社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2003－CSEC－23 (4)
2003／12／19

DNSの正引き応答連動型パケットフィルタリングによるワーム増殖抑制

岡本 剛†

本報告では、DNSの正引き応答から得られたIPアドレスへの送信を許可し、それ以外のIPアドレスへの送信をWilliamsonのウイルス・スロットルによってパケットフィルタリングする手法を提案する。実験では、本手法は、ウイルス・スロットルと比べて、送信時の遅延時間とワームの増殖阻止に要する時間を短縮し、ワーム感染時でも、名前解決されたホストであれば通信できることを示した。

Packet filtering using DNS responses against worm propagation

TAKESHI OKAMOTO†

Packet filtering using DNS responses that permits outgoing packets to resolved IP addresses and limits all the other outgoing packets to unresolved IP addresses by the "virus throttle" method has been proposed and simulated. The simulation result has shown that the packet filtering makes delay time and time to stop worm propagation smaller than the virus throttle. Furthermore, the packet filtering enables to connect to only the resolved IP addresses even if a computer is infected with a worm.

1. はじめに

インターネットの拡大とブロードバンド化によって、ワーム☆は、極めて短い期間に広範囲かつ大量に拡がるようになった。

ワームの増殖を抑える方法として、異常検出が注目されている[2)~6)]。Williamsonは、1) ワームの感染時における一定時間内に通信するIPアドレスの個数と、未感染時におけるその個数には大きな差があること、2) ユーザが通信する宛先IPアドレスは、最近、通信したIPアドレスに偏っていることを示した[3)]。Williamsonは、この知見を元に、コンピュータ1台が単位時間あたりに送信できるIPアドレスの個数を制限し、送信を許可されなかったパケットは破棄しないでキューに入れて送信を遅延させる手法（ウイルス・スロットル）を提案した。さらに、翌年には、そのキューの長さが閾値を超えたら、すべての送信を停止する機能を追加し、Linux上で、Nimdaは増殖を始めてから0.25秒、Slammerは0.02秒でその増殖を阻止できることを示した[5)]。

ウイルス・スロットルでは、一定時間内に通信するIPアドレスの個数が多いとき、ワームに感染していなくても遅延が起こる。Williamsonによると、送信されたTCPの全SYNパケット中、2.14%のパケットが遅延し、最大で5秒も遅延する[6)]。また、ワームに感染すると、すべての送信が拒否されるため、たとえ感染を発見しても、セキュリティホール修正プログラムをダウンロードできない。感染したコンピュータしか所有しないユーザにとって、ワームの駆除やセキュリティホールの修復は容易ではない。

そこで本研究では、ウイルス・スロットルにおける送信の遅延時間を減らし、ワームに感染した後でも、ユーザによる通信だけを許可する手法を提案する。本手法は、ユーザによる通信とワームによる通信を識別するために、DNSからの正引き応答を利用する。正引き応答で得られたIPアドレスを許可し、それ以外のIPアドレスへの送信には、ウイルス・スロットルを適用する。実験では、正引き応答を利用することによって、送信時の遅延時間とワームの増殖阻止に要する時間を短縮し、ワーム感染時でも、名前解決されたホストであれば通信できることを示す。

† 神奈川工科大学 工学部 情報ネットワーク工学科
Department of Information Network Engineering, Kanagawa Institute of Technology

☆ 本研究で扱う「ワーム」はStaniford らの定義に従う[1)]。ワームはユーザが実行しなくても増殖できるが、コンピュータウイルスはユーザが実行しなければ増殖できない。

2. 正引き応答を利用したパケットフィルタリング

ユーザはインターネット上のホストと通信するとき、ホスト名で指定することが多い。力武らは、DNS と TCP コネクションに関する挙動の解析を行い、正引き応答の後に、正引き応答で得られたアドレスへの接続要求が続く傾向が強いことを確認した[7]。一方、CodeRed など 2001 年から 2003 年に大規模な感染が確認されたワームは、ホスト名ではなく、IP アドレスをランダムに選んで通信を開始する。つまり、これらのワームは名前解決を行わない（名前解決を行うワームに関しては、4 章の考察で述べる）。

しかし、名前解決が行われなかった通信がワームによる通信とは限らない。名前解決を行わない正当な通信には、DNS サーバやブロードキャストアドレスへの通信がある。この他には、ユーザが IP アドレスを指定したときや、ホームページのリンクが IP アドレスで指定されているときなども名前解決は行われない。名前解決を行わない通信におけるユーザとワームの違いは、一定時間内に送信されるパケットの宛先 IP アドレスの個数である。したがって、これらの通信には、Williamson のウイルス・スロットルが有効であると考えられる。

以上から、本手法は、DNS サーバからの正引き応答をモニタし、名前解決によって得られた IP アドレスへの送信を許可し、それ以外の IP アドレスへの送信にはウイルス・スロットルを適用する。ただし、本手法は、DNS サーバから正引き応答を受信できることを前提としているので、正引き要求に対してウイルス・スロットルを適用しない。

図 1 に、本手法の導入例を示す。ユーザのコンピュータと DNS サーバの間に、本手法を適用したコンピュータを設置したとき、本手法は 1 台のコンピュータで複数のコンピュータに有効である。最も単純な導入例は、ユーザが利用するコンピュータ毎に本手法を適用した場合である。以下では、コンピュータ毎に適用した場合について、詳細な処理の流れを述べる。

2.1 受信パケットの処理

受信処理の対象は、事前に登録された DNS サーバ (ポート 53) からの正引き応答パケット (TCP と UDP の両方) とする。正引き応答を受信したら、正引き応答に含まれる IP アドレスをリスト（リゾルブセット）に追加する。正引き応答に複数の IP アドレスが含まれる場合、最初の 1 つだけを追加する。ただし、リゾルブセットの大きさが、閾値 r 以上のときは追加しない。また、DNS から得られる IP アドレスにはキャッシュ有効期限（TTL）が設定されているため、現在時刻が IP アドレスの TTL を過ぎたら、その IP アドレスを削除する。

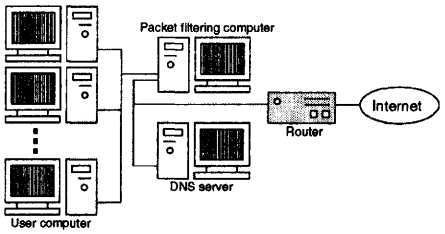

図 1 本手法の導入例
Fig. 1 Example of a packet filtering system.

2.2 送信パケットの処理

フィルタリング対象のパケットは UDP、SYN フラグが 1 の TCP パケット (3 Way Handshake の接続要求パケット) とし、デフォルトポリシーは破棄とする。ルールの適用順序は次の通り。

(1) 宛先 IP アドレスが DNS サーバの IP アドレスで宛先ポートが 53 であれば、送信を許可する
(2) 宛先 IP アドレスが正引き応答から得られた IP アドレスであれば（リゾルブセットに含まれるなら)、送信を許可する
(3) それ以外の IP アドレスであれば、ウイルス・スロットルを適用する

ウイルス・スロットル

ウイルス・スロットルでは、最近に通信した n 個の宛先 IP アドレスのリスト（ワーキングセット）を保持している。送信するパケットの宛先 IP アドレスがワーキングセットに含まれるなら、送信を許可する。また、ワーキングセットに含まれないが、ワーキングセットに空きがあれば（ワーキングセットの大きさが n より小さければ)、送信を許可し、その宛先をワーキングセットに追加する。それ以外のとき、そのパケットをキューに追加する。キューに追加したとき、キューの長さが閾値 q を超えたら、キューからの送信を停止し、それをユーザに知らせる。本手法では、これ以降でも、すでに接続が確立したホストと名前解決されたホストにはパケットを送信できる。

キューの先頭パケットはタイムアウト時間 d 秒毎に送信される。送信時に、ワーキングセット内の最も長い間利用されていない IP アドレスを 1 つ破棄し、送信したパケットの宛先 IP アドレスを追加する。新しく追加された IP アドレスと同じ宛先のパケットが

表 1 採取したパケットの属性
Table 1 Property of captured packets.

ユーザ	OS	日数	送信パケット数
A	Windows®XP	55	22,064
B	Windows®98	40	81,982
C	Windows®XP	39	45,859
D	Windows®2000	33	5,952
E	Windows®2000	22	12,528

表 2 宛先 IP アドレスが正引き応答に含まれた割合
Table 2 Rate of destination IP addresses included in DNS responses among all destination IP addresses.

ユーザ	割合 (%)
A	88.36
B	98.79
C	96.09
D	95.47
E	91.29

キューに含まれるなら、それらを直ちに送信する。このとき、キューが空になり、ワーキングセットの大きさが n であれば、ワーキングセット内の最も長い間使用されていない IP アドレスを 1 つ破棄して、ワーキングセットに空きを 1 つだけ用意する。

3. 実　　験

3.1 実験データ

本手法の有効性を評価するために、5 人のユーザと 3 種のワームから採取したパケットを用いて、本手法をシミュレーションした。5 人のユーザは互いに異なるネットワークに属し、日常的にインターネットを利用している。一方、ワームは CodeRedv2 (別名 : CodeRed, Baby)、Slammer (別名 : SQLSlammer, Sapphire)、Blaster (別名 : Lovsan, MSBLAST, Poza) である。

パケットは、次の 3 種類を WinDump 3.8α (パケットキャプチャドライバーは WinPcap 3.0 である) で採取した。カッコ内は WinDump のオプション (条件式) を示している。hostname は WinDump を実行するコンピュータのホスト名または IP アドレスである。

- DNS サーバから受信した正引き応答パケット
  (`port 53 and dst host hostname`)
- 送信した TCP の接続要求パケット
  (`tcp[13]&18=2 and src host hostname`)
- 送信した UDP のパケット
  (`udp and src host hostname`)

表 1 に、ユーザが利用した Windows® OS の種類、採取した日数とそのパケット数を示す。ワームのパケットは、VMware® Workstation 4 のゲスト OS (Windows® 2000 Server) で採取した。実験に用いたワームは増殖を始めると一定の速度で大量に増殖を試みるので、増殖を開始してから約 1 時間だけ採取した。ただし、Slammer は、高速に増殖を試みるので約 1 分で採取を終了した。

3.2 実験結果

3.2.1 パケットの宛先 IP アドレスと正引き応答から得られた IP アドレスの関係

まず、送信パケットの宛先 IP アドレスと正引き応答から得られた IP アドレスとの関係を調べた。**表 2** に、宛先 IP アドレスが正引き応答に含まれた割合を示す。

表 2 から、多くの宛先 IP アドレスは正引き応答に含まれることが確認された。つまり、宛先 IP アドレスと正引き応答に含まれる IP アドレスには強い相関がある。正引き応答に含まれなかった IP アドレスは、主に、ブロードキャストアドレスやネットワークプリンタなど LAN 内のアドレスであった。LAN 外の IP アドレスには、インスタント・メッセンジャーやオンラインゲームなどアプリケーション固有の IP アドレスや、接続先のサーバが指定した IP アドレスがあった。本手法では、これらの IP アドレスに対して、ウイルス・スロットルを適用する。

3.2.2 リゾルブセットの閾値 r

本手法では、正引き応答から得られた IP アドレスへの送信を許可するために、その IP アドレスをリゾルブセットに保存している。リゾルブセットの IP アドレスは、正引き応答から得られた TTL に従って削除されるが、本実験では、処理を簡単にするため、その TTL をすべて 24 時間に設定した。また、リゾルブセットには閾値 r があり、リゾルブセットの大きさがその閾値 r に達したら、リゾルブセットに IP アドレスを追加しない。そのため、その閾値 r は、ユーザの通常の操作で追加された IP アドレスの個数より大きくしなければならない。ユーザ毎にリゾルブセットの大きさを調べると、その大きさは 102 から 248 であった。そこで、ユーザの操作やアプリケーションの影響を避けるため、リゾルブセットの閾値 r を 300 に設定する。ただし、本実験では被験者の数が少ない上に、リゾルブセットの最大サイズが被験者によってばらつきがあったことから、リゾルブセットの閾値 r は再検討する必要がある。

3.2.3 ウイルス・スロットル

ウイルス・スロットルの性能を決めるパラメータには、ワーキングセットの大きさ n (一定時間内にパケットを送信できるホスト数) と、タイムアウト時間 d、キューの閾値 q がある。ここから、本手法におけるウ

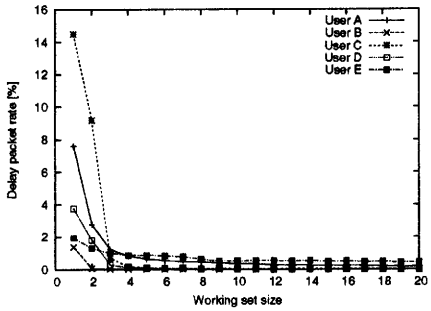

図 2 ワーキングセットの大きさ n と遅延パケットの割合
Fig. 2 Working set size vs. delay packet rate.

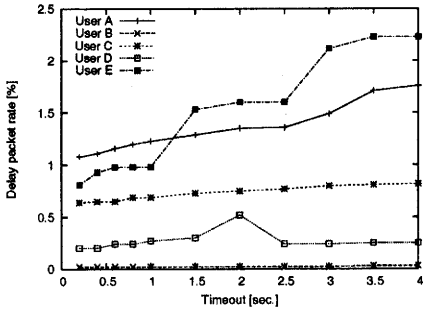

図 3 タイムアウト時間 d と遅延パケットの割合
Fig. 3 Timeout vs. delay packet rate.

イルス・スロットルのパラメータを求める。

ワーキングセットの大きさ n

ワーキングセットに含まれない IP アドレスへの送信は、キューに追加されるので、その送信は最低でもタイムアウト時間 d 秒だけ遅れる。ワーキングセットが大きければ、IP アドレスがワーキングセットに含まれる可能性が高くなるため、キューに追加されるパケット数は減少する。そのため、ワーキングセットの大きさ n によって、遅延するパケット数は変化すると考えられる。そこで、適切なワーキングセットの大きさ n を求めるために、ワーキングセットの大きさ n と遅延パケット数の関係を調べた。ワーキングセットにおける IP アドレスの置換方式は FIFO (First In First Out)、LRU (Least Recently Used)、LFU (Least Frequently Used)、ランダムの中で、LRU が最も遅延パケット数が少なかったことから、IP アドレスの置換方式を LRU とした。**図 2** に、ユーザ毎のワーキングセットの大きさ n と、全フィルタリング対象パケットに占める遅延パケットの割合に関する関係を示す。

遅延パケットの割合は、ワーキングセットの大きさ n が 3 以上になるとほとんど変化しない。これは、最近に通信した IP アドレスと相関がある IP アドレスの個数は 3 個程度であることを意味する。したがって、本手法では、ワーキングセットの大きさを 3 とする。一方、ウイルス・スロットル単体では、その大きさは 5 から 6 程度であった。

タイムアウト時間 d

タイムアウト時間 d は、キューの先頭パケットが送信される時間間隔であるとともに、ワーキングセットの IP アドレスが新しい IP アドレスに置き換えられる時間間隔である。つまり、タイムアウト時間 d は単位時間あたりに送信可能な新しい IP アドレスの個数を決める。したがって、タイムアウト時間 d も遅延パケット数に影響を与える。そこで、タイムアウト時間 d と、全フィルタリング対象パケットに占める遅延パケット数の関係を調べた。その結果を**図 3** に示す。

遅延パケットの割合は、ユーザ A と E を除いて、タイムアウト時間 d を増加しても大きな変化はなかった。タイムアウト時間 d は小さくなるにつれて、全体の遅延時間は短くなるが、誤ってワームのパケットが送信されることが多くなる。また、1 秒以下の短い時間間隔で処理する場合、システムの負荷や OS の実装によって、タイムアウト時間 d の誤差が大きくなる。図 3 では、タイムアウト時間 d が 1 秒以下のとき、いずれのユーザも遅延パケットの割合に大きな差はないので、タイムアウト時間 d を 1 秒に設定する。Williamson は遅延パケットの割合を 1%以下におさえる値として 1 秒を設定している[3]。

キューの閾値 q

最後に、キューの閾値 q を決める。ウイルス・スロットルでは、ワーキングセットに含まれない IP アドレスを宛先とするパケットは、キューに追加される。単位時間に異なる IP アドレスへ大量のリクエストが発生したとき、長いキューができる。その長さが閾値 q を超えたとき、キューのパケットは破棄され、これ以降、パケットはキューへ追加されない。したがって、その閾値 q はユーザの通常の操作で発生したキューの最大サイズより大きくしなければならない。実験では、本手法のキューの最大サイズは 2 から 16、ウイルス・スロットル単体では 16 から 103 であった。本手法では、ユーザの操作やアプリケーションの影響を避けるため、実験で得られた最大値 16 より十分に大きな値として閾値 q を 100 に設定する。ウイルス・スロットル単体では、最大値が 103 であったので、200 に設定する。

表 3 実験パラメータの一覧
Table 3 Experimental parameters.

	n	d	q	r	置換方式
本手法	3	1	100	300	LRU
ウイルス・スロットル	6	1	200	–	LRU

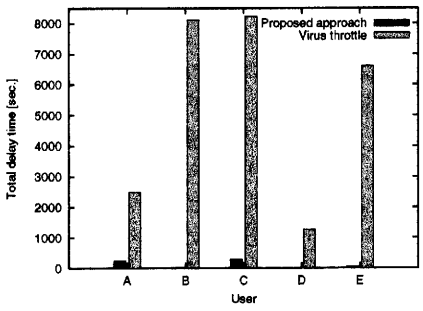

図 4 総遅延時間
Fig. 4 Total delay time.

3.2.4 性能評価

本実験で求めたパラメータ（**表 3**）で、本手法とウイルス・スロットル単体について、性能を評価した。

図 4 に各ユーザの通信に関する総遅延時間を示す。各ユーザ毎に、左側の棒グラフが本手法、右側の棒グラフがウイルス・スロットル単体の総遅延時間を表す。正引き応答を利用することによって、総遅延時間が飛躍的に改善されたことがわかる。

さらに、各ユーザについて、1 つのパケットがキューに追加されてから送信されるまでの遅延時間の最大値（最大遅延時間）を評価すると、本手法では、1 秒から 2 秒で、平均値は、1.2 秒であった。また、その遅延が発生する頻度も非常に少ないことから、この最大遅延時間は、実用に耐えられると考えられる。一方、ウイルス・スロットル単体では、最大遅延時間は 5 秒から 16 秒で、平均値は、10.2 秒であった。

次に、ワームの増殖を停止するまでにかかる時間を評価する。本実験で用いたワームは、すべて名前解決を行わないので、単純に閾値 q の大きさで、増殖を停止するまでにかかる時間が決まる。本手法のキューの閾値 q は、ウイルス・スロットル単体の 2 分の 1 であるため、本手法はウイルス・スロットル単体よりも約 2 倍程度だけ速くワームの増殖を停止できた。

4. 考　　察

実験で採取したパケットでは評価できなかった内容について考察する。まず、ユーザの通信とワームの通信が混在している場合について考える。本手法は、ワームに感染した後でも、名前解決されたホストであれば、それらのホストと通信できる。感染時に、ユーザのパケットが誤って破棄される割合は、正引き応答で得られなかった IP アドレスへ送信したパケットの割合に近い値が得られることが予想される。そこで、仮に、ユーザ A が Blaster に感染したことを想定して、ユーザ A がある日の 1 時間に送信したパケットと、Blaster が 1 時間に送信したパケットを融合した。そのデータを本手法で評価した結果、ユーザのパケットが誤って破棄される割合は、正引き応答で得られなかった IP アドレスへ送信したパケットの割合に一致した。つまり、ワームに感染した後であっても、名前解決されたホストであれば、それらのホストと通信できる。

次に、名前解決を行うワーム（増殖先のホストをホスト名で指定するワーム）について考察する。名前解決を行うワームは、2 種類のワームが考えられる☆。1 つは、ホスト名をランダムに生成するワームである。もう 1 つは、トポロジースキャンを行うワーム[8] である。トポロジースキャンワームとは、侵入したホスト内部の情報からホスト名を探し出し、それらに増殖するワームである。例えば、Web サーバを標的とするワームの場合、侵入したサーバ内の HTML ドキュメントからホスト名を抽出する。

本手法は、リゾルブセットの大きさが閾値 r に達したら、IP アドレスをリゾルブセットに追加しない。そのため、過去に発見されたワームのように、単位時間あたりに多くのホストへパケットを送信すれば、瞬く間に、リゾルブセットとワーキングセットの大きさがそれぞれの最大値に達して、最終的に増殖は停止する。その機能を確認するために、CodeRedv2、Slammer、Blaster が全て名前解決を行ったと仮定して、増殖を停止するのに要した時間を評価した。その結果を**図 5** に示す。各ワーム毎に、左側の棒グラフは名前解決を行わない場合（Original worm）、右側の棒グラフは名前解決を行う場合（Name resolution worm）に関する結果である。ただし、Slammer は、CodeRedv2 や Blaster と比べて、非常に速く停止できるため、Slammer に関する時間のスケールは右側の y 軸に従う。

図 5 から、名前解決を行うワームは、名前解決を行わないワームと比べて、増殖を停止するまでに時間を要することがわかる。これは、リゾルブセットの閾値 q が大きいためである。この値は、利用頻度の低い IP

☆ 現時点では、Windows® を標的にした名前解決を行うワームは確認されていない。

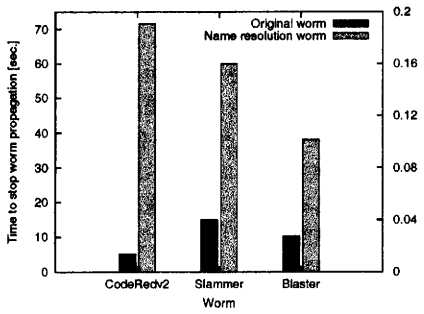

図 5 増殖停止までに要した時間
Fig. 5 Time to stop worm propagation.

アドレスを積極的に削除するなどの工夫によって、さらに小さい値に改善できると考えられるので、今後の課題としたい。

これらの他にも、名前解決を行えないようにするワームや、パケットフィルタリング自体を無効にするワームが考えられる。前者に感染すると、ユーザはほとんどの通信を行えないが、ウイルス・スロットルによって、ワームの増殖は阻止されると考えられる。後者の場合、図1のように、パケットフィルタリングを行うコンピュータと、ユーザが利用するコンピュータを分離することによって、フィルタリング機能を保護できると考えられる。

最後に、通信が遅延しやすいアプリケーションについて考察する。本手法では、ウイルス・スロットルを適用するため、名前解決を必要としないアプリケーションの通信は遅延しやすい。そのアプリケーションは、例えば、WinMX^{TM} など P2P ネットワークを利用するアプリケーションである。P2P アプリケーションは、接続先のホストを IP アドレスで指定することが多い。さらに、単位時間あたりに通信するホストの個数は従来のアプリケーションよりも多い。ウイルス・スロットルは、ユーザが単位時間あたりに通信するホストの個数は少数であることと、ユーザが通信するホストは最近、通信したホストに偏っていることに着目した手法である。そのため、ユーザが P2P アプリケーションを利用すると、ウイルス・スロットルの性能は低下する。実際に、P2P アプリケーションを1週間程度利用したときに採取したデータを評価すると、遅延パケットの割合は、20.13%に達した。この値は、P2P アプリケーションを利用しないときと比べて約 16 倍の値である。また、正引き応答に含まれた IP アドレスの割合は、P2P アプリケーションを利用しないときと比べて非常に小さな値 (16.66%) であった。本手法は、ウイルス・スロットル単体より、総遅延時間を約2分の1に短縮したが、これは、57.75%のパケットが名前解決されたホストを宛先としていたためであった。今後は、P2P アプリケーションを利用しても性能を低下させない方法を検討する必要がある。

5. おわりに

本報告では、正引き応答を利用したパケットフィルタリングを提案した。実験では、ユーザ5人と、CodeRedv2、Slammer、Blaster から採取したパケットを用いて、ワームに対する有効性を評価した。その結果、正引き応答を利用することによって、ウイルス・スロットル単体よりも、送信時の遅延時間とワームの増殖停止に要する時間を短縮できることを示した。また、ワームに感染している状態でも、名前解決されたホストであれば通信が可能であることを示した。

今後は、被験者やアプリケーションの種類を増やして、リゾルブセットの最小化や、プロトタイプの実装およびその性能評価を行いたい。

謝辞 本研究は、文部科学省 科学研究費補助金 若手研究 B (課題番号:15700075) の助成による。

参考文献

1) Staniford, S., Ellis, D. and Weaver, N.: NetWorm.org - The Worm Information Center (2003), http://www.networm.org.
2) Messmer, E.: Behavior Blocking Repels New Viruses, Network World Fusion News (2002).
3) Williamson, M. M.: Throttling Viruses: Restricting Propagation to Defeat Malicious Mobile Code, *In Proc. the ACSAC Security Conference*, pp. 61–68 (2002).
4) 面和成, 鳥居悟: ワームによるランダムスキャンの検知方式の検討, コンピュータセキュリティシンポジウム 2003, pp. 103–108 (2003).
5) Twycross, J. and Williamson, M. M.: Implementing and Testing a Virus Throttle, *In Proc. 12th USENIX Security Symposium* pp. 285–294 (2003).
6) Williamson, M. M., Twycross, J., Griffin, J. and Norman, A.: Virus Throttling, Virus Bulletin, Vol.3, pp. 8–11 (2003).
7) 力武健次, 野川裕記, 管谷史昭, 中尾康二, 下條真司: DNS と TCP コネクションに関する挙動の解析, コンピュータセキュリティシンポジウム 2003, pp. 521–526 (2003).
8) Staniford, S., Paxson, V. and Weaver, N.: How to Own the Internet in Your Spare Time, *USENIX Security Symposium 2002*, pp. 149–167 (2002).