，中継サーバを階層構造にすることでコードと指令の配信を効率的に行っている．中には，中継サーバを P2P 構造にすることで，並列度を高めたものもある[1][2]．これにより，コードも複数の中継サーバで分散管理される．こうした仕組みは，ネットワーク上での中継サーバの駆除に対して,ボットネットならびにコードの耐性を高めることになる．また，ボットについても多数のコードを PC に送り込むことで，AV による完全な駆除を難しくしている．このコードの中には，指令の受信から攻撃の発信まで一連の機能を持つもの，駆除されないで残ったコードが新たなコードを取得するもの，AV 処理を妨害するものなどが含まれており,様々な工夫でボットの確保に努めている．ここで,PC に送り込まれる単位をコードセットと呼ぶことにする．こうしたボットネットおよびコードセットの耐性の仕組みを把握することは，今後の対策を検討する上で重要である．

そこで本稿では，ネットワーク上での中継サーバの駆除に対するボットネットの耐性について，(i)中継サーバの並列化と(ii)コードの分散管理に着目して，CCC DATAset 2008 の攻撃通信デ

ータ（以降，CCC2008 攻撃通信データ）と攻撃元データ（以降，CCC2008 攻撃元データ）を用いて解析する．また，PC 内のボット駆除に対する耐性について，(iii)PC に送り込まれるコード数，(iv)コードの多機能性，(v) AV 駆除後の残存コードの挙動，(vi) AV 処理の妨害の仕組みに着目して，CCC2008 攻撃元データを用いて解析する．(i)と(ii)については，ボットネットの中継サーバの並列度を視覚化するツールを実装し[3]，解析を行う．(iii)から(vi)については，我々が開発してきたハニーポット，通信プロセスモニタ[4]，通信パターンの視覚化ツール[5]を用いて，CCC2008 攻撃元データを参考に，コードの収集と解析を行う．

以下，2 章でボットネットに関する基本的な統計結果を紹介する．3 章でネットワーク上でのボットネットの耐性について，4 章で PC 内でのコードの耐性について紹介する．そして最後に，5 章でまとめる．

2. 基本的な統計情報の調査

本章では，CCC2008 攻撃元データからわかる基本的な統計情報について紹介する．調査単位は，中継サーバの IP 変動の影響が小さい 1 日分のデータとし，攻撃元データに含まれる最新の 2008 年 4 月 30 日に着目した．

2.1. 中継サーバ IP の分布

コードを配布する中継サーバのインターネット上での分布について調査した．配布数の多い上位 5 件の中継サーバとそれ以外の中継サーバの統計値を図 1 に示す．2008 年 4 月 30 日に配布された全コードのうち，32.8%を配布した中継サーバが 1 つあったが，他は多数の中継サーバが少しずつコードを配布していた．このことより，中継サーバはインターネット上に広く薄く分布しており，ボットネットと配布されるコードが高い耐性を持っていることがわかる．

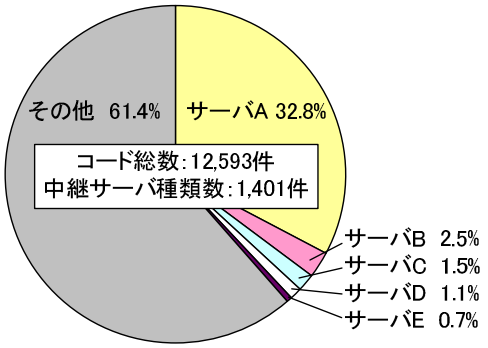

図 1 コード配布を行う中継サーバの分布

2.2. 中継サーバ通信 Port の分布

ボットが中継サーバからコードを取得するときの通信 Port について調査した．利用された通信 Port の上位 5 件とそれ以外の統計値を図 2 に示す．全コードの 38.9%が Port80/TCP で取得されているのに対して，他は様々な Port が利用されている．広く薄く通信 Port が分布していることから，Intrusion Detection System(IDS)による Port の監視やルータによる Port フィルタリングが難しいことがわかる．

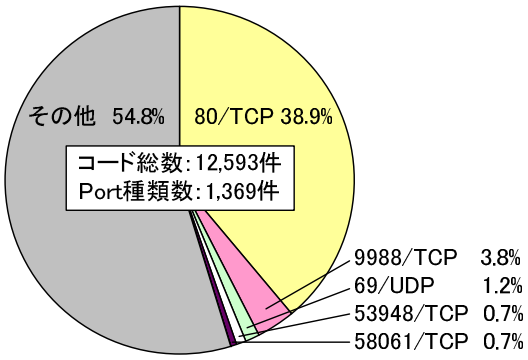

図 2 コード配布に利用される通信 Port の分布

2.3. 未知のコード数

Cyber Clean Center（CCC）[6]が利用している AV が，コードを取得した時点でこれを検知できずに”Unknown”と報告される未知のコード数について調査した．コードの識別にはハッシュ値を利用した．2008 年 4 月 30 日に中継サーバから取得されたコード総数 12,592 件に対して，未知のコード総数は 2,543 件（20.2%）であった．コードの種類数に注目すると，取得された総コード種類数 817 件に対して，未知のコード種類数は 52 件（6.4%）であった．

3. ボットネットの耐性についての解析

ここでは，(i)中継サーバの並列化，(ii)コードの分散管理の様子について，実装したボットネットの視覚化ツールを用いて解析する．

解析単位は，中継サーバの IP の変動の影響が小さい 1 日分のデータとし，CCC2008 攻撃通信データに含まれる最新の 2008 年 4 月 28 日のデータに着目した．

3.1. ボットネットの視覚化ツールの実装

著者らはボットネットの構成を解析するために，中継サーバ層，ボット層，ターゲット層に分割して相互にリンク付ける視覚化ツールを提案してきた[3]．これは，ボットネットは指令者ごとに独立して運用されていることを想定して，

ボットから発信される中継サーバへのパケットについて，1 つでも Destination IP が一致する場合を，同じボットネットに属するボットと考え，グループ化を行っている．今回これを実装し，CCC2008 攻撃通信データの中から抽出したある 1 つのボットネットを視覚化した様子を図 3 に示す．○印はボットを，□印は中継サーバを表している．□内の数値は，アクセスしたボットの数であり，指令サーバとしての重要度を表している．尚，本視覚化は中継サーバ層の並列度とそのボットネットが攻撃する Port を明らかにすることを目的に，階層構造型と P2P 構造型を考慮していない．

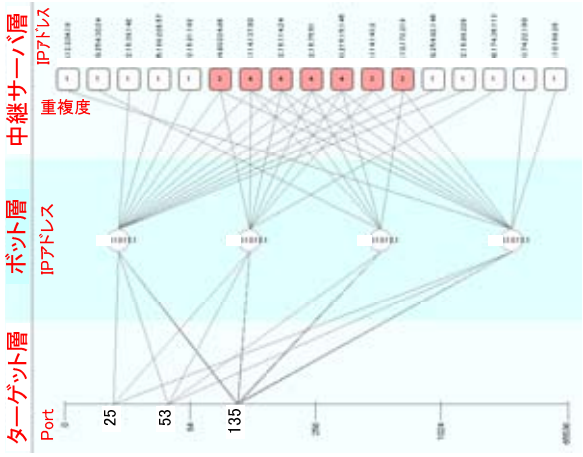

図 3. 実装したボットネット構成の視覚化ツール

図 3 の視覚化は，具体的に次の手順で行う．

Step 1) NTP サーバや LAN などの正常な宛先 IP を予め与えておき，これらを除外する．

Step 2) ボットにみられる特徴的なパケット[3]に注目して，この Source IP をボット層に描画する．ただし，ハニーポットを用いて収集された CCC DATAset 2008 の Pcap ファイルの場合には，ハニーポットが初期化されるごとに，Source IP を独立なものとして扱う．

Step 3) ボットから発信される通信のうち，攻撃拡散に利用される Port25，53，135-139，445 などを，ターゲット層に描画する．

Step 4) 上記以外の Destination IP を中継サーバとみなして，中継サーバ層に描画する．

Step 5) 中継サーバ層，ボット層，ターゲット層を通信履歴から相互にリンク付けする．特に中継サーバ層については，ボット層からのリンク数をカウントする．

3.2. 中継サーバの並列化(i)

図 3 に，対象とする CCC2008 攻撃通信データの中から，1 つのボットネットを構成する通信データを抽出し，中継サーバ層，ボット層，ターゲット層の関係を視覚化した様子を示した．□の数が多いほど，中継サーバの並列度が高く，ボットネットとしてネットワーク上での耐性を持っていることになる．図 3 の上段には，合計 17 台の中継サーバが描かれており，特に上段中央には，3 つ以上のボットからアクセスされている活発な中継サーバが描かれている．これより，17 重の中継サーバの冗長化と，7 重の活発な配信が行われていることがわかる．尚，図 3 は比較的小規模な指令サーバ群であることに注意されたい．

3.3. コード管理の並列化(ii)

コードを複数の中継サーバで管理・配布することで，中継サーバが駆除された場合でも，ネットワーク上でコードの生き残りを図ることができる．ここでは，CCC2008 攻撃元データに記録されているコードのハッシュ値を参考に，各々のコードがいくつの中継サーバから配布されているかを調査した．図 4 に，1 つのコードの配布に関った中継サーバ数の分布を示す．74.3%のコードが，1 台の中継サーバから配布されているのに対して，2～5 台，6～10 台，11～15 台，16～20 台，20 台以上のサーバから配布されるコードがそれぞれ 19.5%，3.4%，1.2%，0.4%，1.2%であった．1 つのコードを配布する中継サーバ数は，平均 2.1 台で，最大 69 台であった．CCC2008 攻撃元データがボットネットの全てを監視できているとすると，25.7%のコードが分散管理されていることになる．これらのコードは，配布を行っている 1 つの中継サーバを停止させただけでは，ネットワーク上から駆除できず，耐性の高さをうかがえる．

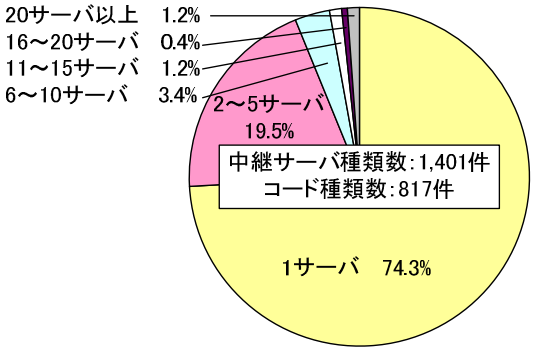

図 4. 1 つのコード配布に関った中継ｻｰﾊﾞ数の分布

4. コードセットの耐性についての解析

ここでは，(iii)送り込まれるコード数，(iv)コードの多機能性，(v) 残存コードの挙動，(vi) AV 処理の妨害について紹介する．

4.1. 解析対象のデータ

CCC2008 攻撃元データは，時刻，中継サーバの IP，ダウンロードされたコード名とハッシュ値が記されたログであり，1 台のハニーポットが複数のコードを取得している様子がうかがえる．ここで，著者らが開発したハニーポットでも，同じ中継サーバ，コード名，ハッシュ値が観測されている．そこで本章では，著者らのハニーポットで収集したコードのうち，CCC2008 攻撃元データから特定された 8 種類のコードに注目して解析を進める．

ボットに感染すると，中継サーバから複数のコードを取得し，HDD にそのまま保存する場合，複数のコードに変態して保存する場合，元からあるファイルに感染する場合，レジストリを改変する場合などがある．表 2 に，1 つのコードに感染して 10 分後の，追加・変更されたコードと，追加・変更・削除された設定状態を示す．この判定には Tripwire[7]を用いている．

表 2. 感染 10 分後のコードと設定の状況

セット	コード		設定の追加･変更･削除	
	追加	変更	レジストリ	hosts
1	4 (exe)	527(exe,htm,scr)	305	有(127型)
2	3 (exe)	422 (exe,scr)	311	有(255型)
3	107(exe)	106 (htm)	15	無
4	5 (exe)	0	3	有(255型)
5	6 (exe)	0	3	有(127型)
6	6 (dll,exe)	3 (ini,sys)	115	無
7	5 (exe)	0	4	有(127型)
8	4 (dll,exe)	1 (ini)	7	有(255型)

4.2. PC に送り込まれるコード数(iii)

表 2 のコードセット 1 と 2 では，数個の exe ファイルが追加され，既存の数百個の exe,htm,scr ファイルに感染し，多数のレジストリが改変された．コードセット 3 では，取得された数個のコードが，ファイル名を変化させながら多数のファイルを作成して HDD に追加し，既存の 106 個の htm ファイルにも感染した．コードセット 4 から 8 では，数個の exe ファイルが追加され，レジストリの改変も行われた．

図 5 にコードセット 4 について，Tripwire による追加・変更の様子と AV による駆除の様子を示す．追加された 5 つの exe ファイルのうち，コードを取得した当日の AV では 3 つしか駆除できず，2 つの exe ファイルが未知のコードとして PC 内に残っていることがわかる．

こうした，未知のコードを含む多数のコードを HDD に保存する仕組み，多数の設定を改変する仕組みで，AV の駆除から逃れるコードが現われる．未知のコードが 20.2%含まれる統計より，AV で 4 つのコードが検知された PC には，1 つの未知のコードが潜んでいることになる．

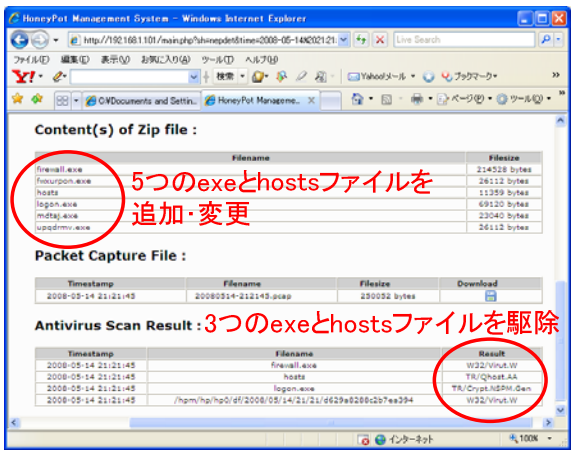

図 5 コードセットの追加･変更と AV 駆除の様子

4.3. コードの多機能性(iv)

表 2 のコードセット 1 について，追加・変更されたコードの一例を図 6 に示す．これらの殆どが起動されることなく，HDD に保存されたままである．しかし，cmd.exe や explorer.exe，bedaula.htm，などは，Windows PC が元々持つプログラムに感染しており，ユーザ操作の中で起動が期待されるトロイの木馬であることがわかる．このコードセットに感染すると，自動起動されたボットを駆除できたとしても，ユーザ操作により再び感染することになる．尚，htm ファイルの多くは，ActiveX の起動を制御する行が追加されており，HTML のローカルなヘルプ画面を参照するだけでボットが起動する．

図 6 追加･変更されたコードセット

コードセット 5 では，6 個のコードが追加され，exploere.exe，winamp.exe，winlogon.exe の 3 個のコードが起動して外部の PC と通信を行っている．この様子を，著者らが開発したホスト型の通信プロセスモニタ[4]を用いて視覚化した様子を図 7 に示す．本ツールは，Windows PC から発信されるパケットの Source/Destination IP:Port，該当する通信プロセス名，Fully Quantified Domain Name(FQDN)などをモニタする．図 7 では，explorer.exe が Destination Port 80/TCP と 7000/TCP の通信を行い，winlogon.exe が 34387/TCP，8080/TCP，135/TCP の通信を行っている．これら 3 つのボットプロセスが 10 分間で通信した Destination Port と，正常なアプリケーションを特別な設定や操作を行うことなく利用したときの Destination Port について表 3 にまとめる．ボットである explorer.exe と winlogon.exe のプロセスは，2 つの High Port を含む 4 つの Port を利用しているのに対して，正常なアプリケーションは 1 つもしくは 2 つの Well-known Port を利用している．このようにボットの場合，1 つのプロセスが High Port を含む複数の Port と通信する異常な挙動がモニタされる．また，3 つのボットプロセスとも，80/TCP の通信を行っていることがわかり，1 台のボット感染 PC から発信される 80/TCP の通信は，複数のボットプロセスから発信されるパケットが集合した通信であることがわかる．また図 8 に，著者らが開発してきた通信パターンの視覚化ツール[5]を用いて，図 7 に示したデータを描画した様子を示す．このツールは，32bit の IP アドレスを 8bit ごとに分割して，Source/ Destination IP/Port の合計 10 軸に，通信セッションの接続関係を，通常再生や 10 倍速再生で動画表現する視覚化ツールである．図 8 より，80/TCP で多数の中継サーバからコードや指令を取得して，25/TCP によるスパムメールの送信の試み，135/TCP による攻撃拡散の試みを行っている様子がわかる．尚，挙動解析においては，Outbound 方向の攻撃パケットを棄却しており，攻撃に関する通信シナリオは把握していない．複数のボットプロセスが複数の Destination Port へ個々のタイミングで通信するため，PC の送受信パケットを観測しただけでは，ユーザ操作のタイミングで発生する通信とボットの通信を見分けることはできない．また，起動中のボットプロセス数を推定することもできない．よって，図 7 のように通信プロセスをモニタして，複数の Port を利用する異常なプロセス，未知の IP と通信する異常なプロセスを特定する必要がある．

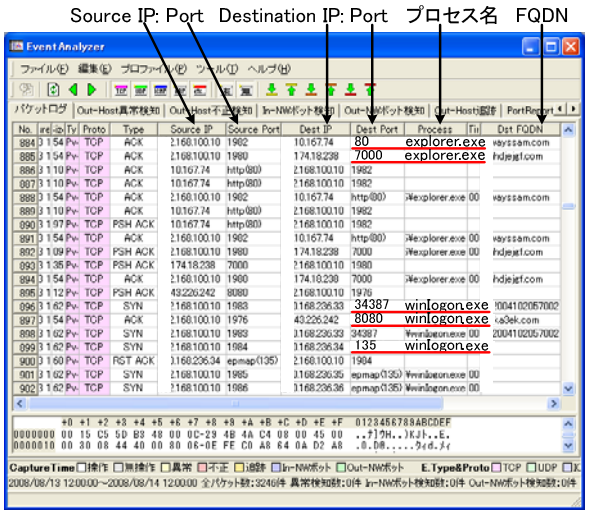

図 7 開発した通信プロセスモニタ

表 3. 正常アプリケーションとボットの通信 Port

アプリケーション	プロセス名	Destination Port
Outlook Express	msimn.exe	25,110/TCP
Internet Exproler	iexproler.exe	80,443/TCP
Acrobat Pro.	AcroRd32.exe	80/TCP
ボット	explorer.exe	25,80,1867,7000/TCP
	winamp.exe	80/TCP
	winlogon.exe	80,135,8080,34387/TCP

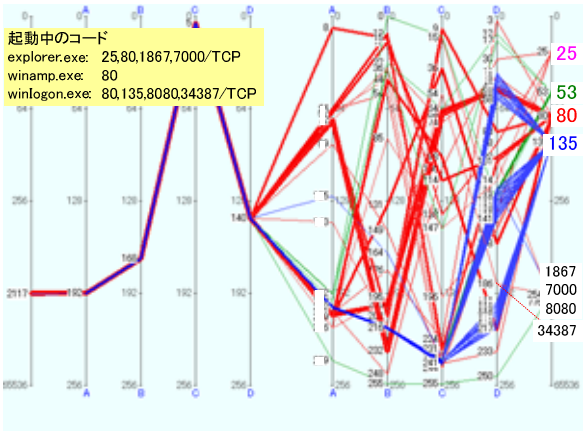

図 8 開発した通信パターンの視覚化ツール

一般的に，数十～数百 KB のサイズのコードが多い中，コードセット 6 では僅か 7KB のコードが追加された．このコードについて挙動解析を行ったところ，80/TCP による指令の取得と，25/TCP によるスパムメールの送信を行なうと

いう，複数の機能を持つことがわかった．尚，今回の 8 つのコードセットでは，指令の受信や攻撃の発信をコード間で分担するような協調処理はみられず，個々のコードで処理が完結していた．よって，一部のコードを駆除することで，コードセットを機能不全にすることはできない．

4.4. 残存コードのその後の挙動(v)

コードセット 7 では，AV で駆除されずに未知のコードとして残った logon.exe がモニタされた．このコードをさらに 10 分間そのまま放置したときの様子を解析した．その結果，135/TCP への攻撃拡散を行いながら，新たに 80/TCP で二つのコードを取得して，これらを起動した．これら 3 つのコードは，同じボットネットに属していながらも，様々な IP:Port と個々に通信を行っていた．上記以外にも， 3 つのコードを HDD に追加していた．このように，AV による駆除を逃れたコードが一つでも残っている場合には，それが新たなコードを取得して，感染状態に戻ってしまう．AV や IDS をインストールしてある PC の場合，残ったコードが既知のコードを自動取得することで，身に覚えの無い AV アラームが次々と出ることになる．

4.5. hosts ファイルの変更(vi)

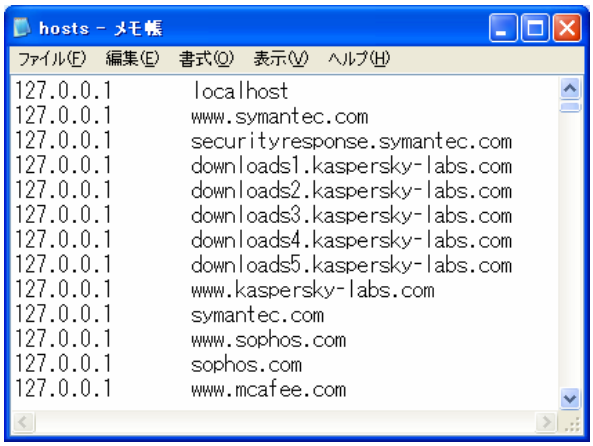

図 9 127 型に改変された hosts ファイル

改ざんされた hosts ファイルの一例を図 9 に示す．各種 AV のパターンファイルの配布サーバのホスト名に対する IP アドレスが全て 127.0.0.1（127 型）と 255.255.255.255（255 型）に改ざんされていた．著者らの調査では，255 型については，日本で販売されている多くの AV が駆除できるものの，127 型については，多くの AV が駆除できないことがわかった．ひとたび 127 型のボットに感染すると，2008 年 8 月現在の AV ではパターンファイルの更新は望めない．もし Windows XP 端末にボット感染が疑われる場合には，以下のフォルダにある hosts ファイルの状態を確認すると良い．

- C:¥WINDOWS¥system32¥drivers¥etc¥hosts

5. おわりに

CCC DATAset 2008 の解析を通じて，多数の中継サーバが並列化されていること，コードを広く分散配置していることが明らかになり，ボットネットの高い耐性とネットワークからの駆除が難しいことがわかった．また，多数のコードを HDD に作成すること，トロイの木馬型が含まれること，AV の駆除から逃れたコードが新たなコードを取得して感染状態に戻ること，hosts ファイルを改変して AV のパターンファイルの更新を妨げることなど，コードセットが持つ様々な PC 内での駆除耐性も明らかになった．

著者らからの提言として，AV は初めにボットをダウンロードするコードへの感染予防には役立つが，コードセットが送り込まれた後では，完全な駆除や修復は望めない．よって，ユーザファイルのバックアップを取った後に，OS の再インストールを推奨する．また，身に覚えのない AV アラームが出るような場合には，通信プロセスをモニタすることと，hosts ファイルの状態を確認することを勧める．

文　献

[1] Ping Wang, Sherri Sparks, and Cliff C. Zou, “An Advanced Hybrid Peer-to-Peer Botnet,” Proc. of HotBots’2007, USENIX, April, 2007.

[2] Julian B. Grizzard, Vikram Sharma, Chris Nunnery, Brent B. Kang, and David Dagon, “Peer-to-Peer Botnets: Overview and Case Study,” Proc. of HotBots’2007, USENIX, April, 2007.

[3] 竹森敬祐，藤長昌彦，佐山俊哉，西垣正勝，“ボット攻撃における加害者 PC および指令サーバの探索”，DICOMO2008，1A-3，2008 年 7 月．

[4] 竹森敬祐，高見知寛，西垣正勝，三宅優，“セキュリティインシデントをトリガとしたボット検知方式:宛先 IP とドメインに注目した不正検知”，CSS2007，3C-3，2007 年 10 月．

[5] 竹森敬祐，三宅優，“Firewall ルールの作成に適した通信パターンの視覚化と攻撃パターン検知”,SCIS2008, 2C1-4, 2008 年 1 月．

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

[7] Tripwire，http://www.tripwire.com/