PAPER

Evaluation of Anomaly Detection Method Based on Pattern Recognition

Romain FONTUGNE[†a)], **Yosuke HIMURA**[††b)], ***and*** **Kensuke FUKUDA**[†††,†c)], ***Members***

SUMMARY The number of threats on the Internet is rapidly increasing, and anomaly detection has become of increasing importance. High-speed backbone traffic is particularly degraded, but their analysis is a complicated task due to the amount of data, the lack of payload data, the asymmetric routing and the use of sampling techniques. Most anomaly detection schemes focus on the statistical properties of network traffic and highlight anomalous traffic through their singularities. In this paper, we concentrate on unusual traffic distributions, which are easily identifiable in temporal-spatial space (e.g., time/address or port). We present an anomaly detection method that uses a pattern recognition technique to identify anomalies in pictures representing traffic. The main advantage of this method is its ability to detect attacks involving mice flows. We evaluate the parameter set and the effectiveness of this approach by analyzing six years of Internet traffic collected from a trans-Pacific link. We show several examples of detected anomalies and compare our results with those of two other methods. The comparison indicates that the only anomalies detected by the pattern-recognition-based method are mainly malicious traffic with a few packets.

key words: *anomaly detection, pattern recognition, Internet traffic*

1. Introduction

Identification of anomalies in Internet backbone traffic is an important task for securing operational networks and maintaining optimal network resources. However, analyzing traffic taken from a high speed Internet backbone — where the payload data is usually inaccessible, the traffic is asymmetric and often sampled — is a challenging issue. A significant difficulty is to accurately characterize anomalous traffic while a wide diversity of threat is constantly emerging. Researchers have mainly tried to handle anomaly detection as a statistical issue [1]–[3], but they have faced a common problem in estimating relevant statistics from small (mice) flows. Detecting low-intensity anomalous traffic is essential since sophisticated or large-scale attacks tend to be distributed processes involving numerous hosts with small amount of traffic each.

The main idea of our work is to apply an image processing technique to anomaly detection; traffic is monitored in 2-D scatter plot where each plot represents packets and anomalous traffics appear as "lines." Anomalies are easily extracted with a line detector and the original data can be retrieved from the identified plots. The main advantage of this method is its ability to quickly and precisely report anomalies involving a tiny number of packets. The method inspects only packet header information at a single point in the network, and it requires no prior information on the traffic or port numbers.

In [4] we proposed the basic idea of this new approach based on pattern recognition of network-related information. Also, the proposed method was partially validated with a single traffic trace. In this paper, we thoroughly investigate this method; first, we estimate the dependencies of its parameter set. Next, we characterize anomalous behaviors in a large-scale publicly available traffic data set (for 6 years) taken from a trans-Pacific link. We also compare the results of our method with those of different methods based on multiresolution gamma modeling [2] and K-means [5]. Finally, we highlight the different strengths and weaknesses of each method, and emphasize the need for using different detection approaches together.

2. Related Work

Researchers have taken an interest in anomaly detection and general traffic classification (e.g., [6], [7]) in Internet backbone traffic. Most of their proposals address the anomaly detection problem through volume variance or traffic feature discrimination.

Volume based approaches (e.g., [1]) are effective for identifying local or global variances over the entire traffic volume. However, a large number of anomalies do not alter the traffic volume, so these methods are only good for a limited class of anomalies.

Since numerous anomalies cause abnormal utilization of ports or addresses (source and/or destination), the quality of anomaly detectors can be considerably improved by dealing with these traffic features. The methods define common characteristics of traffic and discriminate unusual flows, by using statistical method such as principal component analysis [3], or gamma modeling [2]. Statistical techniques compute data in a manner that original flows are difficult to recover from the discovered anomalies, and the detected anomalies cannot be accurately identified. Often, a more precise identification can be obtained with random aggregated traffic (or sketches) [2], [8]. Statistical analyses do not lend themselves to detecting mice, but anomalies tend to be

Manuscript received July 3, 2009.
Manuscript revised September 18, 2009.

[†] The authors are with the Graduate University for Advanced Studies, Tokyo, 101-8430 Japan.
[††] The author is with the University of Tokyo, Tokyo, 113-8656 Japan.
[†††] The author is with National Institute of Informatics/PRESTO, JST, Tokyo, 101-8430 Japan.
a) E-mail: romain@nii.ac.jp
b) E-mail: him@hongo.wide.ad.jp
c) E-mail: kensuke@nii.ac.jp

DOI: 10.1587/transcom.E93.B.328

distributed on various hosts generating small amount of dispersed traffic (e.g., worms, DDoS).

A few image-based approaches have been proposed for anomaly detection. Kim and Reddy [9] introduced a way to represent the traffic as a movie and used a scene-change algorithm to detect significant changes in the traffic. This method uses image-processing techniques; it can identify anomalies altering the traffic volume, and it has a short latency of detection. However, the design of frames is mainly based on packet counters and this restricts it being able to detect only those anomalies generating a large number of packets.

Similar problems arise for anomaly-based intrusion detection systems [5]. These systems are designed for network-edge analysis, and they are usually based on clustering techniques applied to packet header and payload data. However, they are unsuitable for backbone traffic because of their computation time and the lack of payload.

3. Temporal and Spatial Behavior of Anomalous Traffic

Here, we focus on how to highlight anomalies through their unusual uses of network traffic features during a period of time. We consider four traffic features — source address, destination address, source port, and destination port — and demonstrate that anomalous traffic may be manifested by some of them having abnormal distributions. By mapping traffic into a 2-D space (one feature and time), anomalies can be intuitively identified as lines.

Figure 1 shows two scatter plots generated from the same traffic trace taken at a trans-Pacific link (MAWI Samplepoint-F 2007/01/09) [10]; the horizontal axes stand for time, while the vertical axes represent the source port space in the upper sub-figure and the destination port space in the lower one. The intensity of the plots indicates the amount of packets. The apparent "lines" represent excessive uses of traffic features; traffic is either concentrated on a specific instance of a feature (horizontal line), or dispersed on numerous instances (vertical and diagonal line). The angle of diagonal lines acquaints the propagation speed of traffic within the feature space observed.

For example the two "lines" labeled (a) in the upper panel clearly stand for malicious traffic since all source port numbers are used in only 14 minutes. Manual inspection reveals that it is only SYN packets initiated from the same source address and directed to a few destination addresses on port 443 (HTTP over SSL). This is a typical behavior of an attack against a protocol of the Microsoft SSL library. The other slanted "lines" are the same kinds of attack mounted against other services. In particular, label (b) in Fig. 1 corresponds to a DDoS attack against a few HTTP servers (SYN packets). Because the displayed traffic is bidirectional, we can see "lines" similar in the bottom scatter plot (b') representing the acknowledgments sent from the servers to the aggressors (SYN-ACK packets). Also, two kinds of "lines" are repeated several times (see labels (c) and (d) in Fig. 1); these are ACK floods from two distinct hosts against different targets. The horizontal "lines" are anomalies consuming bandwidth, such as DoS attacks, misconfigurations or heavy-hitters.

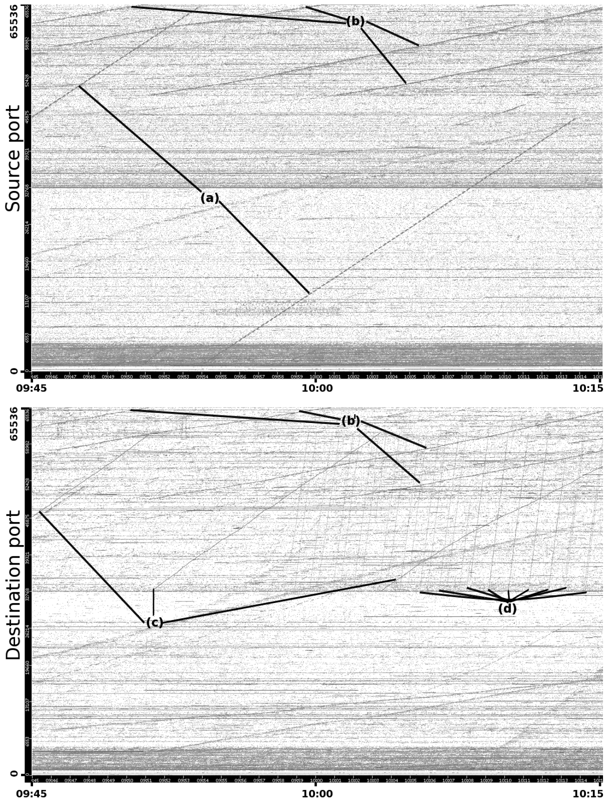

Fig. 1 Scatter plots of trans-Pacific traffic data. Source port vs. time (top) and destination port vs. time (bottom).

4. Anomaly Detection Method

On the basis of observations presented in Sect. 3, we devised an anomaly detection method based on pattern recognition [4]. The key idea underlying this approach is that traffic is monitored in pictures in which anomalous behaviors are displayed as "lines" that can be easily identified with a line detector.

This approach inspects only IP addresses and port numbers and requires no knowledge of the port numbers (such as application-related information). The method does not examine the packet payload, and its low computational time allows on-line detection.

4.1 Algorithm

The pattern-recognition-based method is outlined as Algorithm 1. The core of the detection process consists of three steps (see [4] for details):

(1) Computation of pictures (lines 4-6)

Four picture categories are considered to emphasize anomalous traffic ($f = 4$); all of them have time on the x axis and

Algorithm 1 Anomaly detection and classification

```
1:  $f$ is the number of traffic features considered
2:  Set the sliding window at the beginning of the data
3:  while window != EOF do
4:      for all packets in the window do
5:          Plot packet in $f$ pictures and store header information
6:      end for
7:      for all pictures do
8:          Compute the Hough space for the picture
9:          Extract lines from the Hough space
10:         for all lines found do
11:             New event $e$
12:             Retrieve all packet header from the line
13:             $e \leftarrow$ Summarize traffic features from packet headers
14:             if $\exists$ anomaly $a$ with main features = main features of $e$ then
15:                 Add $e$ to $a$
16:             else
17:                 Create a new anomaly
18:             end if
19:         end for
20:     end for
21:     Slide the window
22: end while
```

a different traffic feature on the y axis (source/destination address or port).

In order to reduce the IP address space and the port number space to match the size of pictures, we implemented two mechanisms. (1) Let say A is an IP address represented on 32 bits. v is the mapped value defined as $v = A \ mod \ 2^{\alpha}$ ($\alpha = 13$). Thus, A is mapped to a value in 2^{13} space. We divide this space into 16 pictures (512 pixels high) to improve the accuracy of the Hough transform. (2) Port numbers are directly aggregated into 16 pictures; in this case a pixel represents $2^{16}/16 * 512 = 8$ ports. All these values have been selected empirically and permit a low traffic aggregation not altering detection performance.

(2) Detection: Hough transform (lines 8-9)

Our method is based on the Hough transform [11] to detect lines in pictures. This technique has been frequently used in image analysis, and its basic form allows one to discover lines in a picture. We point out two important assets of the Hough transform: (1) It allows imperfect instances of objects to be detected; in our case, it can identify lines with missing parts (e.g., dotted lines). Consequently, anomalies interrupted by network or process latencies and displayed as segmented lines are also detected. (2) It is robust against noise; it can detect anomalies surrounded by legitimate traffic that appear as noise on the analyzed pictures.

The Hough transform consists of a voting procedure, where each plotted point (x, y) of a picture elects lines that can pass through its position. It enumerates all ρ and θ solving the equation of a line in polar coordinates: $\rho = x \cdot \cos\theta + y \cdot \sin\theta$. All votes are collected in an array called a Hough space, and all candidate lines are determined as the maximum values in this array. We distinguish two ways to sum votes: all votes are equal so that the values of the Hough space increase linearly, or votes increase proportionally to the current accumulated values (exponential growth). In the former case, long and short lines are handled equally, whereas, in the latter case, the Hough transform privileges longer lines and avoids false detections.

The peaks in the Hough space are extracted with a threshold relative to the average value of accumulated votes. Naturally, in the case of a linear vote, the choice of threshold can be an involved task. We discuss the role of these parameters in section 4.3.

(3) Identification (lines 10-19)

For each line extracted by the Hough transform, the initial data are recovered from all plots involved. Packet information is summarized as a set of statistics called *events*. An *event* constitutes a report for a specific line in a picture. Anomalies are monitored by more than one line and cause several *events*. That is, *events* from the same address source or aimed at the same address destination are grouped together to form an *anomaly*. Since anomalies usually raise several *events*, single *events* are ignored to reduce the number of false-positive alarms. This heuristic is a trade-off between false-positive and false-negative alarms. It permits to avoid about 50% of false-positive alarms, but decrease the number of true-positive alarms by about 20%.

4.2 Computational Complexity

The computational complexity of our method is mainly the one of the Hough transform performed on all pictures. In our experiments, we implemented the standard Hough transform which have a computation complexity linear to the number of plots in picture. In the worst case, each plot represents a single packet, and the number of plots in a picture category is equal to the total number of packets N. Let f be the number of picture categories, p the number of pictures for each picture category, t the traffic duration divided by the time bin, and $n_{i,j,k}$ the number of plots in the picture k of category i at the time bin j. The cost of Algorithm 1 in the worst case is linear and specified as:

$$\sum_{i=1}^{f} \sum_{j=1}^{t} \sum_{k=1}^{p} O(n_{i,j,k}) = \sum_{i=1}^{f} O(N) = f \cdot O(N)$$

4.3 Parameter Space

The performance of an anomaly detector strongly depends on the tuning of its parameters. In practice, satisfactory values are obtained by finding the best false-positive/false-negative trade-off through several tests run on well-known traffic traces. However, these values may not be suited for traffic with different properties. A relationship between parameter values and traffic characteristics is difficult to establish; thus selecting optimal parameters a priori is a challenge faced by every researchers. Automatic and dynamic tuning are still open problems.

This section pays close attention to the most significant parameters, namely the Hough transform parameters and the time bin, and evaluates their role in detecting anomalies in real Internet traffic.

The MAWI archive [10] contains traffic traces collected

Table 1 Heuristics. (based on [12])

Category	Label	Details
Attack	Sasser	Traffic on ports 1023/tcp, 5554/tcp or 9898/tcp
Attack	RPC	Traffic on port 135/tcp
Attack	Ping	High ICMP traffic
Attack	Other attacks	Traffic with more than 50% of SYN, RST or FIN flag. And http, ftp, ssh, or dns traffic with more than 30% of flag SYN
Attack	NetBIOS	Traffic on ports 137/udp or 139/tcp
Special	Http	Traffic on ports 80/tcp and 8080/tcp with less than 30% of SYN flag
Special	dns, ftp, ssh	Traffic on ports 20/tcp, 21/tcp, 22/tcp or 53/tcp&udp with less than 30% of SYN flag
Unknown	Unknown	Traffic which does not match other heuristics

by the WIDE Project since 1999. We analyzed three sets of traces, from this archive, taken from a trans-Pacific link between Japan and United States. Two sets were collected from samplepoint-B (a 18-Mbps Committed Access Rate on a 100 Mbps link) over the course of one week in 2004/08 and one week in 2005/08, and one set was collected from samplepoint-F (a full 100 Mbps link) over the course of one week in 2006/08. The throughput at samplepoint-B was increasing during this period, and the data taken in 2004 and 2005 showed minor differences in volume. Moreover, samplepoint-B was replaced by samplepoint-F in July 2006, and this considerably increased the amount of data transmitted.

Simple heuristics helped us to evaluate the amount of anomalous traffic identified by our method. These heuristics were deduced from known attacks that occurred during the period of time analyzed and improper uses of TCP flags. Table 1 lists them in the same order as executed; the first five categorize traffic as “sure attacks,” and the last three categorize “suspected” traffic (meaning that either more inspection is needed, or it is a false-positive alarm). The quality of detection is measured as the ratio of “suspected” anomalies over the total number of anomalies reported (a lower ratio is better, see Fig. 2).

4.3.1 Hough Parameters

During the voting procedure of the Hough transform, a vote for a line l is defined by a function of the form w^x, where x is the current number of votes for l, and w is a constant value named *weight*. A relative threshold is used to extract the detected lines in the Hough space.

The weight and threshold are the principal parameters of our method. To evaluate their impact on the anomaly detection, we executed our detection method on three data sets and changed the weights and threshold (other parameters were fixed). This analysis confirmed our expectations, that is: (1) Large weights ($w > 1$) help to highlight well-marked lines, whereas, $w = 1$ permits small lines to be elected. (2) The threshold is significant only when $w = 1$. Using the heuristics of Table 1 we deduced that the detection method performed better inspections on every trace analyzed with $w = 1.6$ (all thresholds tested led to similar results). The left graph in Fig. 2 displays the average result for data during a week in August 2004. The two other data sets have provided similar results; hence, we concluded that this parameter is robust to throughput variances.

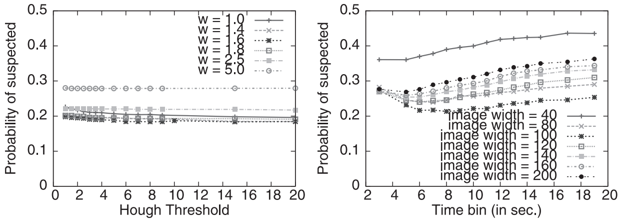

Fig. 2 Evaluation of parameters with traces from 2004/08. For the left figure the image width is set to 100 and the time bin is set to 6 seconds. For the right figure the weight is set to 1.6 and the threshold is set to 10.

4.3.2 Image Size and Time Bin

The manner of mapping traffic in a 2-D space is a key feature of our method; however, setting the proper resolution (pixel/second) of pictures is not intuitive.

Numerous tests on the three sets of traffic traces (the right graph of Fig. 2 shows the tests proceed on the traffic traces taken in 2004/08) indicated that the optimal image width for most cases is 100 pixels, whereas the ideal time bin depends on the analyzed traffic. The appropriate time bin for traces taken in 2004 is around 6-8 seconds (see right graph of Fig. 2). A smaller time bin (around 6 seconds) was found to be best for data recorded in 2005, whereas 3 seconds was found to be best for data from samplepoint-F. The main differences in the three sets of traces are their throughput and link bandwidth; in particular, the set collected in 2006 has more than twice the traffic volume of the one from 2005. Consequently, for the same time bin, pictures representing the traffic taken in 2006 might plot two times more points than those standing for the traffic from 2005 (depending on the traffic distribution). The Hough transform works properly only if enough points are plotted in the pictures and the pictures are not saturated.

To maintain a certain quantity of data displayed in pictures, the time bin was selected in accordance with the measured traffic rate of the observed traffic. In order to avoid tiny time bin while dealing with high throughput our method can be used in combination with a traffic aggregation method (e.g., sketch).

5. Evaluation

The evaluation of a detection method is an important step in validating its effectiveness; however, the lack of a common database with real backbone traffic and labeled anomalies raises a complicated issue. In Internet research community, the evaluation of an anomaly detection technique

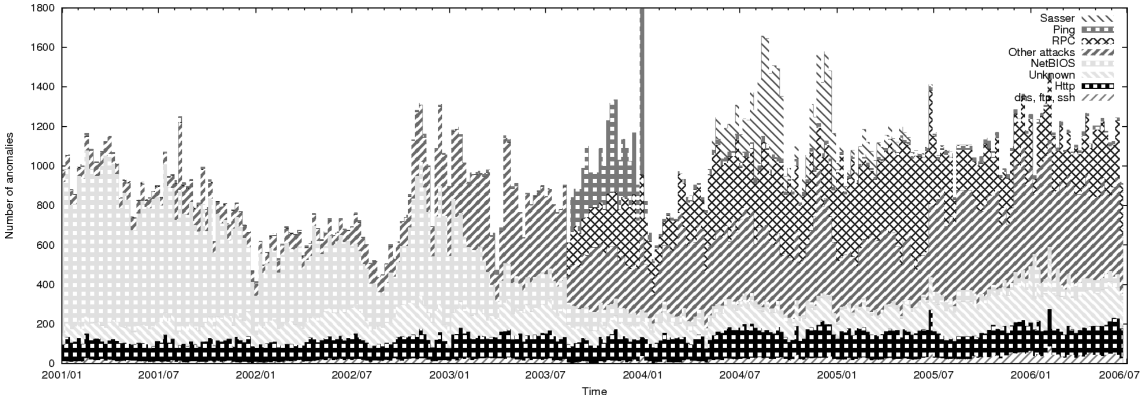

Fig. 3 Average number of anomalies per week reported by our method on all traffic traces collected on the MAWI samplepoint B from 01/01/2001 to 30/06/2006.

usually consists in one of the following processes: (1) Comparison of anomalies reported by a few different approaches [9]. (2) Analysis of real data and manual estimation of the number of false-positives reported [1]–[3]. (3) Injection of malicious traffic into traces supposed to be anomaly-free and computation of false-positive/false-negative rates [3].

We used the processes (1) and (2) to evaluate our detection method in realistic conditions. In Sect. 5.1 we identify anomalies in a large data set and carefully inspect the results. In Sect. 5.2 we compare the anomalies detected by our method with those identified by a method based on gamma modeling and a method based on K-means.

5.1 Anomalies of MAWI Database for 6 Years

We analyzed all traces of the MAWI database collected at samplepoint-B from 01/2001 to 06/2006; each trace represents 15 minutes of traffic with anonymized IP addresses. The same data set has been dissected in [12], to show the detailed evolution of the traffic as well as an application breakdown. Although [12] did not aim at labeling anomalies in MAWI systematically, it does mention several prominent anomalies that significantly altered the traffic. For example, a major ping flood occurred on 2003/08-12, and outbreaks of the Sasser worm were identified in 2004/08, 2004/12 and 2005/03.

5.1.1 Results

We used our method to analyze all traces collected from 2001 to 2006. The traces were processed with same parameters (weight=1.6, time bin=8 seconds, image width=100 pixels and threshold=10). Since the weight was set to 1.6 the threshold has been arbitrary chosen (see Sect. 4.3.1). Figure 3 summarizes the results and classifies them by the heuristics in Table 1. This graph plots the number of anomalies, whereby an anomaly is as described in Sect. 4.1 (namely as a set of grouped events with respect to their sources or destinations IP).

The large anomalies noticed in [12] can be observed in Fig. 3; the ping flood appears from 2003/08 to 2004/01, and the three Sasser outbreaks are represented as three peaks between 2004/05 and 2005/06. Our method also identified important activity on port 135 starting in 2003/08 and lasting several years (labeled RPC Fig. 3). This traffic also appears in the application breakdown of [12], and it has been attributed to MS vulnerabilities. Our manual inspection revealed that this anomalous traffic was initiated by a large outbreak of the Blaster worm (also known as MS-Blast/Lovsan) spreading through an exploit in the Remote Procedure Call (RPC) protocol of almost all versions of Windows at this time. Security holes in RPC have been frequently reported since then, and this protocol is still a common medium for various attacks.

Mainly NetBIOS traffic was reported from January 2001 to August 2002. We deduced from our manual verification that most flows contained a tiny number of packets with both the port source and destination set to 137/udp. This traffic is a manifestation of the normal behavior of the name resolution service implemented in the Windows networking shares (even though this mechanism is designed for local networks). We concluded that the traffic was principally failed name resolution requests initiated by a large number of distinct hosts and aimed at numerous destinations. We noticed that most of the sources and destinations of the identified flows did not have other network activity, and their bandwidth consumption was really low. In addition, the average number of packets observed in the analyzed backbone link steadily increased during the six years. Our detection method identified this category of traffic in 2001 and 2002 because of the fixed parameters it employed for the analysis. This means that not enough points were displayed in the pictures to compute the Hough transform properly. Although malicious behavior is not evident, these anomalies still reflect a misuse of the NetBIOS protocol.

However, from 2002/10 onwards, the distribution of NetBIOS traffic completely changed and clearly indicated malicious behavior. Indeed, we observed that various hosts were probing entire sub-networks to take advantage of the security flaws of the Windows file sharing mechanism, and several viruses were released during the same period (e.g., Opaserv, Bugbear).

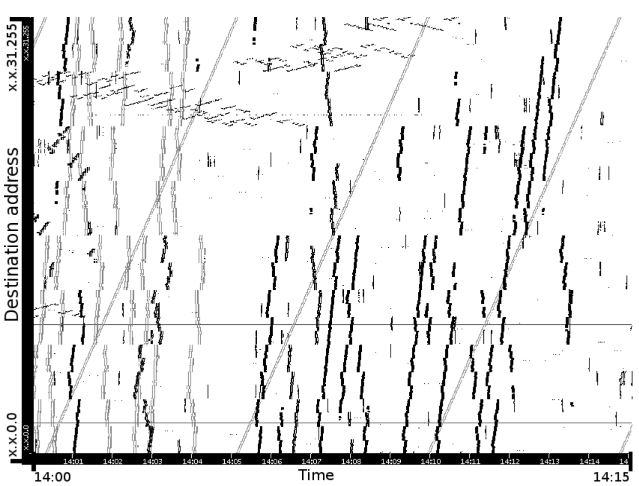

Fig. 4 Several examples of anomalies detected in one traffic trace of 15 minutes (2004/10/14). Two horizontal lines: 8000/udp (iRDMI). On the left light-colored: 5900/tcp (VNC) Three long slanted lines: 445/tcp (MS Service) Black: 1023/tcp, 5554/tcp and 9898/tcp (Sasser).

Other attacks were mainly related to the NetBIOS protocol, but the heuristics classified these as due to a high rate of SYN flags (on port 139/tcp).

This analysis of the MAWI database exposed large-scale attacks, and it demonstrates our methods ability to identify numerous anomalies. However, quantitative observations conducted over a long period (for 6 years) naturally omit occasional anomalies. Hence, the next section discusses anomalies detected in a single day.

5.1.2 Examples of Anomalies Detected in the Same Day

Figure 4 illustrates several examples of anomalies detected in the same day; legitimate traffic and other identified anomalies have been excluded for clarity.

The light-colored lines on the left side of Fig. 4 are generated by one host probing a large sub-network on port 5900 (VNC, a remote control application). The attack is aimed at 16^2 hosts of the same sub-network, but due to the routing policy, only half of them have been contacted via the analyzed link. Despite missing packets, the anomaly is still easily identifiable. The activity was initiated by only one source IP address, so the detection method reports it as a single anomaly.

The three long slanted lines stand for a similar behavior against a Windows service (port 445), whereas the two horizontal lines display abnormally high traffic between a couple of hosts on port 8000. These two long-lasting anomalies started before and stopped after the detection process, meaning that they could not be revealed by methods analyzing traffic volume. Our method had no difficulty in identifying them.

The traffic on this day is flanked by two significant outbreaks of the Sasser worm. Sasser activity is shown in black in Fig. 4, and two different propagations of the worm are shown. On the one hand, long vertical lines, depicting a large and quick spread, appear on the whole picture. On the other hand, the small slanted lines at the top of the figure show a slowly spreading worm. These two observations illustrate either two variants of the worm or the network/process latency effect on the worm spread.

5.2 Cross-Validation

We compared results of our method with those of two other methods. One consists of random projection techniques (sketches) and multiresolution gamma modeling [2]. Hereafter we call it as the gamma-based method. The traffic is split into sketches and modeled using Gamma laws, and anomalous traffic is reported by using the statistical distance from the average behavior. The other method is a distance-based outlier detection method using K-means [5]. The traffic is clustered with K-means regarding 14 traffic features and outliers are reported depending on their density and distance to other clusters.

5.2.1 Methodology

The three methods were tested on several trans-Pacific traces captured during August 2004. A great deal of anomalous network activity concerning the Sasser worm was reported during this time. Analysis of each data set leads to similar conclusions, so we only present the results for one traffic trace (2004/08/01). We tuned all methods until they report approximately the same number of alarms. The alarms are reported differently by these methods, so we checked whether an alarm reported by one method had also been detected by the others, and vice-versa.

5.2.2 Results

The gamma-based method was executed with the values of 0.8 for the alpha parameter and 500 for the threshold and it reported 1083 alarms. K-means was computed with 100 clusters and it reported 917 alarms. Our method was run with a time bin of 10 s, $w = 1.6$ and it reported 1063 alarms. For a 15-minutes trace with a mean throughput of 20.77 Mbps, 6 591 957 packets, and 614 324 different IP addresses (57 862 source addresses), the execution time of our method was about 3.5 minutes on a standard desktop PC (Core2Duo 2.6 GHz, 2 GB of RAM). Table 2 shows these alarms classified by using the heuristics of Table 1.

We checked if the alarms reported by our algorithm had also been reported by the gamma-based method. We inspected all alarms not reported by either method and noticed that the 574 (854 − 280) alarms labeled as *ATTACK* were true-positive alarms related to worms (Sasser and Blaster) or scan activity (mainly on NetBIOS). Our method detected twice as much anomalous traffic for this class of anomaly than the statistical one did. Several of these anomalies could not be detected with the gamma-based method because of the small number of packets involved (< 500 packets). However, the 24 (130 − 106) and 27 (79 − 52) alarms labeled as *SPECIAL* and *UNKNOWN* reported by our method

Table 2 Alarms reported by the Hough-transform-based (HT), gamma-based (G), and K-means-based (KM) methods.

	HT	G	KM	HT&G	HT&KM	G&KM
Attack	854	323	306	280	75	50
Special	130	517	488	106	23	75
Unknown	79	243	123	52	49	26
Total	1063	1083	917	438	147	151

but not by the gamma-based one were heavy traffic between two hosts using HTTP, HTTPS, or peer-to-peer protocols. Although the traffic in most of these cases seemed to be harmless elephants, their packet payloads would have to be checked to conclude if they were indeed false-positives alarms.

The gamma-based method reported 1083 alarms; 579 (1083 − 438) of these were not detected by our method. Of these 579 alarms, 375 were labeled as *SPECIAL*, and 161 were classified as *UNKNOWN*. We deduced from a manual inspection that most of them were heavy traffic with uncommon properties using http or peer-to-peer protocols; we were not able to determine if they were false-positive alarms without payload. However, our method missed 43 (323 − 280) events reported by the gamma-based method and labeled as *ATTACK*; 21 of them represents worms (mainly Sasser) and 11 stand for PING flooding.

The K-means-based method identified 917 alarms; 770 (917−147) of these were not detected by our method. 439 of these 770 were labeled as *SPECIAL*, and 100 were classified as *UNKNOWN*. Manual inspection has shown that they were mainly harmless traffic with uncommon properties. Only 75 alarms labeled as ATTACK were reported by both the K-means-based and our method. The 231 (306 − 75) alarms labeled as ATTACK only reported by the K-means-based method are mainly flows with a high percentage of TCP flags set to SYN, FIN or RST. Although these events are mainly true-positive alarms missed by our method, we had difficulty in determining the threat posed by 116 of them where the number of packets send by a suspicious host is really low (≤ 10).

In order to validate the sufficiency of the heuristics of Table 1, we inspected the 445 (79 + 243 + 123) alarms labeled as *UNKNOWN* reported by the three methods. 411 are considered as peer-to-peer traffic because using both higher ports. The rest of them are usual traffic, RSYNC (10), NNTP (6), POP3 (5), RTP (4), etc.

5.2.3 Discussion

The proposed method has reported a large number of alarms labeled as *ATTACK* not detected by other methods, indicating that our method has a high probability of reporting true-positive alarms compare to others. However, our method still missed 249 (231 + 43 − 25(double counted)) *ATTACK* alarms (false-negative) because it does not take TCP flag into account and due to the absence of port number in ICMP protocol. Considering the 116 suspicious *ATTACK* alarms reported by K-means (i.e., host sending less than 10 packets), the detection ratio (true-positive rate) of our method is about 77 ∼ 87%.

Many alarms labeled as *UNKNOWN* and *SPECIAL* have been reported by the gamma-based and K-means-based methods. Although these alarms could be true positives misclassified by the heuristics, our manual inspection revealed that they were false-positives alarms. Also, our method reported only 209 (130 + 79) false-positive alarms over the 56759 benign source IP (reported by none of the three detection methods as ATTACK). These observations show the low false-positive rate (0.3%) of the proposed method.

Furthermore, we have manually observed 426 source addresses related to the Sasser activity, 84 (19%) have been identified by the K-means-based method, 156 (36%) by the method based on gamma modeling, and 321 (75%) by our method.

We deduced from Table 2 that even though our method and the gamma-based one are quite different, they had almost 50% of their results in common. Our method detected two times more traffic related to worms and scan activity than the gamma-based method did. This category of anomaly is characterized by small flows and its reflects the fundamental weakness of statistical methods. By analyzing TCP flags, the K-means-based method could detect several anomalous traffics not reported by other methods. However, this method is designed to identify outliers and since the Sasser activity has been dominating analyzed traffic, it failed in detecting such traffic and reported many false-positive alarms. Also, the K-means-based method does not scale to backbone traffic because of its computation time. The three methods have distinct weaknesses and advantages; hence, they would be a good combination.

6. Conclusion

We illustrated the characteristic shapes of anomalous traffic in time and space and presented an approach to anomaly detection based on pattern recognition. This method takes advantage of a graphical representation to reduce the dimensions of network traffic and the techniques of image analysis. Only header information is required; no inspection of the packet payload and no prior information about the traffic or port numbers are needed. We conducted a detailed evaluation of our method by analyzing the principal parameters and by validating it on actual Internet traffic. The analysis of traffic from a trans-Pacific link revealed that our method can identify various anomalies (e.g., worms and network/port scans), and mice anomalous flows.

The comparison of our method with a gamma-based method and a K-means-based method indicates that the three approaches identified distinct classes of anomalies. Therefore, their use in combination would have a synergistic effect.

Statistics-based and clustering-based methods have obtained a certain maturity, whereas pattern recognition is a promising novelty in anomaly detection. Hence, we believe that one important future project is to apply the various pattern recognition tools of image analysis. Various graphical representations can be designed to better highlight anomalies and their capabilities of processing sampled data could then be evaluated. Precisely defining the relation between the amount of traffic and the time bin would permit our method to be automatically tuned. Also, the combination of our method with a traffic aggregation technique could increase the amount of traffic handle by our method and improve the computation time.

References

[1] P. Barford, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies," IMW'02, pp.71–82, 2002.

[2] G. Dewaele, K. Fukuda, P. Borgnat, P. Abry, and K. Cho, "Extracting hidden anomalies using sketch and non gaussian multiresolution statistical detection procedures," SIGCOMM LSAD'07, pp.145–152, 2007.

[3] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," SIGCOMM'05, pp.217–228, 2005.

[4] R. Fontugne, T. Hirotsu, and K. Fukuda, "An image processing approach to traffic anomaly detection," AINTEC'08, pp.17–26, 2008.

[5] R. Sadoddin and A.A. Ghorbani, "A comparative study of unsupervised machine learning and data mining techniques for intrusion detection," MLDM'07, pp.404–418, 2007.

[6] H. chul Kim, K. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee, "Internet traffic classification demystified: Myths, caveats, and the best practices," CoNEXT'08, 2008.

[7] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "Blinc: Multilevel traffic classification in the dark," SIGCOMM'05, vol.35, no.4, pp.229–240, 2005.

[8] X. Li, F. Bian, M. Crovella, C. Diot, R. Govindan, G. Iannaccone, and A. Lakhina, "Detection and identification of network anomalies using sketch subspaces," SIGCOMM'06, pp.147–152, 2006.

[9] S.S. Kim and A.L.N. Reddy, "A study of analyzing network traffic as images in real-time," INFOCOM'05, pp.2056–2067, 2005.

[10] K. Cho, K. Mitsuya, and A. Kato, "Traffic data repository at the WIDE project," USENIX 2000 Annual Technical Conference: FREENIX Track, pp.263–270, June 2000.

[11] R.O. Duda and P.E. Hart, "Use of the hough transformation to detect lines and curves in pictures," Commun. ACM, vol.15, no.1, pp.11–15, 1972.

[12] P. Borgnat, G. Dewaele, K. Fukuda, P. Abry, and K. Cho, "Seven years and one day: Sketching the evolution of internet traffic," INFOCOM'09, 2009.

Romain Fontugne received his master's degree in Computer Science from Joseph Fourier University, France, in 2008. He is now a Ph.D. candidate in Department of Informatics, the Graduate University for Advanced Studies (SOKENDAI). His research interests are Internet traffic analysis and Internet security.

Yosuke Himura is a master course student in Department of Information and Communication Engineering, the University of Tokyo. His research interests are Internet traffic analysis and Internet security.

Kensuke Fukuda is an associate professor at the National Institute of Informatics and SOKENDAI from 2006. He received his Ph.D. degree in computer science from Keio University at 1999. He worked in NTT labs. from 1999 to 2005. In 2002, he was a visiting scholar at Boston University. His current research interest is internetworking. He is also a researcher of PRESTO JST.