2003－DSM－31　(18)
2003／11／26

フローを用いた特定トラフィック検出システムの運用

藤井 聖†、中尾 嘉宏†、中村 豊††、藤川 和利††、砂原 秀樹†††

高機能なエッジルータでは、トラフィックをフローと呼ばれる単位で集約し、そのサマリ情報を出力する機能が提供されている。提供されたサマリ情報を蓄積し、分析することにより、トラフィックに関する様々な情報を得ることが可能であり、ネットワーク運用への貢献が期待されている。我々の研究グループでは、そうした情報を利用することで、ワームや DoS 攻撃、P2P など、様々な特徴を持ったトラフィックを検出可能なシステムを提案し、運用している。本稿では、提案システムの概要および構成について述べる。また、8 月に奈良先端科学技術大学院大学において、本システムにより W32/MSBlaster、W32/Welchi ワームの検出が行われた事例について報告する。

Management of the System, which can Detect Specified Traffics by Using Flows

FUJII Satoshi†, NAKAO Yoshihiro†, NAKAMURA Yutaka††,
FUJIKAWA Kazutoshi††, SUNAHARA Hideki†††

High-performance edge routers have a function to observe packets as flows and to export summary information of the flows. By storing and analyzing the summary information, we can get various kinds of traffic information, which is expected to contribute to the network management. We proposed a system, which can detect specified traffics, such as worm activities, DoS attacks, and P2P traffics by using the information, and we also have been managing our system. In this paper, we describe a structure of our system. In addition, we also discuss how our system was applied to detect W32/MSBlaster and W32/Welch worms in August, 2003 in our campus.

1. はじめに

コンピュータネットワーク、特にインターネット、イントラネットの広帯域化に伴い、ネットワーク上で利用されるサービスの多様化が進んでいる。

従来、ネットワーク上を流れるトラフィックのほとんどは、HTTP や FTP などのサーバクライアント型サービスに関するものであった。こうした旧来型のトラフィックにおいては、クライアントの挙動はすべてサーバ側で把握が可能である。そのため、ネットワーク管理者は、回線使用量の把握や輻輳回避のために、時間ごとの通信量変化、サブネット間の通信量比較などを把握しておくだけで十分であった。

しかし、Instant Messenger や各種ファイル共有ソフトウェアなどの P2P アプリケーションや、ワーム(ネットワークを介して脆弱性を持つコンピュータに自動的に感染するコンピュータウイルス)など、インターネット上を流れるトラフィックが多様化している。

このようなネットワークでは、ネットワーク管理者が把握すべきトラフィック情報は、全体トラフィック量だけではなく、異常な通信を行っているユーザや、ワームによるトラフィックなど、特定の特徴を持つトラフィックを検出することが重要となっている。こうした背景の中、各ネットワーク機器ベンダは、様々な角度からトラフィックを観測できる機能を提供している。Cisco Systems 社が策定した NetFlow[1]はその一つであるが、実際の運用に利用されている例は少なく、ノウハウは十分に蓄積されていない。

このため、我々の研究グループでは、NetFlow を用い現実的なコストで柔軟なトラフィック分析を目指したシステム[2]を提案している。本稿では、提案システムを用いたトラフィック傾向の分析方法や特定トラフィックの検出手法について述べる。

以下、2.で既存のトラフィック検出および計測技術を挙げ、その問題点を述べる。3.で提案システムの概要を述べ、4.で計測を行った環境について述べる。5.で提案システムを利用した一般的なトラフィック観測の例と、ワームの検出に応用された事例について報告する。

†　奈良先端科学技術大学院大学 情報科学研究科 〒630–0192 奈良県生駒市高山町 8916–5
††, †††　奈良先端科学技術大学院大学 情報科学センター 〒630–0192 奈良県生駒市高山町 8916–5
†　Graduate School of Information Science, NARA Institute of Science and Technology, Takayama-cho 8916–5, Ikoma-shi, Nara, 630–0192 Japan.
††, †††　Information Technology Center, NARA Institute of Science and Technology, Takayama-cho 8916–5, Ikoma-shi, Nara, 630–0192 Japan.
E-mail: † {sato-fu, yoshih-n}@is.aist-nara.ac.jp, †† {yutaka-n, fujikawa}@itc.aist-nara.ac.jp, ††† suna@wide.ad.jp

2. 既存技術

本節では、既存の特定トラフィック検出技術、および、トラフィックの計測技術について述べる。

2.1. 特定トラフィック検出技術

2.1.1. Snort

Snort[3] は Open Source の IDS(Intrusion Detection System)ソフトウェアであり、事前に用意したルールパターンにマッチするトラフィックをリアルタイムに検出することができる。ルールパターンは充実しており、ウイルスの攻撃によるトラフィックや一部の P2P 型ファイル共有アプリケーションによるトラフィックなどが検出可能である。また、ユーザが、前後のパケットとのつながりや、パケットのペイロード内のバイト列を考慮したルールを記述可能であり、様々なトラフィック検出に応用可能である。

しかし、Snort は一般 PC 上での動作が想定されているソフトウェアであり、広帯域なネットワークリンク上での動作は困難である。また、当然、既知のパターンのトラフィックしか検出できない。長期に渡って蓄積されたトラフィック情報から、特定のトラフィック検出をするといった用途にも不向きである。

2.1.2. ハードウェア IDS 製品

ハードウェア実装された IDS 製品も Snort と同様に、特定トラフィック検出に利用可能である。こうした製品は一般的に、Snort を一般 PC で利用した場合に比べて、パフォーマンスの点で有利であり、数 Gbps 程度の広帯域リンク上で利用できる製品もある。

しかし、一般に、ハードウェア実装された IDS 製品は、Snort のようなソフトウェアに比べ、ユーザ定義ルールの柔軟性に欠け、高価である。

2.2. トラフィック観測技術

2.2.1. MRTG

MRTG[4]は SNMP[5]プロトコルを利用したグラフ描画ツールであり、主にルータを通過するトラフィック量の視覚化に用いられている。手軽に使える反面、トラフィック量以外の測定に利用するのは難しい。また、古い計測結果は自動的に粒度が落とされて保存される。このことは、長期の計測において十分に分析できない可能性がある。

2.2.2. tcpdump

tcpdump[5]は一般コンピュータで動作する、トラフィック観測ツールであり、ネットワーク上を流れるパケットそのものの表示、記録が可能である。パケットをそのまま表示・記録するので、計測結果の正確性という面では最も優れている。

しかし、数 Gbps を超えるような広帯域ネットワークにおいて、パケットそのものを記録することは、一般コンピュータのパケットキャプチャ性能、ストレージ容量などを考慮すると、現実的には不可能である。

2.2.3. NetFlow

NetFlow は本稿の提案システムが利用しているトラフィック観測技術である。NetFlow は Cisco Systems 社が策定したトラフィック観測技術であり、同社のルータ、スイッチ製品に実装されている。Version 1 から Version 9 まで各種バージョンが存在しており、Juniper Networks 社や Extreme Networks 社など他のネットワーク機器ベンダの製品も一部バージョンに対応してきている。Cisco Systems 社のルータ製品では、NetFlow の情報はルーティング用のキャッシュから作成することで、計測によるルータの負荷を低く押さえている。また、作成される情報も 1 フローあたり 48byte とコンパクトであり、長期計測での利用が可能である。

しかし、NetFlow に対応した機器は一部のベンダ製品に限られ一般的に高価である。また、実際の運用に活用されている例は少なくノウハウの蓄積が十分であるとはいえない。

3. 提案システムの概要

本節では、フローの定義、収集項目を述べた後に、我々が提案するシステムの構成を述べる。

3.1. フローの定義

我々は、フローを、Cisco Systems 社の NetFlow の定義に準じ、次の 7 つの項目が等しい一連の片方向パケットとしている。

1. input logical interface
2. Layer 3 protocol type
3. source IP address
4. destination IP address
5. source port number
6. destination port number
7. IP type of service (ToS)

TCP におけるコネクションに近似した概念である。しかし、片方向であるという点が異なる。

どこまでの一連パケットを同一フローと見なすかは、対応機器ベンダの実装により多少違いがあり、一部のパラメータは設定により変更が可能である。Cisco Systems 社の機器の実装では、次のいずれかが満たされた場合にそのフローが終了したと見なし、そのフローに関する情報を出力する。

- TCP に関するフローの場合、FIN または RST フラグが立ったパケットを受信した場合。
- あるフローに関して 15 秒間[1]パケットが流れなかった場合。
- そのフローに関して 15 秒未満の間隔[1]で 30 分間以上[2]パケットが流れた場合。
- NetFlow 情報用キャッシュ (65536 フロー分[2])があふれた場合の一番古いフロー。

---

[1] 初期設定値であり、設定により変更可能。Juniper Networks 社製品の場合、初期設定値は 60 秒。

[2] 初期設定値であり、設定により変更可能。

3.2. 収集項目

提案システムは NetFlow Version 5 で定義されている情報のサブセットを蓄積し、それを基にトラフィック解析やトラフィック検出を行う。

NetFlow Version 5 では、フローごとに、表 1 の情報を取得し、出力する。ただし、機器の動作モードによっては一部取得できない情報もある。

表 1. NetFlow Version 5 パケットに含まれる情報

1	Source IP address
2	Destination IP address
3	IP address of next hop
4	SNMP index of input interface
5	SNMP index of output interface
6	number of packets in the flow
7	cumulative bytes in the flow
8	system up time when the first packet of the flow was received
9	system up time when the last packet of the flow was received
10	TCP/UDP source port number
11	TCP/UDP destination port number
12	IP protocol type
13	IP type of service (ToS)
14	Cumulative OR of TCP flags

3.3. 提案システム構成

提案システムの構成を、図 1 に示す。

NetFlow 機能を持ったネットワークデバイス(主としてルータ)は、自デバイスを経由するトラフィックをフロー単位で収集し、各フローに関する表 1 に示された情報を UDP パケットとして出力する。提案システムの「Capture and Store プログラム」はそのパケットを受信し、表 1 に示された情報の一部をデータベースに格納する。提案システムにおいてデータベースに格納する情報は、表 1 における、1 から 7、および 10 から 12 の情報である。

本稿では、フロー情報を出力するデバイスを「NetFlow Exporter」、出力されるパケットを「NetFlow パケット」と呼ぶ。また、データベースに格納された各フローに関する情報をフローレコードと呼ぶ。「Visualizer プログラム」はデータベースに対して SQL クエリを発行し、蓄積されたフローレコードを表やグラフの形に加工する。SQL クエリは自由に設定することが可能であり、様々な角度から、トラフィックを計測・分析することができる。また、現在、フローレコードに対して、数学的解析手法などを施し、その結果に基づいて計測・分析を行う「Analyzer プログラム」の開発も並行して行っている。

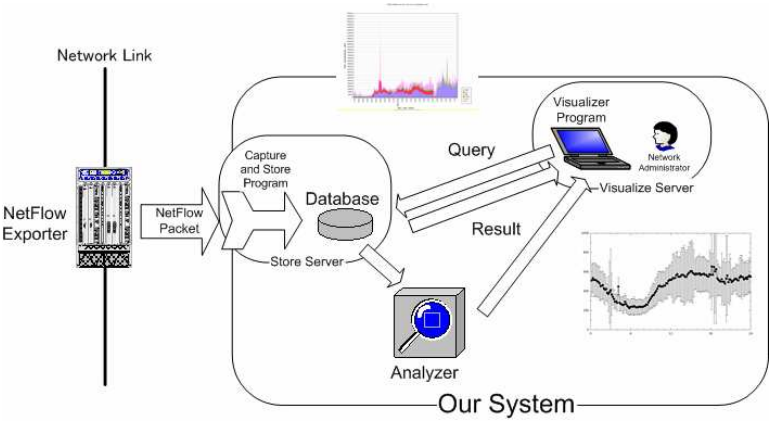

図 1. 提案システムの構成図

3.4. データベースの設計

フローレコードを格納するための、flow テーブルの定義を図 2 に示す。

```
CREATE TABLE `flow` (
  `fid` bigint(20) NOT NULL auto_increment,
  `unix_secs` int(11) default NULL,
  `srcaddr` int(10) unsigned default NULL,
  `fqdn_srcaddr` text,
  `dstaddr` int(10) unsigned default NULL,
  `fqdn_dstaddr` text,
  `nexthop` int(10) unsigned default NULL,
  `fqdn_nexthop` text,
  `input` int(10) unsigned default NULL,
  `output` int(10) unsigned default NULL,
  `pkts` int(10) unsigned default NULL,
  `bytes` int(10) unsigned default NULL,
  `srcport` smallint(5) unsigned default NULL,
  `dstport` smallint(5) unsigned default NULL,
  `prot` smallint(5) unsigned default NULL,
  PRIMARY KEY  (`fid`),
  KEY `idx_srcaddr` (`srcaddr`),
  KEY `idx_dstaddr` (`dstaddr`),
  KEY `idx_srcport` (`srcport`),
  KEY `idx_dstport` (`dstport`),
  KEY `idx_prot` (`prot`),
  KEY `idx_unix_secs` (`unix_secs`)
) TYPE=MyISAM
```

図 2. flow テーブルの定義

fid はテーブル上のプライマリキーであり、unix_secs はそのフローレコードを受信した時間である。srcaddr、dstaddr、nexthop、input、output、pkts、byte がそれぞれ表 1 の 1 から 7 に、srcport、dstport、prot がそれぞれ 10 から 12 に対応している。その他、SQL クエリ実行時の高速化を図るため、検索に高頻度に使用される項目に対してインデックスを作成している。fqdn_srcaddr、fqdn_dstaddr および、fqdn_nexthop は我々が、独自に拡張を行った項目であり、srcaddr、dstaddr、nexthop それぞれの IP アドレスを逆引きした FQDN が格納される。逆引き作業はフローレコードの格納とは非同期に行っている。

flow テーブルの他には、「/etc/services」の内容を格納した、service テーブル、「/etc/protocols」の内容を格納した protocol テーブルを作成した。これらのテーブルは Well-known ポート以外を利用したトラフィック検出時や、トラフィック観測結果としてサービス名やプロトコル名を表示したい時に利用する。

4. 運用環境

本節では提案システムを用いて奈良先端科学技術大学院大学(以下本学)におけるトラフィックを収集した際のポリシーおよび環境について述べる。

4.1. 計測データ収集ポリシー

トラフィックにはユーザのプライバシに関する情報やネットワーク運用に直結する情報が大量に含まれている。そのため、我々はトラフィックを計測するにあたって以下の 3 点について考慮している。

- パケットのレイヤ 3 およびレイヤ 4 ヘッダ部分の情報のみを利用する。
- 収集したトラフィックデータは研究、運用目的にのみ用いる。
- 研究目的に用いる場合においても、データを外部に公開する際はホストやユーザの情報が特定されることがないようにデータを隠蔽もしくは変換する[7]。

4.2. 機器構成

「NetFlow Exporter」として Juniper Networks 社のルータ製品である M-20 を使用している。

M-20 は通過するパケットをサンプリングレート 500[3]でキャプチャし、キャプチャされたパケットに基づいてフロー情報を作成する。作成されたフロー情報は 1 分ごとに「NetFlow パケット」として「Capture and Store プログラム」に対して出力される。サンプリングレートを 500 と設定したのは、「NetFlow Exporter」における、フロー情報作成時の負荷を軽減させるためである。

「Capture and Store プログラム」の動作プラットホームおよびデータベースサーバとして、一般の PC を使用しておりその構成は、

- OS: FreeBSD 4.7-RELEASE
- CPU: Pentium III 1.2GHz
- Memory: 512MB
- Hard Disk: 80GB

である。データベースマネジメントシステム(DBMS)としては、MySQL 4.0.14-standardを用いている。

「Visualizer プログラム」の動作プラットホームとしても同様に PC を使用しており、その構成は、

- OS: Mandrake Linux 9.1
- CPU: Pentium III 1.0 GHz
- Memory: 512MB
- Hard Disk: 120GB

である。

4.3. 測定場所

前述の「NetFlow Exporter」、Juniper Networks M-20 は、本学の境界ルータとして運用されており、本学内から学外へ向かうパケットのほとんどはこのデバイスを経由する。

今回は、このデバイスを経由し、学内から学外へ向かうトラフィックを観測対象とした。

観測対象となる、学内から学外へ向けたトラフィック量は、バイトカウントにして平均 4.3Mbps、パケットカウントにして平均 1kpps 程度である。

[3] 500 パケットに 1 パケットをサンプリング。

4.4. データ量

本システムは 2003 年 7 月 22 日より実験稼働を開始し、10 月 22 日現在までの 4 ヶ月間に約 985 万件のフローレコードを収集した。フローレコードを格納したデータベースのディスク上の容量は、1.4GBytes である。検索高速化用のインデックスファイルを含めても、2.3GBytes となっている。現在普及している PC のディスク容量を考慮すると、提案システムは、年単位の長期計測にも利用可能であるといえる。

5. 収集レコード分析

本節では、提案システムで収集したフローレコードを用いた、一般的なトラフィック観測の例と、ワームの検出事例について報告する。

5.1. 一般的なトラフィック観測例

フローレコードに対して、srcport、または dstport 項目を用いて集約することで、ポートごとのトラフィック傾向を知ることができる。同様に、srcaddr、dstaddr を用いることにより、同様に、サブネットごと、ホストごとのトラフィック傾向の調査も可能である。

図 3[4]は、過去 1 時間の UDP トラフィックについて、転送バイト数が多い宛先ポート番号 10 個を選択する SQL クエリ例であり、表 2 はその実行結果例である。

```
SELECT
    dstport,
    sum(bytes) as sum_bytes
FROM
    flow
WHERE
    unix_secs >= unix_timestamp() - 60 * 60 AND
    prot = 17
GROUP BY
    dstport
ORDER BY
    sum_bytes desc limit 10
```

図 3. 転送量が多い宛先 UDP ポートの検出(SQL クエリ)

表 2. 転送量が多い宛先 UDP ポートの検出(結果例)

dstport	sum_bytes
53	1820
54241	1267
123	1216
2300	1077
600	808
19742	560
10129	210
3544	204
3012	176
13070	140

表 2 から、DNS 問い合わせに利用される 53 番ポートや、NTP に利用される 123 番ポートの他、54251 や 2300 といったポートでの転送バイト数が多いことがわかる。サンプリングレートが 1/500 であるため、実際の転送量は、表 2 の sum_bytes で示された転送量の、500 倍程度となる。

図 5 は、「Visualizer プログラム」を用いて UDP フローのポートごとの転送バイト数を、図 6 はパケットカウントをグラフ化した例である。

[4]本稿で例示する SQL クエリは MySQL 4.0.14-standard 用であるが、無改変もしくは少量の変更を加えることによって、他の DBMS でも動作可能である。

5.2. ワーム検出事例

2003 年 8 月に、W32/MSBlaster、W32/Welchi と呼ばれるワームが世界的に流行した。このワームに感染したホストは特徴のあるトラフィックを発生させることが知られており、W32/MSBlaster に感染したホストは、宛先を TCP の 135 番ポートとするパケットを、W32/Welchi はそれに加えて ICMP パケットを、それぞれ大量に発生させる。

本学内においても、2003 年 8 月 21 日を境に、多数のホストがこれらのワームに感染した。図 7 は、「Visualizer プログラム」を用いて作成した、2003 年 8 月 20 日午前 0 時から 1 週間の、宛先を TCP の 135 番ポートとするパケット数と ICMP パケット数の変化量である。また、図 8 は感染疑惑ホスト数の変化量である。

ここでは、あるホストを source address とする ICMP フローについて、30 分間のパケット数(フローレコードの pkts フィールドの値の合計)が 10 を超えた場合、そのホストを感染疑惑ホストと定義した。パケットベースで 1/500 にサンプリングをしているので、感染疑惑ホストは 30 分間に 5000 パケット程度の ICMP パケットを出力していることになる。通常状態で、一般のホストがこれほど大量の ICMP パケットを出力することなく、感染疑惑ホストの定義は妥当であると考えられる。

これらグラフからは、それまではほとんど観測されていなかった、宛先を 135/tcp とするフローが 20 日 16 時 30 分ごろから、ICMP パケット数が同日 20 時頃から多量に観測され始めたことがわかる。また、その後、21 日 12 時付近に感染疑惑ホストが激増していることもわかる。

また、具体的な感染疑惑ホストの割り出しも可能である。図 4 と、表 3 は 2003 年 8 月 21 日からの 30 分間に大量の ICMP パケットを出力しているホストを特定する SQL クエリとその結果の一部である。

こうしたグラフや SQL クエリを利用することで、感染事実の把握だけにとどまらず、感染ホストの特定作業も容易になる。事実、感染疑惑ホストとして検出されたホストはすべてワームに感染しており、ワームの駆除作業を円滑に行うことができた。

```
SELECT
    inet_ntoa(srcaddr) as srcaddr,
    fqdn_srcaddr,
    sum(pkts) as num_of_icmp_pkts
FROM
    flow
WHERE
    unix_secs >= unix_timestamp('2003-08-21 12:00:00') AND
    unix_secs <  unix_timestamp('2003-08-21 12:30:00') AND
    prot = 1
GROUP BY
    srcaddr
HAVING
    num_of_icmp_pkts > 10
ORDER BY
    num_of_icmp_pkts desc AND
    prot = 1
GROUP BY
    srcaddr
ORDER BY
    num_of_icmp_pkts  desc
```

図 4. 感染疑惑ホストを検出(SQL クエリ)

表 3. 感染疑惑ホストを検出(結果)

srcaddr	fqdn_srcaddr	num_of_icmp_pkts
163.221.	ist-nara.ac.jp.	432
163.221.	t-nara.ac.jp.	413
163.221.	ist-nara.ac.jp.	395
163.221.	ist-nara.ac.jp.	389
163.221.	ist-nara.ac.jp.	388
163.221.	aist-nara.ac.jp.	383
163.221.	ist-nara.ac.jp.	373
163.221.	t-nara.ac.jp.	368
163.221.	aist-nara.ac.jp.	361
163.221.	t-nara.ac.jp.	360
163.221.	ist-nara.ac.jp.	354

6. おわりに

本稿では、既存のトラフィック検出および計測技術とその問題点を述べた。また、提案システムの概要と計測環境について述べ、提案システムが一般的なトラフィックの把握や、ワーム検出に応用できることを示した。

提案システムの、今後の課題として、

- NetFlow Version 5 パケット以外に、tcpdump の出力など複数の入力フォーマットに対応させる。
- 観測拠点を増やし、多地点間のトラフィックの差を利用した計測を可能にする。
- P2P やワーム、DoS 攻撃などのトラフィックの共通成分を抜き出して、これらのトラフィックの自動検出を可能にする。
- 検出結果をネットワークトポロジや QoS 制御に反映させる機構を実装し、IDS に近い機能を実現する。

などが挙げられる。

7. 参考文献

[1] “NetFlow”, http://www.cisco.com/warp/public/732/Tech/nmp/netflow/index.shtml

[2] 中尾嘉宏、許先明、中村豊、藤川和利、砂原秀樹、”自由度の高い解析可能なネットワークトラフィック計測システムの実現”、電子情報通信学会通信方式研究会、2003 年 11 月(発表予定)

[3] “Snort”, http://www.snort.org/

[4] “MRTG”, http://people.ee.ethz.ch/~oetiker/webtools/mrtg/

[5] “SNMP”, RFC1157, May, 1990

[6] “tcpdump”, http://www.tcpdump.org/

[7] “Guidelines for Protecting User Privacy in WIDE Traffic Traces”, http://tracer.csl.sony.co.jp/mawi/guideline.txt

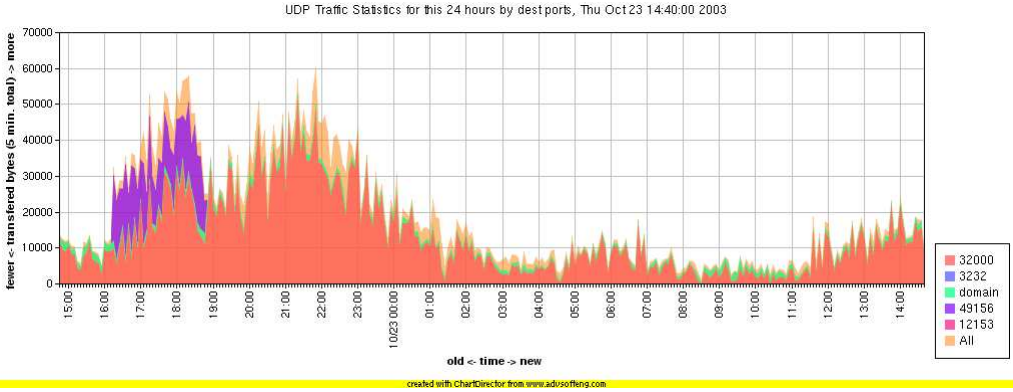

図 5. 宛先ポートごとの転送バイト数(UDP)

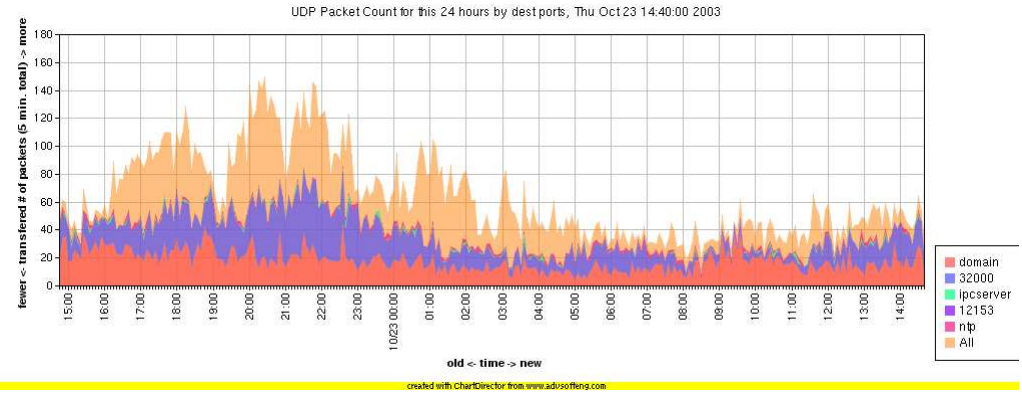

図 6. 宛先ポートごとの転送パケット数(UDP)

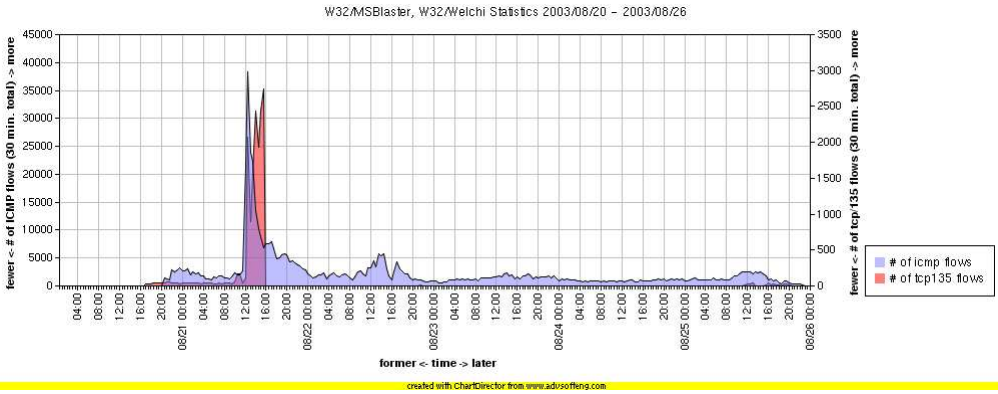

図 7. 135/tcp および ICMP パケットの推移

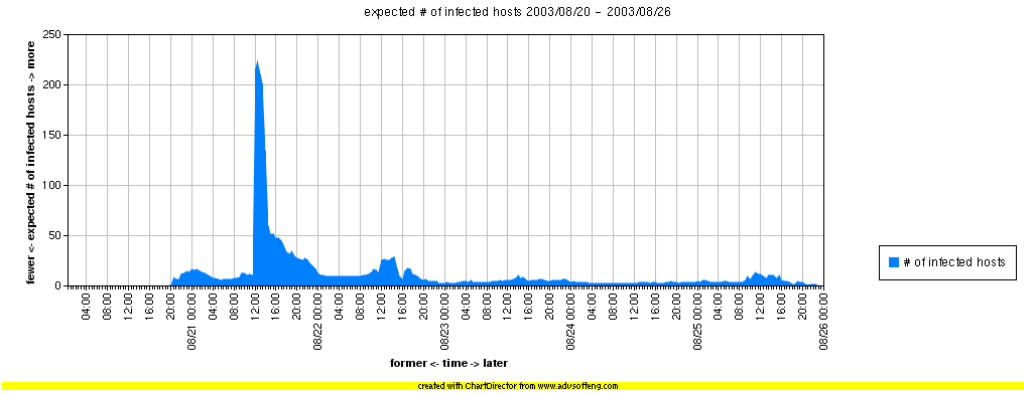

図 8. 感染疑惑ホスト数の推移