International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:3, No:12, 2009

A New Traffic Pattern Matching for DDoS Traceback Using Independent Component Analysis

Yuji Waizumi, Tohru Sato, and Yoshiaki Nemoto
Graduate School of Information Sciences Tohoku University
Sendai–shi, Miyagi, 980–8579 Japan
Email: wai@ecei.tohoku.ac.jp

Abstract—Recently, Denial of Service(DoS) attacks and Distributed DoS(DDoS) attacks which are stronger form of DoS attacks from plural hosts have become security threats on the Internet. It is important to identify the attack source and to block attack traffic as one of the measures against these attacks. In general, it is difficult to identify them because information about the attack source is falsified. Therefore a method of identifying the attack source by tracing the route of the attack traffic is necessary.

A traceback method which uses traffic patterns, using changes in the number of packets over time as criteria for the attack traceback has been proposed. The traceback method using the traffic patterns can trace the attack by matching the shapes of input traffic patterns and the shape of output traffic pattern observed at a network branch point such as a router. The traffic pattern is a shapes of traffic and unfalsifiable information. The proposed trace methods proposed till date cannot obtain enough tracing accuracy, because they directly use traffic patterns which are influenced by non-attack traffics. In this paper, a new traffic pattern matching method using Independent Component Analysis(ICA) is proposed.

Keywords—Distributed Denial of Service, Independent Component Analysis, Traffic pattern

I. Introduction

During recent years, Denial of Service(DoS) attack in which a large number of packets are sent to the target host and Distributed DoS(DDoS) attacks which are DoS attacks carried out from plural hosts so as to scale up the level of the attack, have become social problem. Several enormous damages have been caused by them on macro-scale business sites and root nameservers[1]. This fact indicates that even well-managed networks, in which administrators pay close attention to network attacks for their security, are not fully secure against DoS attacks yet. In order to defend the network from DoS attack, techniques to identify the attackers have become very important as a deterrent against DoS attacks[2]. But, it is difficult to detect attack hosts because the information about the DoS attack hosts is generally falsified. This is called IP Spoofing. Therefore, the method of finding out the attacker by tracing back the path in which the attack traffic traversed, is necessary.

A lot of traceback methods which are not influenced by IP Spoofing have already been proposed. Traceback methods [3], [4], [5], [6] require intermediate routers to write packet routing information into the packet header, or, to generate new packets called ”Traceback Message” to a victim. With sufficient number of these packets from many routers, the traffic source and path can be identified. However, it is pointed out that the attacker can disturb traceback by sending falsified routing information to the victims of these methods[5].

Traceback method [7] records all forwarded packets into intermediate routers. When tracing DoS attacks, routers with records of attack packets are traced back. But, it is pointed out in [8], [9] that there is a possibility of large overhead when routers are equipped with this function due to the large processing load. Moreover, large capacity storage devices are needed when the amount of traffic is large[6].

On the other hand, some IP traceback methods for DoS attacks based on the similarity of shapes of the time-varying number of packets (traffic pattern) have already been proposed [10], [11]. These traceback methods based on the traffic pattern, match input traffic patterns, which flow into a network branch point such as router, and a output traffic pattern, which is flowing out from the branch point, directly. Then the method selects the link traversed by the DoS attack traffic according to the shape similarity of the input and the output traffic patterns. A feature of these method is the use of the changes in packet number over time which cannot be falsified on the attacker side. Moreover, it is considered that these methods have advantage on processing the volume of information over above-mentioned methods, because only counting the number of packets is needed. But it is apparent through simulation that, methods directly using the pattern shapes as the criterion for tracing, can not accurately trace when many attacks confluence and make the pattern shape complicated [11].

In this paper, the reasons behind the difficulty in accurate tracing in the conventional methods that use traffic patterns for analysis is discussed. Then a method to analyse traffic patterns by Independent Component Analysis(ICA) is proposed, and it is shown through simulation that this method can carry out tracing with higher accuracy even when the input and output patterns are complicated.

II. Tracing DoS Attacks by Traffic Pattern Matching

A. Outline

Traffic-pattern-based tracing methods target DoS attacks in particular, which are difficult to defend because they are carried out with normal protocols. The Flood types of DoS are SYN Flood[12] which drives a target into overload ,UDP Flood[13] and ICMP Flood[14] which exhaust network resource. Common characteristic of these flood type DoS is

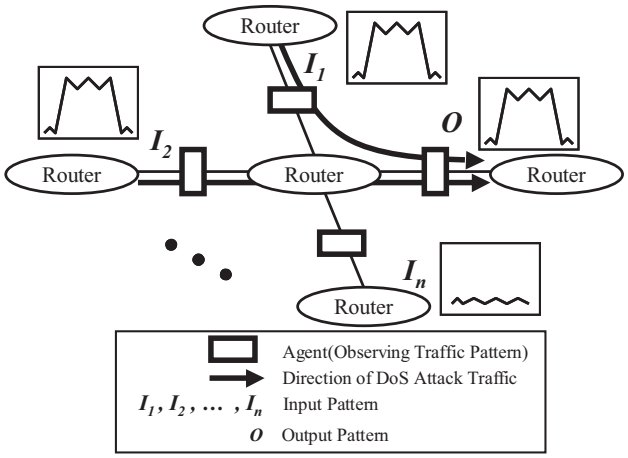


Fig. 1. I/O model of network branch point at DDoS attacks

the transmission of large amount of packets for a certain period. In the tracing method using traffic pattern, agents which always observe the number of packets and preserve them, are located in each link connected with a branch point like a router on a network as shown in Figure 1. When one agent detect an abnormal increase in the number of packets, the agent acquires traffic patterns from other agents. Acquired pattern is treated as time-series data, and when the pattern is composed of m time slot data, it is shown as m dimension vector such as $\mathbf{O} = (\mathbf{O_1}, \mathbf{O_2}, \cdots, \mathbf{O_m})$. And, when tracing DoS attacks, acquired output pattern $\boldsymbol{O}$ and each input pattern $\boldsymbol{I_i}(i = 1, 2, \cdots, n)$ are compared, and relations of each $\boldsymbol{I_i}$ to $\boldsymbol{O}$ are judged. The input links with which the output has high relations are judged to be attack routes.

DoS detecting methods which use a simple threshold for the number of packets, and a detecting method using the Auto-Regressive model(AR-model) which learns from past number of packets and detects DoS attacks based on the number of packets which is forecasted as normal state by the AR-model, have already been proposed[10], [15]. In this paper, it is assumed that the DoS attack can be detected by these methods and the agents can obtain the traffic patterns which are used for comparing their shapes.

In the rest of this section, the outline of each algorithm that compares pattern shape in conventional tracing methods using traffic patterns is described, and the problem of each algorithm is verified.

B. Pattern Shape Comparison Algorithm Based on Correlation Coefficient and Related Issues

The pattern shape comparison algorithm proposed in reference [10] calculates the correlation coefficient between output pattern $\boldsymbol{O}$ and each input pattern $\boldsymbol{I_i}(i = 1, 2, \cdots, n)$. Since the correlation between output pattern $\boldsymbol{O}$ and each input pattern $\boldsymbol{I}$ that takes part in DDoS attack tends to strengthen, the input pattern with the correlation coefficient which is higher than 0.7, is judged as having a relation to DDoS attack. Note that the correlation is strong in general. And, the link that the input

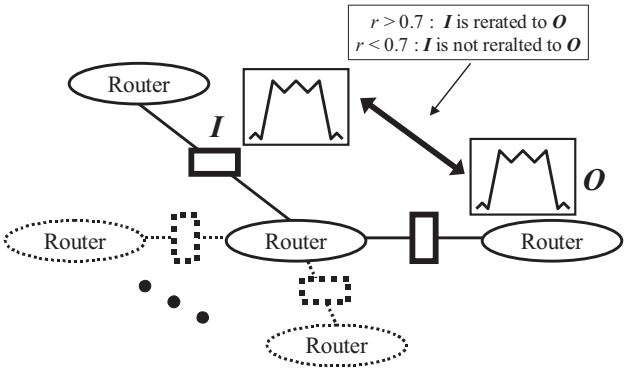


Fig. 2. Outline of tracing method by correlation coefficient

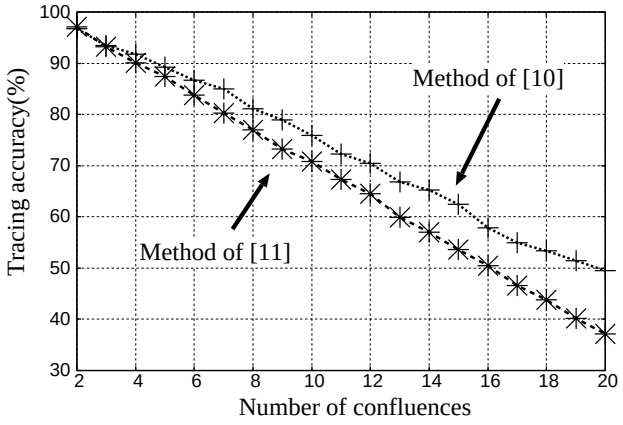


Fig. 3. Traceback success rate of conventional comparison algorithms

pattern passed is traced as an attack route. The correlation coefficient r between output pattern $\boldsymbol{O} = (O_1, O_2, \cdots, O_m)$ and a certain input pattern $\boldsymbol{I} = (I_1, I_2, \cdots, I_m)$ like Figure 2 is calculated as follows:

$$r = \frac{\frac{1}{m}\sum_{i=1}^{m}(I_i - \overline{I})(O_i - \overline{O})}{\sigma_I \sigma_O} \quad (1)$$

where $\overline{I}$ and $\overline{O}$ are averages of data in each pattern, σ_O and σ_I are standard deviations respectively.

However, when plural attack patterns are joined as DDoS attacks, the output pattern shape is complicated. Therefore, the correlation between output pattern and each input patterns which take part in the DDoS attack weaken, and the coefficient becomes small. As a result, tracing all DDoS attack routes is difficult. Figure 3 shows traceback success rate of the comparison algorithm based on correlation coefficient obtained through traceback simulation similar to [11].

Thus, the pattern shape comparison algorithm that uses a correlation coefficient has a problem of decreasing tracing accuracy when the number of attack confluences increase.

C. Pattern Shape Comparison Algorithm Based on Quadratic Programming and Related Issues

The pattern shape comparison algorithm proposed in reference [11] assumes that the majority of output pattern of DDoS attack is composed of certain percentage of each input patterns that take part in the DDoS attack. The relation of I/O traffic is formulated as a quadratic programming problem. The rate of each input pattern included in the output pattern (Contribution Rate) is guessed by solving this quadratic programming problem. Input pattern with high Contribution Rate is judged as taking part in DDoS attack, and routes at which these patterns are observed is traced as attack routes.

Contribution Rate $a_i(0 \leq a_i \leq 1)$ shows the rate where input $\boldsymbol{I_i}(i = 1, 2, \cdots, n)$ flows to output $\boldsymbol{O}$, the relation between each input $\boldsymbol{I_i}$ and output $\boldsymbol{O}$ is shown as follows:

$$\boldsymbol{O} = \sum_{i=1}^{n} a_i \boldsymbol{I_i} + \boldsymbol{x} \tag{2}$$

where $\boldsymbol{x}$ is the difference between presumed input $\sum_{i=1}^{n} a_i \boldsymbol{I_i}$ and output $\boldsymbol{O}$, and it is shown by m dimension vector of $\boldsymbol{x} = (x_1, x_2, \cdots, x_m)$.

Since DDoS attacks send a large amount of packets, unusually high volume of traffic is observed at links in the path of attack traffic. Moreover, most of this attack traffic flow to a specific output. Therefore major portion of the output $\boldsymbol{O}$ is composed of only input patterns that take part in the attack. Hence, difference $\boldsymbol{x}$ at DDoS attacks is considered to be significantly smaller than output $\boldsymbol{O}$. Therefore, Contribution Rate $a_i(0 \leq a_i \leq 1)$ that can be obtained when $\boldsymbol{x}$ becomes small enough under expression (2), is presumed as rate that each input $\boldsymbol{I_i}(i = 1, 2, \cdots, n)$ flows to output $\boldsymbol{O}$. This is a quadratic programming problem of the minimizing function (3) under the subject to condition (4).

$$\texttt{Minimize Function} \quad |\boldsymbol{x}|^2 = x_1^2 + x_2^2 + \cdots + x_m^2 \tag{3}$$

$$\texttt{Subject to Condition} \quad \boldsymbol{O} = \sum_{i=1}^{n} a_i \boldsymbol{I_i} + \boldsymbol{x} \quad (0 \leq a_i \leq 1) \tag{4}$$

The pattern, contribution rate a_i of which obtained by solving quadratic programming problem under expression (3) and (4) is 0.5 or more, or in other words, the input pattern that contributes 50% to output of the pattern $\boldsymbol{O}$ composition is judged as having a relation to DDoS attacks .

The accuracy rates of trace methods using quadratic programing and correlation coefficient decline when the number of confluent flows in an attack increase[11]. As the number of confluent flows in an attack rise, the occurrence possibility of instances, in which for minimum $|\boldsymbol{x}|$, contribution rate of attack traffic patterns become less than 0.5, is incremented. Figure 3 shows traceback success rate of comparison algorithm based on quadratic programming obtained through traceback simulation similar to [11].

From the above, either tracing method using traffic patterns have a common problem of tracing accuracy decrease when the number of attack confluences increases.

III. Proposal of Tracing Method Using Independent Component Analysis

Conventional tracing methods tend to be influenced by traffic not taking part in the attack included in each pattern. And, it is considered that tracing accuracy decreases because the influence grows as the number of attack confluences increases. In this section, a method of resolving traffic pattern to plural independent patterns by using Independent Component Analysis , and judging relation between each pattern and attack from analysis of the result is proposed. A better accuracy can be expected from the proposed method than conventional methods for judging the relation to the attack by comparing patterns directly.

A. Independent Component Analysis

Independent Component Analysis is a method for separating observation signals formed by linear mixing of plural source signals to plural independent signals. Now, n independent source signals are shown as $\boldsymbol{s} = (\boldsymbol{s_1}, \boldsymbol{s_2}, \cdots, \boldsymbol{s_n})$. And, Observed Signal $\boldsymbol{x} = (\boldsymbol{x_1}, \boldsymbol{x_2}, \cdots, \boldsymbol{x_n})$ is the mixture of Source Signal $\boldsymbol{s}$ mixed by Mixing Matrix $\boldsymbol{A}$ = $a_{ij}(i = 1, 2, \cdots, n, j = 1, 2, \cdots, n)$,as follows:

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s} \tag{5}$$

Then, Independent Component Analysis presumes Mixing Matrix $\boldsymbol{A}$ and Source Signal $\boldsymbol{s}$. In the case of a model like (5), the problem is to lead Separating Matrix $\boldsymbol{W}$ which makes each element of $\hat{\boldsymbol{s}}$ independent mutually based on Observed Signal $\boldsymbol{x}$ (6).

$$\hat{\boldsymbol{s}} = \boldsymbol{W}\boldsymbol{x} \tag{6}$$

In (6), $\hat{\boldsymbol{s}}$ is n dimension vector and a presumption value of $\boldsymbol{s}$, and $\boldsymbol{W}$ is $n \times n$ matrix. In (6), $\hat{\boldsymbol{s}}$ is n dimension vector and a presumption value of $\boldsymbol{s}$, and $\boldsymbol{W}$ is $n \times n$ matrix. In an ideal case, $\boldsymbol{A^{-1}} = \boldsymbol{W}$ holds true. But, because $\boldsymbol{A}$ is an unknown, $\boldsymbol{W}$ and $\boldsymbol{A^{-1}}$ are brought close by learning during actual calculations.

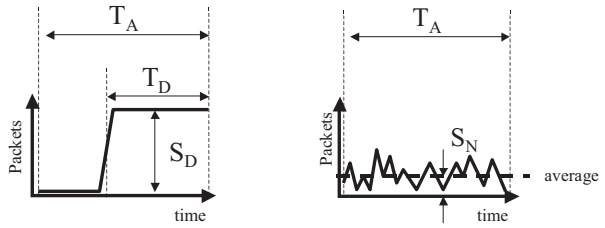
Fast ICA algorithm[16] with the advantage about convergence speed is used in this paper, though other algorithms of Independent Component Analysis also exist. First, Observed Signal $\boldsymbol{x}$ is preprocessed by "Whitening" and "Principal Component Analysis" to calculate independent component, and the presumption of an independent component is simplified. Next, Separating Matrix $\boldsymbol{W}$ is calculated from signal $\tilde{\boldsymbol{x}}$ obtained by preprocessing $\boldsymbol{x}$. In this step, a certain column vector $\boldsymbol{w}$ is decided at random. And, $\boldsymbol{w}$ is updated by (7) until it converges. Convergence means that the inner product between $\boldsymbol{w_{old}}$ before updating and $\boldsymbol{w_{new}}$ after updating is set to 1.

$$\boldsymbol{w_{new}} = E\{\tilde{\boldsymbol{x}}(\boldsymbol{w_{old}^T}\tilde{\boldsymbol{x}})^{\boldsymbol{3}}\} - 3\boldsymbol{w_{old}} \tag{7}$$

where $E\{\tilde{\boldsymbol{x}}(\boldsymbol{w_{old}^T}\tilde{\boldsymbol{x}})^{\boldsymbol{3}}\}$ is the covariance of $\tilde{\boldsymbol{x}}(\boldsymbol{w_{old}^T}\tilde{\boldsymbol{x}})^{\boldsymbol{3}}$, and $\boldsymbol{w^T}$ is one of the row vectors of $\boldsymbol{W}$. In order to make sure the obtained independent component $\boldsymbol{w_1^T}\tilde{\boldsymbol{x}}, \boldsymbol{w_2^T}\tilde{\boldsymbol{x}}, \cdots, \boldsymbol{w_n^T}\tilde{\boldsymbol{x}}$ does not fall into the same extreme value, orthogonal vector $\boldsymbol{w_i}$ to $\boldsymbol{w_1}, \boldsymbol{w_2}, \cdots, \boldsymbol{w_{i-1}}$ obtained previously, is calculated. The algorithm finishes at $i = n$. Independent component $\hat{\boldsymbol{s}}$ is calculated from $\boldsymbol{W}$ obtained finally.

B. Independence of DoS Traffic Pattern

Each row vector of Mixing Matrix $\boldsymbol{A}$ obtained by ICA is mutually orthogonal, when observation signals are mutually independent. Then, the independence of the DoS attack traffic pattern and the normal pattern is examined by using this character. The independence between DoS attack traffic pattern and normal traffic pattern is investigated with simple models of DoS attack traffic patterns and normal traffic patterns like Figure 4. In Figure 4, S_D is a scale of the DoS attack traffic, S_N is the mean of the amount of normal traffic, T_D is the length of the DoS attack traffic pattern, and T_A is the total length of the normal traffic pattern. From the traffic pattern of TCP packets distributed by "The Internet Traffic Archive"[18], the pattern at a time duration T_A is randomly selected, and assigned as the normal traffic pattern. To examine



(a) DoS traffic pattern (b) Normal traffic pattern

Fig. 4. Traffic pattern model

the independence between the DoS attack pattern and the normal traffic pattern of the above-mentioned model, the angle between each row vectors of Mixing Matrix $\boldsymbol{A}$ is investigated. $T_R(=\frac{T_D}{T_A})$, and $S_R(=\frac{S_D}{S_N})$ are parameters. When the DoS attack traffic pattern and the normal traffic pattern of the above-mentioned model are analyzed with ICA, Mixing Matrix A becomes 2×2 matrix. The angle between two row vectors of A is shown in Figure 5 as a function of T_R. Each value is a mean value of 100 different pairs of the normal traffic pattern and the DoS attack traffic pattern. In Figure 5, the value of angle ranges from about 60° to 90°. Consequently, it is considered that the DoS attack traffic pattern is a pattern with a strong independence in ICA.

However, when DoS attacks occur, the pattern actually

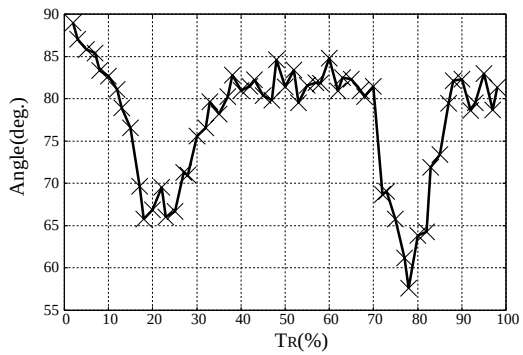


Fig. 5. Relation between T_R and angle

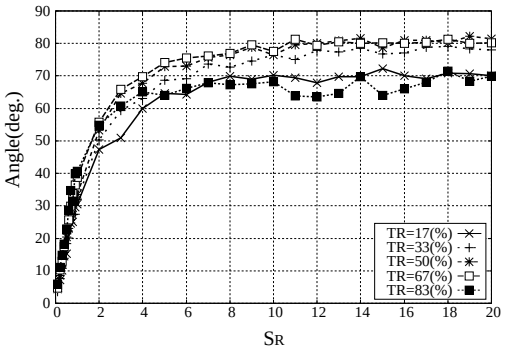


Fig. 6. Relation between S_R and angle

observed is a mixture of DoS traffic and normal traffic. Hence, if S_R is small, the shape of the DoS attack traffic pattern of the model like the above-mentioned is not observed. Then, the independence of the normal traffic pattern including the DoS attack traffic pattern, and the original normal traffic pattern is investigated. The angle between two row vectors of A is shown in Figure 6 as a function of S_R. Figure 6 shows that the angle between the DoS attack pattern and the normal traffic pattern grows as S_R increases. Therefore, it is considered that a large-scale DoS attack traffic pattern has strong independence from the normal traffic pattern.

Therefore, it can be regarded that there is a relation between DoS attack and input traffic pattens of each link, by evaluating the inclusion of the independent pattern, which is not included in past normal traffic patterns in input and output patterns.

C. DDoS Attack Route Judging Algorithm

A method of judging the relation to the attack by comparing traffic patterns using Independent Component Analysis is proposed. In the proposal method, output pattern $\boldsymbol{O}$ at DDoS attack and input pattern $\boldsymbol{I}$ observed in a input link at the same time is acquired. Moreover, from the past pattern observed in the input link, plural patterns of length equal to DDoS attack continuance time are acquired as Sample Patterns to define a normal state of becoming the criterion of relevance between $\boldsymbol{O}$ and $\boldsymbol{I}$. By applying Independent Component Analysis to these, Independent Components ($\boldsymbol{IC}$) which compose observed patterns, and Mixing Matrix $\boldsymbol{A}$ which is the mixing ratio of $\boldsymbol{IC}$ are obtained. At the following, the elements of the Mixing Matrix $\boldsymbol{A}$ is called as Independent Component Score, and analyze them. Table I shows an example of obtained Independent Component Scores by applying Independent Component Analysis to patterns at DDoS attack in a certain link generated through simulation. Scores of $\boldsymbol{O}$, $\boldsymbol{I}$, and each $\boldsymbol{I_{sample}}$ in Table I are judged by following expressions.

$$d_{ave} - 2d_{std} < d \quad \rightarrow \quad Judge\ as\ Normal\ Route \tag{8}$$

$$d \leq d_{ave} - 2d_{std} \quad \rightarrow \quad Judge\ as\ Attack\ Route \tag{9}$$

where d is Euclidean distance between $\boldsymbol{O}$ and $\boldsymbol{I}$, d_{ave} is the mean value of Euclidean distance between $\boldsymbol{O}$ and each $\boldsymbol{I_{sample}}$, and d_{std} is the standard deviation of those distances.

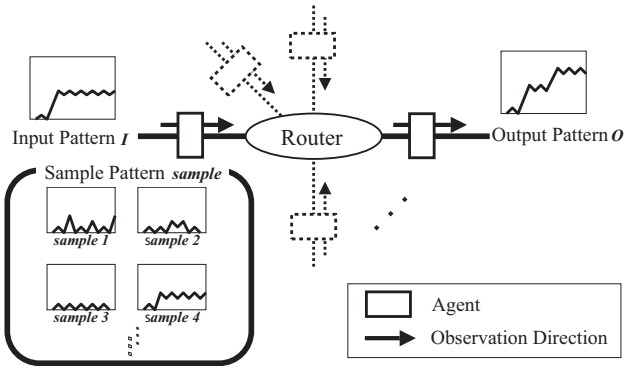


Fig. 7. Acquisition of pattern in proposal method

TABLE I
AN EXAMPLE OF INDEPENDENT COMPONENT SCORES OF EACH PATTERN

	$IC1$	$IC2$	$IC3$	$IC4$	$IC5$	$\cdots$
$\boldsymbol{O}$	-1235	154	-1076	-411	-4105	$\cdots$
$\boldsymbol{I}$	-921	43	-790	-265	-3369	$\cdots$
$\boldsymbol{I}_{sample1}$	152	-201	309	109	-1297	$\cdots$
$\boldsymbol{I}_{sample2}$	147	377	115	48	-212	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Because Independent Component Scores of $\boldsymbol{I}$ which takes part in DoS attack become nearer to the Scores of $\boldsymbol{O}$ than those of each $\boldsymbol{I}_{sample}$, d tends to take a value which is smaller than Euclidean distance between $\boldsymbol{O}$ and each $\boldsymbol{I}_{sample}$. Therefore, it is judged by above expressions. When Euclidean distance d is extremely small compared to the Independent Component Scores between $\boldsymbol{O}$ and each $\boldsymbol{I}_{sample}$, it is interpreted that $\boldsymbol{I}$ has a relation to the attack. All input links are judged by above-mentioned method. Threshold d_{ave} - $2d_{std}$ is set depending on Chebyshev's inequality that 75% of data exists within the range of $d_{ave} \pm 2d_{std}$ in any distribution.

All links connected with a branch point are individually judged by the above-mentioned algorithm. A link judged to have the relevance to the attack is traced to next branch point which to the link connects, and, in the branch point, tracing is carried out by the same algorithm. Finally, a network where DoS attack traffic is generated or the internal host of the network is identified.

IV. SIMULATION

In this section, traffic patterns at DDoS attack are modeled and simulated, and tracing accuracy of the proposed method is compared with two conventional methods.

A. Simulation Model

In this paper, tracing DDoS attack in which more than one DoS attacks confluence is considered as our target. The network branch point shown as Figure 8 is used in our simulation. The total number of links is 21. Using two conventional methods and the proposed method, the relation is found

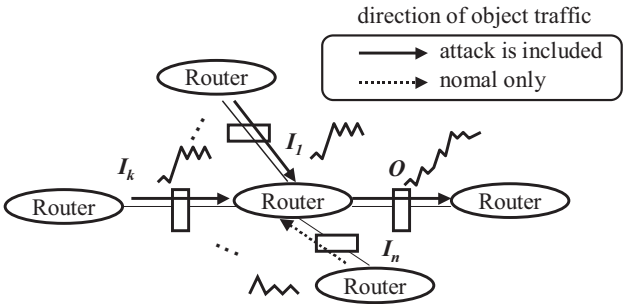


Fig. 8. Simulation model

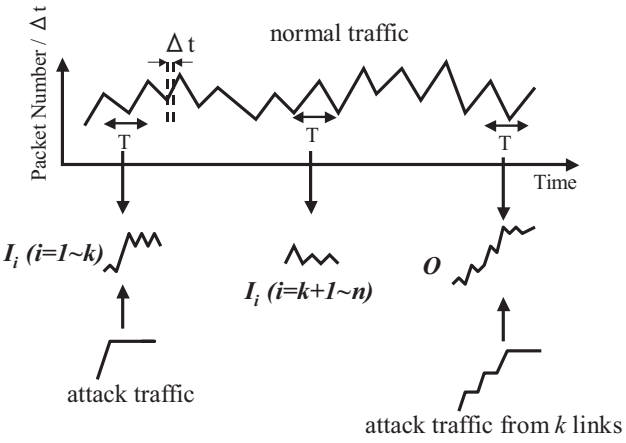


Fig. 9. Generation of traffic pattern

between the patterns observed in 20 input links and 1 output link during DDoS attack. For all the links, from the pattern observed during each unit of observation time Δt of the traffic observed in a certain local network in one day, the pattern at a time duration T which is equal to the length of attack is selected randomly, and assign it as the normal traffic as shown in Figure 9. Here, Δt=10[sec] and T=600[sec] are assigned. Considering the report which says that the length span of DoS attack is usually 300 seconds[10], and also considering the chances that the patterns of DDoS attacks confluent with some delay, the length of DDoS attack T is 600 [sec]. In addition to the normal traffic generated this way, for the patterns of k input links where DoS attacks occur, it is considered that the same scale of normal traffic assigned, to be the traffic pattern of the DoS attack. Moreover, for the sample patterns that are required in the proposed method, patterns from the pattern just before the pattern considered an attack are selected. Then the patterns at lengths T consecutively is extracted and assigned as the sample patterns. Probabilities of normal input patterns being judged as being related to the attack (False Positive), and of the attack-related input patterns being judged as not being related to the attack (False Negative), change depending on the number of sample patterns. For this reason, it is necessary to fix the number of patterns to a proper value. In this simulation, the value in which the sum of False Positives and False

Negatives was minimum are found out, and set it to 10.

Moreover, regarding the traffic flow in the simulation, it is considered that the normal traffic in each link splits and flows to the links beside itself, all the attack traffics in each link flow to the output O, and at each output O, the traffic where $k(2 \leq k \leq 20)$ attacks confluent, is observed. For the confluence of attack traffics, taking the time when the first attack is observed in a certain link as the reference, it is considered that the attacks in other links occur and confluent to the first one within 5 minutes.

B. Performance Comparison with Conventional Tracing Methods

Each tracing method is applied to the pattern generated through the simulation described in the previous subsection, and compared each performance. When a pattern containing attack traffic is judged to have relation to the attack, or a pattern which doesn't contain attack traffic is judged as not having relation to the attack, a judgment of the input link to which pattern is observed is defined to have succeeded. Under this definition, the number k of attack confluences is changed from 2 to 20, and perform 100 simulations respectively. And, the tracing accuracy is evaluated by the ratio of the link which succeeds in the judgment. In a word, tracing accuracy is evaluated by the following expression:

$$\begin{aligned} TracingAccuracy &= \frac{LinksJudgedCorrectly}{JudgedLinks} \\ &= 1 - \frac{FPs + FNs}{JudgedLinks} \end{aligned} \quad (10)$$

Figure 10 shows performance of each tracing method to change in number k of attack confluence. As shown in Figure 10(b) and Figure 10(c), False negative rate of the proposed method is greatly lower though false positive rate is a little higher compared to two conventional methods. The tracing accuracy of the proposed method is the highest when many attacks confluence. About this reason, it is considered that the proposed method is not influenced easily by attack traffic from other links and normal traffic, because the proposed method separates patterns and judges the relation to DDoS attack. From this character, it can be said that the proposed method can trace DDoS attacks with higher accuracy.

C. Performance to other DoS shape

In [19], other types of DoS attacks have been examined. The DoS attack decreases the throughput by intermittent traffic pulses causing TCP congestion control. The feature of this DoS is to decrease throughput by the traffic of the minimum requirement. The shape of such DoS attack traffic pattern becomes different from that of the above-mentioned flooding type DoS. Then, the tracing accuracy is compared against pulse type DoS attacks through simulations. The network branch point and the setting of each parameter in the simulation are similar to 4.1. Moreover, the number of pulses that each attack input pattern contains is set at random between 1 and 30($= \frac{T}{2\Delta t}$). These pulses are transmitted at proper interval.

Figure 11 shows performance of each tracing method for pulse type DoS attack. The tracing accuracy of the proposed method is higher than those of conventional methods when many pulse type DoS attacks confluence. However, the overall tracing accuracy lower than that in case of flooding type DoS. This cause is considered that the pattern does not become unique shape easily, because the traffic of the pulse type DoS attack is less than that of the flood type DoS attack.

V. Conclusion and Future Work

As one of the measures against DoS attack, there are some methods for tracing it using traffic pattern. Because these conventional methods directly use the pattern shapes as the criterion for tracing, accurate tracing is difficult when many attacks confluence and make the pattern shape complicated.

Next, the independence of DoS attack traffic pattern is examined from the result of ICA, and found out that DoS attack traffic patterns have strong independence from normal traffic patterns. And a method to trace DoS traffic patterns using ICA is proposed, and shown through simulation that this method can carry out tracing with higher accuracy even when the input and output patterns are complicated.

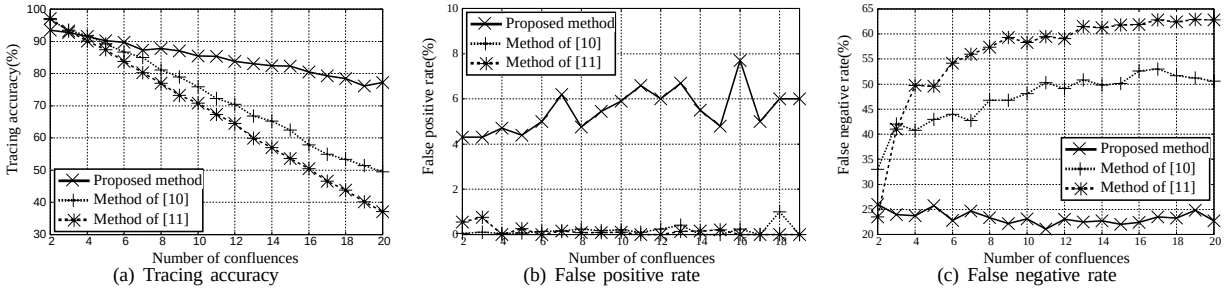
In this simulation, because patterns assigned as sample patterns have little short-term traffic characteristic change, it has been able to define normal state well. However, in the link into which traffic characteristic greatly changes rapidly during the short term, sample patterns cannot define normal state well, and there is a possibility that tracing accuracy decreases when tracing by the proposed method using such sample patterns. Therefore, the examination of the influence of the sample pattern is lest as a future work.

Acknowledgement

This work was supported by Grant-in-Aid for Young Scientists (B) (21700066).

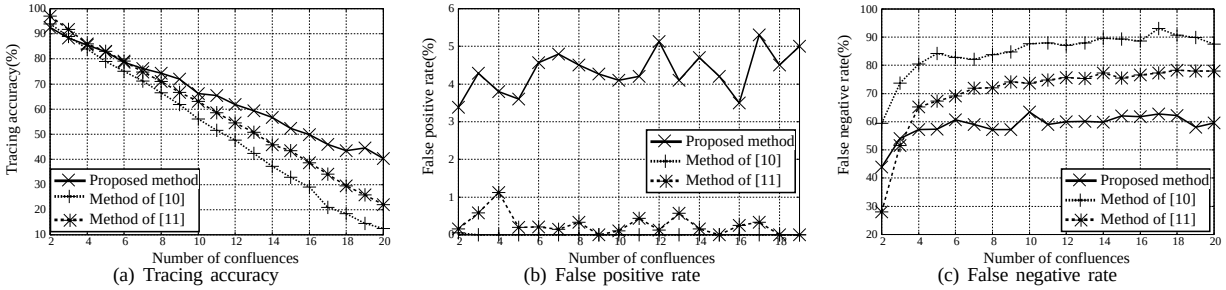
References

[1] Rocky K.C.Chang, Defending against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial, IEEE Communications Magazine, Octorber 2002.
[2] Kohei OHTA and G.Mansfield, Illigal Access Detection on the Internet-Present status and future directions-, IEICE Trans., vol.83-B, no.9, pp.1209-1216, Sep. 2000.
[3] S.Savage, D.Wetherall, A.Karlin, and T.Anderson, Network Support for IP Traceback, IEEE/ACM Trans. Networking, vol.9, no.3, pp.226-237, Jun. 2001.
[4] D.Song and A.Perrig, Advanced and authenticated marking schemes for IP Traceback, Proc. IEEE Infocom 2001 Conf., Anchorage, Alaska USA, April 2001. University of California, Berkeley, Jun. 2000.
[5] H.Lee and K.Park, On the effectiveness of probabilistic packet marking for IP Traceback under denial of service attack, Proc. IEEE Inforcom 2001 Conf., Anchorage, Alaska USA, April 2000.
[6] S.M.Bellovin, ICMP Traceback Messages, InternetDraft, IETF, draft-ietf-itrace-02.txt(work in progress), Nov. 2002.
[7] A.C.Snoren, C.Partridge, L.A.Sanchez, C.E.Jones, F.Tchakountio, S.T.Kent, and W.T.Strayer, Hash-Based IP Traceback, Proc. of ACM SIGCOMM '01, Aug. 2001.
[8] T.Peng, C.Leckie, and K.Ramamohanarao, Adjusted Probabilistic Packet Marking for IP Traceback, Networking, May, 2002, Pisa, Italy.
[9] T.Peng, C.Leckie, and K.Ramamohanarao, Defending Against Distributed Denial of Service Attacks Using Selective Pushback, 9th IEEE International Conference on Telecommunications, June, 2002, Beijing, China.



(a) Tracing accuracy (b) False positive rate (c) False negative rate

Fig. 10. Performance comparison



(a) Tracing accuracy (b) False positive rate (c) False negative rate

Fig. 11. Performance comparison for pulse type DoS

[10] Y. Takei, K. Ohta, N. Kato,and Y. Nemoto, Detecting and Tracing Illigal Access by using Traffic Pattern Matching Technique, IEICE Trans., Vol.J84-B, no.8, pp.1464-1473, Aug. 2001.
[11] K. Sakaguchi, K. Ohta, Y. Waizumi, N. Kato, and Y. Nemoto, Tracing DDoS Attacks by Comparing Traffic Patterns Based on Quadratic Programming Method, IEICE Trans., Vol.J85-B, no.8, pp.1295-1303, Aug. 2002.
[12] CERT Advisory CA-96.21, TCP SYN Flooding and IP Spoofing Attacks, Feb.8, 1996.
[13] CERT Advisory CA-96.01, UDP Port Denial-of-Service Attacks, Feb.8, 1996.
[14] CERT Advisory CA-96.26, Denial-of-Service Attack via ping, Dec.18, 1996.
[15] Y. Uchiyama, Y. Waizumi, N. Kato, and Y. Nemoto, Detecting and Tracing DDOoS Attacks in the Traffic Analysis Using Auto Regressive Model, IEICE Trans., vol.E87-D, No.12, pp.2635-2643, Dec. 2004. E. Leland, M.S. Taqque, W. Willinger, and D.V. Willson, On the self-similar Nature of Ethernet Traffic (Extended Version), Proc. IEEE/ACM Trans. Networking, vol.2, no.1, pp.1-15, Feb. 1994.
[16] A.Hyvärinen, E.Oja, A fast fixed-point algorithm for independent component analysis, Neural Computation, 9(7):pp1483-1492, 1997
[17] D.Moore, G.Voelker, and S.Savage, Inferring Internet Denial-of-Service Activity, Proc. of the 2001 USENIX Security Symposium, May 2001.
[18] The Internet Traffic Archive, Available from http://ita.ee.lbl.gov/contrib/DEC-PKT.html.
[19] A.Kuzmanovic and E.W.Knightyly, Low-Rate TCP-Targeted Denial of Service Attacks(The Shrew vs. the Mice and Elephants), Proc. of ACM SIGCOMM 03, Aug. 2003.