Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 34 (2008) 1659–1665
Expert Systems with Applications
www.elsevier.com/locate/eswa

DDoS attack detection method using cluster analysis

Keunsoo Lee *, Juhyun Kim, Ki Hoon Kwon, Younggoo Han, Sehun Kim

Department of Industrial Engineering, Korea Advanced Institute of Science and Technology, 373-1, Guseong-dong, Yuseong-gu, Daejeon 305-701, Republic of Korea

Abstract

Distributed Denial of Service (DDoS) attacks generate enormous packets by a large number of agents and can easily exhaust the computing and communication resources of a victim within a short period of time. In this paper, we propose a method for proactive detection of DDoS attack by exploiting its architecture which consists of the selection of handlers and agents, the communication and compromise, and attack. We look into the procedures of DDoS attack and then select variables based on these features. After that, we perform cluster analysis for proactive detection of the attack. We experiment with 2000 DARPA Intrusion Detection Scenario Specific Data Set in order to evaluate our method. The results show that each phase of the attack scenario is partitioned well and we can detect precursors of DDoS attack as well as the attack itself.

Keywords: DDoS; Proactive detection; Security; Cluster analysis

1. Introduction

With the rapid development of network technologies, security becomes one of the most important issues today. Especially, there have not been developed fundamental defense solutions of Distributed Denial of Service (DDoS) attacks since these attacks have firstly appeared in June of 1998 (Lin & Tseng, 2004). DDoS attacks make a victim to deny providing normal services in the Internet by flooding a great number of malicious traffic. Attackers do not use the security holes of a network-connected system but launch attacks against its availability. In reality, the widely known web sites, such as Yahoo, eBay, and Amazon.com, were damaged by DDoS attacks in 2000, although these were well-equipped in security. Such web sites were unfortunately attacked only because they are connected through the Internet. Therefore, it is agreed that DDoS attack has become a major threat to the stability of the Internet (Computer Emergency Response Team, 1999).

In a DDoS attack, an attacker compromises a large number of network-connected hosts by exploiting network software vulnerabilities (Xu & Lee, 2003). Then, attack software is installed on these systems through secure channels. A large number of the compromised hosts on which attack software is installed send useless packets toward a victim at a same time. The volume of malicious traffic generated by such hosts is so high that a victim cannot afford it and be instantly paralyzed.

In terms of a victim side, the apparent things which can be differentiated from the other kinds of hackings are that, in summary, the high volume of traffic converges on a victim, the source IP addresses of the malicious packets are spoofed, and the source and/or destination port numbers of the packets are randomly generated depending on the type of attack. For example, Trinoo uses random destination port numbers and Shaft selects random source port umbers. When an attack is going on against a system, some types of traffic which cause serious congestions can be easily observed near the victim. The types of these problematic packets can be TCP, UDP, or ICMP type because the

* Corresponding author. Tel.: +82 42 869 2954; fax: +82 42 869 3110.
E-mail addresses: kslee@tmlab.kaist.ac.kr (K. Lee), jhkim@tmlab.kaist.ac.kr (J. Kim), khkwon@tmlab.kaist.ac.kr (K.H. Kwon), yghan@tmlab.kaist.ac.kr (Y. Han), shkim@kaist.ac.kr (S. Kim).

0957-4174/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2007.01.040

attacker should select the traffic type before launching attack. Most of the hackings can be traced for identifying the attacker, whereas it is very difficult to discover the identity of the attacker in DDoS attacks. It is because attackers make the source addresses of IP packets faked by randomly generating. The attack packets are generated by a great number of agent systems which are controlled by attacker through handler systems. Attacker should select agents and handlers as many as possible before launching attack. For this, he/she must perform network scanning and intrude the systems having security vulnerabilities to install attack software. Changes in traffic are expected during these attack preparation phases.

The objective of this study is identifying clues which can be used as precursors for detecting such attacks proactively. The entropy concept is adopted to analyze the traffic based on each attack phase by the use of cluster analysis method. The results of this study can be applied to security devices, such as IDS (Intrusion Detection System) or firewall, for recognizing such attacks proactively, and contribute to the correct attack detection if the attack precursors are considered in a combined way.

The remainder of this paper is as follows. Section 2 introduces the previous researches relevant to DDoS attack detection, and Section 3 explains DDoS attacks. The proposed method is presented in Section 4. Simulation results are included in Section 5. Finally, this paper is concluded in Section 6.

2. Related works

There have been done lots of researches relevant to DDoS attack defense. DDoS attack is commonly well known as a congestion-based attack. To detect such attacks proactively, Cabrera et al. (2001) used Management Information Base MIB traffic variables intimating attack precursors. Network management systems (NMSs) extract these variables from IP-based, TCP-based, UDP-based, ICMP-based, and SNMP-based traffic. Each MIB has different traffic rate when the network or system is between normal and under attack in the perspective of victim side. Each NMS analyzes the correlations between the communication MIB variables during the attack preparations and the rate-based MIB variables during attacks to recognize DDoS attack precursors proactively. This method is applied to one NMS domain. In case of multiple NMS domain, that is, if attacker and victim are not located in one NMS domain, it is impossible to detect the correlations of variables between during attack preparations and during attacks.

Jeong et al. (2006) used queueing model for attack detection. He adopted it to output interfaces of intermediate routers. The output queue exceeding threshold traffic is considered as a partial attack path, and attack is determined if it continues to reach to the victim. Traffic congestion caused by DDoS attack packets is more easily observed at closer points to the victim than to the attack sources (Mahajan et al., 2002). Hence, there can be more false negatives, and pushing down to the victim can be blocked because of the serious congestion of downstream links.

Jung and Krishnamurthy (2002) discovered the fact that most of the IP addresses in a flash crowd appeared at the web site before, while very few IP addresses appeared in the case of DDoS attacks. This experimental result was adopted by Lee and Shieh (2005), who applied the history of past IP addresses to attack detection and packet filtering. But this scheme is inappropriate for the case that a lot of legitimate users who have not visited before can simultaneously try to access a popular web site.

Gowadia et al. (2005) incorporated the occurrence probability of specific attacks in the existing Bayesian Networks-based intrusion detection systems. By observing the input parameters, they suggested to anticipate the occurrence probability of specific attacks corresponding to the sequence of input parameters. This method requires communications among three agents. Exchanging information has vulnerabilities in terms of security, and applying occurrence probability of attack events can lead to biased results in correct attack detection.

Liao and Vemuri (2001) used *K*-nearest Neighbor Classifier (KNNC) to categorize process into normal or intrusive class. The KNNC calculates the similarity between the new process and each training process instance, and basically assumes that the processes belonging to the same class will cluster together in the vector space. It is excellent in attack detection, but the detector is computationally expensive for real-time implementation when the number of processes simultaneously increases.

The Radial-Basis-Function neural network (RBF-NN) (Haykin, 1994) is used to recognize DDoS attacks from the normal traffic (Gavrilis & Dermatas, 2005). RBF-NN detector is a two layer neural network. It uses nine packet parameters, and the frequencies of these parameters are estimated. Based on the frequencies, RBF-NN classifies traffic into attack or normal class. In this study, the IP spoofing characteristic which is one of the most definite DDoS attack evidences is not considered for more correct attack detection. Regarding UDP type attacks, the detection efficiency is lower than that of TCP type attacks, and is apparently low in the beginning period of attacks. Defining *K*-means centers which minimizes the quantization error is also difficult task.

Stereilein et al. (2002) also presented an attack detection system based on neural network. While it showed improved detection rate with a low false alarm rate when tested with DARPA 1999 IDS Evaluation data, using multilayer perception requires relatively more processing time for determination of attack detection. It does not sure whether the time for attack detection is reduced or not.

Akella et al. (2003) proposed a detection mechanism where each intermediate router detects traffic anomalies using profiles of normal traffic. Each router keeps track of destinations whose traffic occupies greater than a frac-

tion of the capacity of the outgoing link, and sends this information to its neighbors. Attack detection is determined by intermediate routers if the gathered traffic information on a specific destination system exceeds the predefined threshold. This scheme cannot distinguish the flash crowds from the DDoS attacks. Hence, false alarm rate will be increased.

Mahajan et al. (2002) proposed a defense mechanism based on congestion of output queues in an intermediate router. The congestion is estimated based on the rate of packet droppings. When it is necessary to limit the rate of incoming traffic responsible for congestion, the router sends pushback message to request upstream routers to limit the bandwidth of its outgoing links. This scheme does not provide intelligent detection method for DDoS attack. It only focused on controlling the traffic which causes congestion.

Considering the previous schemes, there is commonly tradeoff between attack efficiency and cost. Increasing the attack detection rate requires the increase of false alarm rate or increment of computational overheads or memory overheads. While detecting attacks as soon as possible is very important for preparing defense measures in DDoS attacks, most of the previous researches have been focused on the traffic generated by agents to extract detection parameters. It is valuable to analyze the traffic generated during attack preparation phases as well as that generated during attack phases for proactive attack detection. Therefore, it is necessary to develop a method, which compensates for these drawbacks, for proactive DDoS attack detection.

3. DDoS attack

The techniques of DDoS attacks have been evolved since these attacks have first appeared in June of 1998 (Lin & Tseng, 2004). However, the general attack model and procedures were not changed. In Fig. 1, attacker sets up hierarchical attack architecture. For this, at first, an attacker chooses more than one handler which has security vulnerabilities, and intrudes them by gaining access right. And the procedures for selecting agents (or zombies) are performed as the same way for selecting handlers, but the attacker indirectly achieves it through handlers. Attacker selects these network-connected systems as many as possible. The agents will perform DDoS attack actually by sending unaccountable amounts of malicious traffic to a target system simultaneously. The handlers and agents are commonly located in the external networks of victim's and attacker's network. Once the attacker successfully accomplished the selection of handlers and agents, he/she controls communications among the three systems to compromise attack. Attack target, attack time/period, and attack type is compromised through the communication and compromise, which is done in secure way not to reveal the attack. After the completion of preparations for attack, which is selecting handlers and agents and compromising attack, a great number of agents launch DDoS attack to the victim simultaneously. Mostly, for selecting handlers and agents, scanning is performed to find hosts which have security vulnerabilities, and ICMP is usually used for scanning. For secure communication and compromise among the three systems, the messages for information exchange are usually encrypted.

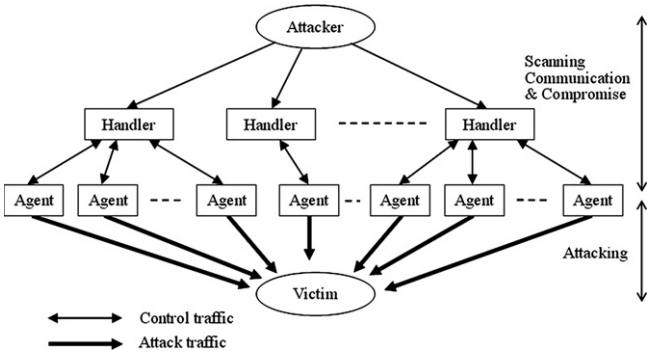

Fig. 1. Architecture of DDoS attack.

The agents generate some types of DDoS attack traffic among TCP, UDP, and ICMP type. Under a DDoS attack, the victim or related network is seriously jammed with specific types of traffic heading for the victim. The agents randomly generate the source IP addresses of attack packets to hide their real addresses. They also randomize the destination and source port numbers depending on the attack type, whereas flash crowds (Jung & Krishnamurthy, 2002) traffic does not. In a DDoS attack, tracing and identifying the real attacker is very difficult because the source IP addresses are spoofed based on the hierarchical attack architecture.

There are two ways to paralyze a victim or network (Lin & Tseng, 2004). The one is only sending a great number of malicious packets toward a victim, such as UDP flood attack and ICMP flood attack. UDP flood attack is possible when an attacker sends an enormous number of UDP traffic with random destination port numbers to a victim. ICMP flood attacks make the agents send large volumes of ICMP Echo Request packets (''ping'') to a victim. These packets require so many ICMP Echo Reply packets as a response from the victim, and induce the saturation of bandwidth of the victim. The consequence of the flood attacks is that the victim or related network is occupied with such malicious traffic. Hence, it has not sufficient bandwidth to allocate for the legitimate users. The other way for paralyzing a system or network is that attackers make use of the vulnerabilities of network protocol. For example, TCP-SYN flooding attack uses the connection characteristic of three-way handshaking of TCP protocol. By spoofing the source IP addresses of attack packets, the victim has a lot of half-opened connections, which result in the resource consumption of the victim system.

For detecting DDoS attacks proactively, the traffic features observed in each attack procedure are used in this research using cluster analysis.

4. Proposed method

In the previous section, we discussed about the architecture of a DDoS attack. This section explains the proposed approach for proactive detection of the DDoS attack. The main idea of our approach is based on the detection of each phase of the DDoS attack separately. Considering the features of the DDoS attack, we can extract several traffic variables which give information about occurrence of each phase in the DDoS attack. These variables can be used to recognize and classify the phases of the DDoS attack, thus we can become aware of the DDoS attack from the initial preparation stage to final attack. This method makes us possible to establish adaptive defense mechanism corresponding to the processes of the DDoS attack. In the first work, we select detection parameters by observing the characteristics of the DDoS attack. After the selection of the parameters, we employ cluster analysis for the proactive detection of the attack.

4.1. Selection of the detection parameter

To detect early stage of DDoS attacks, it requires many measures which can describe the steps of DDoS attacks well.

Let us observe the procedure of a DDoS attack to find out traffic parameters which change abnormally in each step. As mentioned earlier, the DDoS attack is performed by following steps:

- Selection of handlers and agents.
- Communication and compromise.
- Attack.

In the first step, real attacker sends ICMP Echo Request packets to find handlers and agents that help attack, which is called IPsweep (Akella et al., 2003; Cabrera et al., 2001). In this scanning procedure for the DDoS attack, a lot of ICMP traffic is transmitted from an attacker to hosts located in Internet. Therefore, the occurrence rate of ICMP packets may be abnormally high compared to that in usual network traffic. For the communication and compromise between handlers and agents, increased volume of a specific traffic type such as ICMP, UDP, and TCP SYN packet can be observed because any type of packets can be used for message exchange. Hence, the occurrence rates of these packets can be measures which indicate that an attacker prepares to launch a DDoS attack.

The distribution of source IP, destination IP, source port and destination port gives us additional information about each step of the DDoS attack. In IPsweep phase, an attacker spread packets to find handlers and agents. In this period, destination IP address in network flow would be distributed randomly. In contrast, attack packets have diverse source IP address and focused on the destination IP address of victim host in the period of real attack. In order to measure the degree of divergence, we use the concept of entropy (Feinstein et al., 2003).

Let an information source have n independent symbols each with probability of choice P_i. Then, the entropy H is defined as follows (Shannon & Weaver, 1963):

$$H = -\sum_{i=1}^{n} P_i \log_2 P_i$$

Hence, entropy can be computed on a sample of consecutive packets. Comparing the value for entropy of some sample of packet header fields to that of other samples of packet header fields provides a mechanism for detecting changes in the randomness.

When we use entropy value, the value of source IP address becomes small and that of destination IP address increases in the IPsweep phase. In other hand, in the DDoS attack period, the entropy value of source IP address increases and that of destination IP address converges to a very small value. Hence, the entropy values of source and destination IP addresses can be good measures for proactive detection.

Similarly, we can find useful detection parameters by using entropy value. The entropy values of source and destination port numbers can be applied to detect DDoS attacks because some types of DDoS attacks use random port numbers in the attack period (Criscuolo, 2000). In addition, the entropy value of packet type is worth observing because DDoS attacks use specific packet type such as ICMP flood attack and UDP flood attack (Houle & Weaver, 2001; Criscuolo, 2000). If the entropy of packet type converges to a small value, it needs to suspect to be under attack.

Finally, the agents generate enormous packets heading to the victim in the period of real attack and the related network is seriously jammed. The number of packets is a definite evidence of taking place of the attack.

Arranging them, our parameters for the detection of DDoS attack are as follows:

- Entropy of source IP address and port number.
- Entropy of destination IP address and port number.
- Entropy of packet type.
- Occurrence rate of packet type (ICMP, UDP, TCP SYN).
- Number of packets.

Undoubtedly, there would be more variables which are helpful for the accurate detection of DDoS attacks. However, a detection model having too many parameters requires additional operation time.

4.2. Cluster analysis

Cluster analysis is to group data so that objects in a given group are similar to each other and dissimilar from other groups. By using cluster analysis, we can separate normal traffic and each phase of the DDoS attack into partitioned groups if variables involved to form cluster have

dissimilarities among them. Hence, in this paper, we apply cluster analysis to separate each phase of the DDoS attack and identify precursors for detection.

There are two main types of cluster algorithms; hierarchical and partitioning (Kaufman & Rousseeuw, 1990). Partitioning method is inappropriate for our case because the number of clusters should be pre-determined in partitioning, even though we have no information about it. Therefore, we adopt a hierarchical method. This method is often used to classify plants and animals, and is expected to be adequate for classifying the phases of the DDoS attack by the use of their features.

We use nine variables, which are explained in the previous section, in the process of forming clusters. Each variable is normalized to eliminate the effect of difference between scales of the variables. With normalization, variables become

$$z = \frac{x - \bar{x}}{s},$$

where x is the value of each variable, $\bar{x}$ is the mean of sample data set, s is sample standard deviation.

To measure dissimilarities among clusters, cluster analysis compute distance measures from the variables. The most common distance measures are Euclidian distance, the geometric distance in multidimensional space, and Mahalanobis distance based on the covariance matrix of the variables (Staniford-Chen et al., 1998). In the proposed method, we use Euclidian distance since Mahalanobis distance requires the variables to be distributed multivariate normal. Normality is often violated by many data sets and may not be true for network traffic data. The formula of Euclidian distance is as follows:

$$D(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2},$$

where x and y are two records to be clustered and n is the number of variables measured.

After calculating distance measures, Ward's minimum-variance method is employed as a linkage rule. In Ward's method, the distance between two clusters is the ANOVA sum of squares between the two clusters added up over all the variables. At each generation, the within-cluster sum of squares is minimized over all partitions. To determine the number of cluster, we use Cubic Clustering Criterion (CCC) developed by W. S. Sarle of SAS Institute (Johnson, 1998; SAS Institute, 1990). CCC measure plots the CCC values versus the number of clusters and watches for peak. The CCC should be grater than three and form a peak at a possible cluster solution.

5. Simulation results

In the simulation tests, the 2000 DARPA Intrusion Detection Scenario Specific Data Set is used which includes a DDoS attack run by a novice attacker (MIT Lincoln Lab, 2000). This attack scenario is carried out over multiple network and audit sessions. These sessions have been grouped into five attack phases. The five phases are as follows:

1. IPsweep to the DMZ hosts from a remote site.
2. Probe of live IP's to look for the sadmind daemon running on Solaris hosts.
3. Breaks-in via the sadmind vulnerability, both successful and unsuccessful on those hosts.
4. Installation of the Trojan mstream DDoS software on three hosts in the DMZ.
5. Launching the DDoS.

Fig. 2 shows the network structure of this data set. In this simulation, we consider the DMZ concept on the network architecture. Since we adopt the DMZ, attackers cannot access the victim hosts in the inside network directly. To attack the victim host in the inside network, attackers have to control the agent hosts in DMZ network. This Data Set has two types of Tcpdump file. One is DMZ Tcpdump which is collected at the sniffer on the DMZ network, the other is inside Tcpdump which is collected at the sniffer on the inside network. In this attack scenario, the attacker only communicates with agent hosts in the DMZ network and can not communicate with the victim host in the inside network. For this reason, we use the DMZ Tcpdump file to detect the DDoS attack in early phases. In phase 5 of the attack, packets collected to DMZ Tcpdump are not the attack packet but the response packets to the spoofed IP of the attack packets.

The data files were collected over a span of approximately 3 h. In our simulation, each input variable of proposed method is calculated in certain unit time which is 1 s. The variables collected are normalized. After normalization, we perform cluster analysis using SAS Enterprise Miner. In this simulation we use hierarchical method and Ward's linkage rule and CCC measure to determine cluster number, as mentioned earlier (SAS Institute, 1990).

Table 1 shows the result of the cluster analysis. The data set is partitioned into six clusters. From the descriptions of data sets, we can examine which cluster corresponds to the specific phase. Cluster 1 and 2 correspond to the normal period. Cluster 3 and 4 correspond to the phase 1 and 2, respectively. Cluster 5 corresponds to the DDoS attack phase which continues 5 s. Cluster 6 has only one member and its feature is very similar to cluster 5, attack phase.

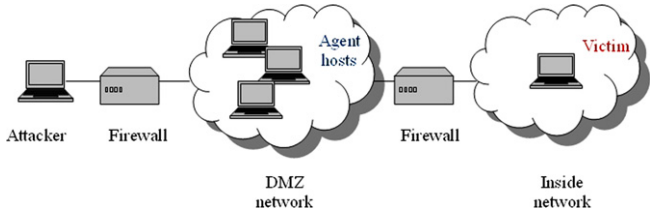

Fig. 2. Network structure used in data set.

Table 1
Result of cluster analysis

Cluster	Phase	Frequency	Nearest cluster
1	Normal	9589	2
2	Normal	56	1
3	Phase 1	21	1
4	Phase 2	32	1
5	Attack	5	6
6	Post-attack	1	5

Table 2 shows the average value of each variable of each cluster. Cluster 1 is normal phase, the entropy values in this phase are all nearby 1.5 and occurrence rate of the specific packet type is low. Cluster 2 is also normal. The entropy values of every variable are same as normal state, but the number of packets and the occurrence rate of TCP SYN packets are relatively low. Among the DDoS attacks, TCP SYN flooding attacks exist, but the number of packets is very small, so it is difficult that we conclude that this is flooding attack.

Cluster 3 is phase 1. The attacker sends ICMP Echo Requests in this phase and listens for ICMP Echo Replies to determine which hosts are "up". The attacker sends many ICMP Echo Request packets to many hosts. So, the entropy value of source IP address and source port number and destination port number are relatively low, on the other hand, the entropy value of destination IP address is relatively high. And the occurrence rate of ICMP packet is abnormally high, because Ipsweep occurs in this phase and most of packets passing by network are ICMP packets.

Cluster 4 corresponds to the phase 2, all entropy values in this phase are relatively low and the number of packets is abnormally small and the occurrence rate of UDP packet is very high. In phase 2, the hosts discovered in phase 1 are probed to determine which hosts are running the vulnerable software. In this scenario, each host is probed by sadmind exploit program which generates UDP packets.

In cluster 5, it has very low entropy values of source IP address and packet type, and very high entropy values of source port number, destination IP address and destination port number. Because we use DMZ Tcpdump data, packets collected at DMZ network are the response packets to the attack packets. So the entropy value of source IP address is very low. Since the agents use randomly spoofed source port number and source IP address and target port number, the entropy values of source port number, destination IP address and destination port number are very high. From the value of number of packets, we also observe that the DDoS attack uses many packets that swamps victim's network and the value is more than 160 times of the value of the normal state.

The member of cluster 6 follows the attack phase, this phase means that the effect of attack remains in the network. That is, although attack is over, the response packets are still observed.

In this simulation, we cannot extract phase 3 and phase 4. These phases are the steps that the attacker intrudes agent hosts and installs DDoS software, therefore, the changes in network traffic do not appear.

In this paper, we use nine input variables to detect DDoS attacks. Using the principle component analysis (PCA), we can reduce the dimension of our model from 9 to 3, and we can describe this data set in three-dimensional space (Jolliffe, 1986). Fig. 3 shows the three-dimensional plot of the result of cluster analysis. Our proposed methods show that each phase of the attack scenario is partitioned well and we can detect not only the attack phase but also phase 1 and 2 in the attack scenario.

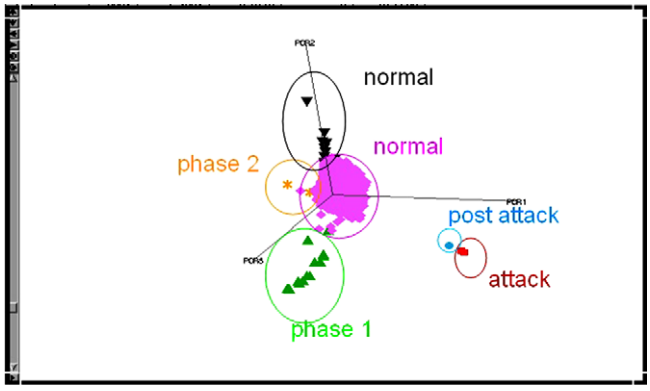

Fig. 3. 3D plot by PCA.

Table 2
Average of each cluster

Variable	Cluster					
	1 Normal	2 Normal	3 Phase 1	4 Phase 2	5 Attack	6 Post-attack
Entropy of source IP	1.59	1.06	0.71	0.08	0.02	0.13
Entropy of source port	1.61	1.07	0.56	0.12	12.4	11.4
Entropy of destination IP	1.58	1.06	4.91	0.07	12.6	11.5
Entropy of destination port	1.50	1.07	0.55	0.12	12.6	11.5
Entropy of packet type	1.12	1.36	0.53	0.04	0.02	0.12
Number of packets	37.0	4.70	41.4	1.19	6225	2876
Occurrence rate of TCP SYN	0.02	0.44	0	0	0	0
Occurrence rate of UDP	0.00	0	0	0.99	0	0
Occurrence rate of ICMP	0.00	0	0.87	0	0	0

6. Conclusions

In this paper, we present an efficient method to detect and control DDoS attacks proactively using cluster analysis. Although there have been lots of studies for the detection of DDoS attacks, they mostly focused on the traffic generated during the attack period. To find precursors of a DDoS attack, we look into the feature of the DDoS attack and select nine parameters which show abnormal changes in traffic according to the phases of the attack. After the parameter selection, cluster analysis is applied to form groups into which normal traffic and each phase of the DDoS attack are partitioned.

In order to evaluate this detection method, we experiment with 2000 DARPA Intrusion Detection Scenario Specific Data Set. As a result, we can divide data set into normal groups, phase 1, phase 2, attack, and post-attack group, respectively. Among the five phases of the DDoS attack, we can detect three phases and our proposed methods show that each phase of the attack scenario is partitioned well.

We can detect precursors of the DDoS attack at early phases by using this method, so we can handle the DDoS attack proactively. Moreover, our method is easy to implement since it uses only normalized distance. These features can help construct a defense mechanism against DDoS attacks.

There are some issues worthy of future research. In the future work, we expect to analyze network traffic more effectively by extracting more variables and develop an advanced detection algorithm. We analyzed our algorithm for DDoS attack included in 2000 DARPA data set only. It may be desirable to apply the proposed method to different types of DDoS attacks and data sets.

Acknowledgement

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment)" (IITA-2005-(C1090-0502-0020)).

References

Akella, A. et al. (2003). Detecting DDoS Attacks on ISP Networks. In *ACM SIGMOD/PODS Workshop on management and processing of data streams (MPDS) FCRC*. <http://citeseer.ist.psu.edu/akella03-detecting.html>.

Cabrera, J. B. D. et al. (2001). Proactive detection of distributed denial of service attacks using MIB traffic variables-A feasibility study. In *Proceedings of the seventh IFIP/IEEE international symposium on integrated network management, Seattle, May, 1–14*.

Computer Emergency Response Team (1999). *Results of the distributed-systems intruder tools workshop*. <http://www.cert.org/reports/dsit_workshop-final.html>.

Criscuolo, P. J. (2000). *Distributed denial of service Trin00, Tribe Flood Network, Tribe Flood Network 2000, and Stacheldraht CIAC-2319*. Department of Energy Computer Incident Advisory (CIAC), UCRL-ID-136939, Rev. 1, Lawrence Livermore National Laboratory.

Feinstein, L. et al. (2003). Statistical approach to DDoS attack detection and response. In *Proceedings of the DARPA information survivability conference and exposition* (pp. 303–314).

Gavrilis, D., & Dermatas, E. (2005). Real-time detection of distributed denial-of-service attacks using RBF networks and statistical features. *Computer Networks, 48*(2), 235–245.

Gowadia, V. et al. (2005). PAID: A probabilistic agent-based intrusion detection system. *Computers and Security, 24*(7), 529–545.

Haykin, S. (1994). Neural networks: A comprehensive foundation. Upper Saddle River, NJ: Predice Hall.

Houle, K. J., & Weaver, G. M. (2001). Trends in denial of service attack technology. CERT and CERT Coordination Center, Carnegie Mellon University.

Jeong, S. et al. (2006). An effective DDoS attack detection and packet-filtering scheme. *IEICE Transactions on Communications, E89-B*(7), 2033–2042.

Johnson, D. E. (1998). Applied multivariate method for data analysis. Brooks/Core Publishing Co.

Jolliffe, I. T. (1986). Principal component analysis. New York: Springer-Verlag.

Jung, J. & Krishnamurthy, B. (2002). Flash crowds and denial of service attacks: Characterization and implications for CDNs and web sites. In *Proceedings of ACM conference on computer and communications security, May 30–41*.

Kaufman, L., & Rousseeuw, P. J. (1990). *Finding groups in data: An introduction to cluster analysis. Wiley series in probability and mathematical statistics*. John Wiley and Sons, Inc.

Lee, F. Y., & Shieh, S. (2005). Defending against spoofed DDoS attacks with path fingerprint. *Computers and Security, 24*(7), 571–586.

Liao, Y., & Vemuri, R. (2001). Use of *K*-nearest neighbor classifier for intrusion detection. *Computers and Security, 21*(5), 439–448.

Lin, S. C., & Tseng, S. S. (2004). Constructing detection knowledge for DDoS intrusion tolerance. *Expert Systems with Applications, 27*, 379–390.

Mahajan, R. et al. (2002). Controlling high bandwidth aggregates in the network. *ACM Computer Communication Review, 32*(2), 62–73.

MIT Lincoln Lab (2000). *DARPA intrusion detection scenario specific datasets*. <http://www.ll.mit.edu/IST/ideval/data/2000/2000_data_index.html>.

SAS Institute (1990) (Fourth ed.). *SAS/STAT user's guide, version 6* (Vol. 1). SAS Institute.

Shannon, C. E., & Weaver, W. (1963). *The mathematical theory of communication*. University of Illinois Press.

Staniford-Chen, S. et al. (1998). GrIDS—A graph-based intrusion detection system for large networks. In *The 19th national information systems security conference* (pp. 361–370).

Stereilein, W. W. et al. (2002). Improved detection of low-profile probe and denial-of-service attacks. In *Workshop on statistical and machine learning techniques in computer intrusion detection, Baltimore, Maryland, June 11–13*.

Xu, J., & Lee, W. (2003). Sustaining availability of web services under distributed denial of service attacks. *IEEE Transactions on Computers, 52*(2), 195–208.