A New DDoS Detection Model Using Multiple SVMs and TRA*

Jungtaek Seo[1], Cheolho Lee[1], Taeshik Shon[2], Kyu-Hyung Cho[2],
and Jongsub Moon[2]

[1] National Security Research Institute,
62-1 Hwaam-dong, Yuseong-gu, Daejeon 305-348, Republic of Korea
{seojt, chlee}@etri.re.kr
[2] CIST, Korea University,
1-Ga, Anam-dong, Sungbuk-Gu, Seoul, Republic of Korea
{743zh2k, mathbank, jsmoon}@korea.ac.kr

Abstract. Recently, many attack detection methods adopts machine learning algorithm to improve attack detection accuracy and automatically react to the attacks. However, the previous mechanisms based on machine learning have some disadvantages such as high false positive rate and computing overhead. In this paper, we propose a new DDoS detection model based on multiple SVMs (Support Vector Machine) in order to reduce the false positive rate. We employ TRA (Traffic Rate Analysis) to analyze the characteristics of network traffic for DDoS attacks. Experimental results show that the proposed model is a highly useful classifier for detecting DDoS attacks.

1 Introduction

As we can see in the incidents of Distributed Denial of Service (DDoS) attacks against commercial web sites such as *Yahoo, e-Bay*, and *E*Trade*, computing resources connected to the Internet are vulnerable to DDoS attacks [1], [2], [3]. DDoS attacks can temporarily disable the network services or damage systems by flooding a huge number of network packets for several minutes or longer.

Since these DDoS attacks are harmful to almost all networked systems which have limited computing resources (e.g. network bandwidth, memory, CPU, etc), these attacks are regarded as a serious problem, and thus much research is in progress to detect and prevent them [4] ,[5], [6].

In our earlier research, we presented Traffic Rate Analysis (TRA) to analyze the characteristics of network traffic for the DDoS attacks [7], [8], [9]. TRA is a network traffic analyzing method which examines the occurrence rate of a specific type of packet within the stream of monitored network traffic and is composed of a TCP flag rate and a Protocol rate. The result of analyzing network traffic using TRA showed us that there are distinct and predictable differences between normal traffic and DDoS

* This work was supported by the Ministry of Information Communication, Korea, under the Information Technology Research Center Support Program supervised by the IITA.

T. Enokido et al. (Eds.): EUC Workshops 2005, LNCS 3823, pp. 976 – 985, 2005.

attack traffic. We were able to generate DDoS detection rules by compiling the experimental results with a SVM [10]. However, the false positive rate of the model using single SVM is too high. In order to reduce the false positive rate and to increase the detection rate, we propose the model based on multiple SVMs instead of single one. The experimental results show the proposed detection method has high degree of performance, and detects various DDoS attacks successfully with low false positive rate.

We introduce related research in section 2, and explain TRA in section 3. The background knowledge of SVM is discussed in section 4. In section 5, the experimental environment is introduced and the detection performance of SVM and other machine learning algorithms are tested and compared. Lastly, we mention the conclusion of this research and the direction of future work in section 6.

2 Related Work

Detecting the DDoS attacks is an essential step to defend DDoS attacks. Thus, there have been many researches to detect the DDoS attacks [4], [5], [6]. When DDoS attacks occur, there is a big mismatch between the packet flows "to-rate" toward the victim and "from-rate" from the victim. Gil and Poletto propose the method that examines the disproportion between "to-rate" and "from-rate" in order to detect DDoS attacks [4]. Kulkarni et al [5] presents DDoS detection methods based on randomness of IP spoofing. Almost DDoS attackers use IP spoofing to hide their real IP addresses and locations. Since spoofed IP addresses are generated randomly, this characteristic of randomness may be used to reveal the occurrence of DDoS attacks. Kulkarni's method uses *Komogorov complexity metrics* to measure the randomness of source IP addresses in network packet headers [11]. Wang et al. proposed the method that detects DDoS attack based on the protocol behavior of *SYN-FIN*(RST) pairs [6]. In the normal situation, the ratio of *SYN* and *FIN* is balanced because of the characteristic of the TCP 3-Way handshake. However, the ratio of *SYN* packet increases drastically during the SYN flooding attack. By monitoring sudden change of the ratio of *SYN* and *FIN*, the method detects SYN flooding attacks.

However, these approaches are based on the specific characteristics of the attacks such as mismatch of "to-rate" and "from-rate", effect of IP spoofing, and unbalance of the ratio of *SYN* and *FIN* packet. Thus, these may not properly detect the attack that use undefined characteristic. For example, Gil's method is not applicable to detect attacks using IP spoofing since the method cannot discriminate legitimated packet and spoofed packet, and Wang's method is only applicable to SYN flooding attacks. On the other hand, the proposed detection model automatically generates detection rules using TRA and multiple SVM.

3 Traffic Rate Analysis

3.1 Definition of Traffic Rate Analysis

Traffic rate analysis was defined as measuring packet traffic in a network [7]. It examines the occurrence rate of a specific type of packets within the stream of monitored network traffic, and is composed of TCP flag rate and Protocol rate. TCP flag rate is defined in the following equation.

$$R_{td}[F\, i \mid o] = \frac{\sum flag\,(F)\ in\ a\,TCP\ header}{\sum TCP\ \ packets} \tag{1}$$

TCP flag rate means the ratio of the number of a specific TCP flag to the total number of TCP packets. In the equation (1), a TCP flag '*F*' can be one of *SYN, FIN, RST, ACK, PSH, URG,* and *NULL*, and '*td*' is the time interval used to calculate the value. The direction of network traffic is expressed as '*i*' (inbound) and '*o*' (outbound). For example, $R_1[Si]$ means the occurrence rate of *SYN* flags within TCP packets when measuring inbound network traffic (toward the monitored network) during interval 1.

$$R_{td}[[TCP|UDP|ICMP]\, i \mid o] = \frac{\sum [TCP|UDP|ICMP]\, packets}{\sum IP\ \ packets} \tag{2}$$

Protocol rate is defined in equation (2). It means the ratio of specific Transport-Layer protocol (e.g. TCP, UDP, and ICMP) packets to total Network-Layer (IP) protocol packets. For instance, $R1[TCPi]$ means the occurrence rate of TCP packets within IP packets when measuring outbound network traffic (from the monitored network) during interval 1.

3.2 Network Traffic Changes Under DDoS Attacks

To analyze the change of network traffic from normal web traffic to DDoS attack traffic or vice versa, it is necessary to make a network environment truly identical with the real Internet environment.

Our experimental target is web traffic, and web traffic is composed of HTTP requests and replies based on TCP sessions. For example, when a user clicks a certain web site address on his or her web browser, the web browser establishes TCP connections to the relevant web server. After that, the web browser sends HTTP requests to the web server, and the web server sends HTTP replies to the web browser.

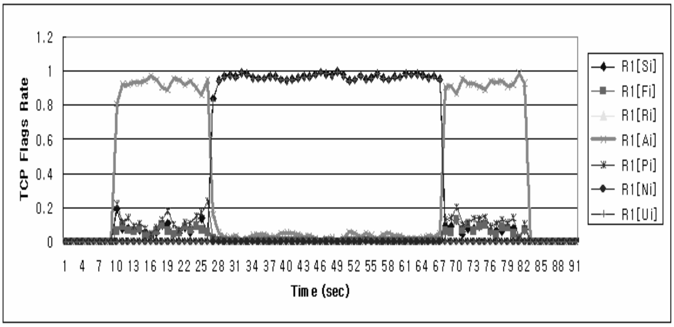

Fig. 1. Network traffic under SYN flooding attack

Since web service is based on TCP connection, the number of HTTP requests in a TCP session (*R/C*: Requests per connection) and the number of TCP sessions simultaneously established (*SC*: Simultaneous Connection) are the key features of web traffic in terms of network traffic analysis. In other words, we can simulate various web traffic environments by adjusting these two features (*R/C* and *SC*).

R/C values include 1, 2, 5, and 10, and *SC* can take on values of 5, 10, 50, 100, 150, and 200. Thus we have twenty-four different network environments. With these various web traffic settings, we compared normal web traffic with DDoS attack traffic.

Fig. 1 shows us that $R1[S\text{i}]$ and $R1[U\text{i}]$ drastically change (go up to almost 1.0) and the other flags decrease (almost 0.0) relatively under SYN flooding attack. When web traffic flows from the 9th second to the 83rd second, a SYN flooding attack occurs between the 26th and 67th second. This phenomenon is caused by the burst of SYN and URG packets, which are generated by SYN flooding attack.

Furthermore, we can also see big changes of network traffic during other types of DDoS attacks such as ICMP flooding attacks or UDP flooding attacks [7], [8], [9].

4 Support Vector Machine

4.1 Background

Support Vector Machine (SVM) is a learning machine that plots the training vectors in high-dimensional feature space, and labels each vector by its class. SVM views the classification problem as a quadratic optimization problem. It combines generalization control with a technique to avoid the “curse of dimensionality” by placing an upper bound on a margin between the different classes, making it a practical tool for large and dynamic data sets. SVM classifies data by determining a set of support vectors, which are members of the set of training inputs that outline a hyper plane in feature space. The SVM is based on the idea of structural risk minimization, which minimizes the generalization error, i.e. true error on unseen examples. The number of free parameters used in the SVM depends on the margin that separates the data points to classes but not on the number of input features. Thus SVM does not require a reduction in the number of features in order to avoid over fitting. SVM provides a generic mechanism to fit the data within a surface of a hyper-plane of a class through the use of a kernel function. The user may provide a kernel function, such as a linear, polynomial, or sigmoid curve, to the SVM during the training process, which selects support vectors along the surface of the function. This capability allows classifying a broader range of problems [12], [13].

4.2 SVM for Categorization

In this section we review some basic ideas of SVM. Given the training data set $\{(x_i, d_i)\}_{i=1}^{N}$ with input data $x_i \in R^N$ and corresponding binary class labels $d_i \in \{-1,1\}$, the SVM classifier formulation starts from the following assumption. The classes represented by the subset $d_i = 1$ and $d_i = -1$ are linearly separable, where $\exists w \in R^N$, $b \in R$ such that

$$\exists w,b \quad s.t \quad \begin{Bmatrix} w^T x_i + b > 0 & for & d_i = +1 \\ w^T x_i + b < 0 & for & d_i = -1 \end{Bmatrix} \tag{3}$$

The goal of SVM is to find an optimal hyper plane for which the margin of separation, ρ, is maximized. ρ is defined by the separation between the separating hyperplane and the closest data point. If the optimal hyperplane is defined by $\left(w_0^T \cdot x\right) + b_0 = 0$, then the function $g(x) = w_0^T \cdot x + b_0$ gives a measure of the distance from x to the optimal hyperplane.

Support Vectors are defined by data points $x^{(s)}$ that lie the closest to the decision surface. For a support vector $x^{(s)}$ and the canonical optimal hyperplane g, we have

$$r = \frac{g(x^s)}{\|w_0\|} = \begin{Bmatrix} +1/\|w_0\| & for & d^{(s)} = +1 \\ -1/\|w_0\| & for & d^{(s)} = -1 \end{Bmatrix} \tag{4}$$

Since, the margin of separation is $\rho \propto \frac{1}{\|w_0\|}$. $\|w_0\|$ should be minimal to achieve the maximal separation margin. Mathematical formulation for finding the canonical optimal separation hyperplane, given the training data set $\{(x_i, d_i)\}_{i=1}^N$, solves the following quadratic problem

$$\begin{Bmatrix} \min \tau(\omega, \xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} \zeta_i \\ s.t \quad d_i(w^T x_i + b) \geq 1 - \zeta_i \quad for \quad \zeta_i \geq 0, \quad i = 1, \ldots, l \end{Bmatrix} \tag{5}$$

Note that the global minimum of above problem must exist, because $\Phi(w) = \frac{1}{2}\|w_0\|^2$ is convex in w and the constrains are linear in w and b. This constrained optimization problem is dealt with by introducing Lagrange multipliers $a_i \geq 0$ and a Lagrangian function given by

$$L(w,b,\zeta,a,v) = \tau(w,\zeta) - \sum_{i=1}^{l} a_i \left[d_i(w_i^T x_i + b) - 1 + \zeta_k\right] - \sum_{i=1}^{l} v_i \zeta_i \tag{6}$$

which leads to

$$\frac{\partial L}{\partial w} = 0 \iff w - \sum_{i=1}^{l} a_i d_i x_i = 0 \quad (\therefore w = \sum_{i=1}^{l} a_i d_i x_i) \tag{7}$$

$$\frac{\partial L}{\partial b} = 0 \iff \sum_{i=1}^{l} a_i d_i = 0 \tag{8}$$

The solution vector thus has an expansion in terms of a subset of the training patterns, namely those patterns whose a_i is non-zero, called Support Vectors. By the Karush-Kuhn-Tucker complementarity conditions, we have,

$$a_i \left[d_i\left(w^T x_i + b\right) - 1\right] = 0 \quad for \quad i = 1, \ldots, N \tag{9}$$

by substituting (7),(8) and (9) into equation (6), find multipliers a_i for which

$$\max \Theta(a) = \sum_{i=1}^{l} a_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{i=1}^{l} a_i a_j d_i d_i \langle x_i \cdot x_j \rangle \tag{10}$$

$$s.t. \quad 0 \le a_i \le c, \quad i = 1,\ldots,l \quad and \quad \sum_{i=1}^{l} a_i y_i = 0 \tag{11}$$

The hyperplane decision function can thus be written as

$$f(x) = \operatorname{sgn}\left(\sum y_i a_i \cdot (x \cdot x_i) + b \right) \tag{12}$$

where b is computed using (9).

To construct the SVM, the optimal hyperplane algorithm has to be augmented by a method for computing dot products in feature spaces nonlinearly related to input space. The basic idea is to map the data into some other dot product space (called the feature space) F via a nonlinear map Φ, and to perform the above linear algorithm in F, i.e nonseparable data $\{(x_i, d_i)\}_{i=1}^{N}$, where $x_i \in R_N$, $d_i \in \{+1, -1\}$, preprocess the data with,

$$\Phi : R^N \rightarrow \Theta(x) \quad where \quad N << \dim(F) \tag{13}$$

Here w and $x_i$are not calculated. According to Mercer's theorem,

$$(\Phi(x_i) \cdot \Phi(x_j)) = K(x_j, x_j) \tag{14}$$

and $K(x, y)$ can be computed easily on the input space. Finally the nonlinear SVM classifier becomes

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{l} a_i d_i K(x_i \cdot x) + b \right) \tag{15}$$

5 Experiment

5.1 DDoS Detection Process

Fig. 2 shows the overall composition of the DDoS detection process.

It is composed of two steps. One is the preprocessing step, and the other is the training and testing step. In the preprocessing step, it captures raw network traffic from both DDoS and legitimate network traffics, and extracts features from the captured raw network traffic using TRA method for each training and test set. For both training and testing, we used 10 features; $R_1[S_i]$, $R_1[F_i]$, $R_1[R_i]$, $R_1[A_i]$, $R_1[P_i]$, $R_1[U_i]$, $R_1[N_i]$, $R_1[TCP_i]$, $R_1[UDP_i]$, and $R_1[ICMP_i]$.

In the training and testing step, they are trained by each machine using the training set. To train the machine, we classify input packets of the training set as *attack (-1)* and *normal (+1)*. Normal web traffic was categorized as *normal*, and the various DDoS attack traffic was categorized as *attack*. The trained machines evaluate test sets, and discriminate legitimate traffic and DDoS traffic. In the experiments, we used two different machine learning models; single SVM and multiple SVMs model. Multiple SVMs model consists of several SVMs that are learned by different training data, and each SVM is specialized to specific attacks (e.g., Smurf attack, Tfn2k attack, SYN flooding). In the experiment, we categorized the attacks into three types; DoS attack, DDoS attack, and DrDoS attack.

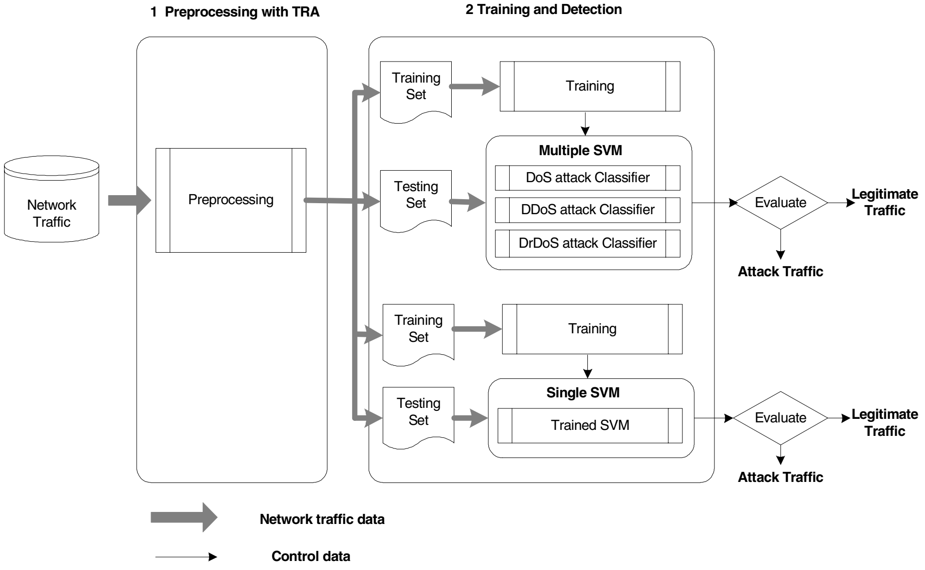

Fig. 2. Overall composition of DDoS detection process

5.2 Experimental Environment

Our traffic monitor was developed with the network packet capturing library *libpcap*. It is divided into two modules. One is the module for capturing network traffic and calculating TCP flag rate and protocol rate from the monitored network, and the other is the DDoS attack detection module tuned by the support vector machine. The TRA analyzer is located on the adjacent site of a target Web server and captures both inbound and outbound network traffic packets through an Ethernet hub, and then calculates TCP flag rate and protocol rate in every second.

Web clients are composed of four hosts using SPECweb99 to generate normal web traffic toward an Apache web server. To generate DDoS attack traffic toward the web server we used several attack tools as shown in the Table 1. We collected network traffic for 100 seconds during training and testing time. Web clients continually generated normal web traffic toward the web server, and various attacks occurred between the 25th and 75th seconds.

In the multiple SVMs model, each attack classifier is trained by own training sets (e.g., DoS training set, DDoS training set, and DrDoS training set), and each training set consists of 500 normal data and 500 attack data. On the other hand, the training data set of single SVM model consists of 1500 normal data and 1500 attack data, and the attack data is not classified. Table 2 shows the composition of training data sets.

Table 1. Classification of Attack Tools

Attack tool	Attacks
Targa3 (DoS)	bonk, jolt, land, nestea, newtear, syndrop, teardrop, winnuke, sai hyousen, oshare, etc.
TFN2K (DDoS)	ICMP flooding, UDP flooding, etc.
pHorgam (DrDos)	DrDoS

Table 2. Composition of training data sets

Model	Classifier	Data Type	Number of Data
Multiple SVM	DoS attack classifier	DoS attack	500
		Normal	500
	DDoS attack classifier	DDoS attack	500
		Normal	500
	DrDoS attack classifier	DrDoS attack	500
		Normal	500
Single SVM		Attack	1500
		Normal	1500

5.3 Detection Performance Analysis

For each training set, we used *dot* and *polynomial* kernel with epsilon 0.01. We used 1000 as capacity parameter and +0.01 as epsilon parameter. The detection performance using multiple SVM and single is shown in Table 3.

Table 3. Detection performance of multiple SVM and single SVM

Kernel	Model	False Positive (%)	False Negative (%)		
			DoS	DDoS	DrDoS
Poly	Multiple SVM	26.82	4.33	0.80	0.19
	SVM	40.15	5.32	1.60	1.38
Linear	Multiple SVM	9.84	0.19	1.40	1.36
	SVM	20.47	2.95	2.60	8.87

As we can see in Table 3, multiple SVMs show slightly higher detection performance than single SVM with decrease of false positive rate. Since SVM is a binary classifier, the normal region decreases according to increasing of attack region. Decreasing of normal region means that the increasing of probability of false positive. Fig. 3 shows the relation between normal region and attack region. Each attack type (e.g., DoS, DrDoS, and DDoS) has own attack region. In the experiment, single SVM model merged these regions into a single huge attack region, while multiple SVMs model does not merge these attack regions. The reason is that multiple SVMs trains each attack classifier independently using different training data. Thus, the false positive rate of multiple SVM model is lower than single the rate of single SVM model.

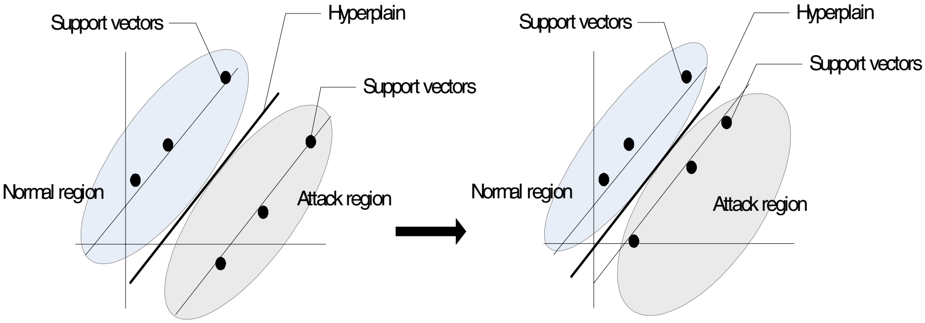

Fig. 3. Relation between normal region and attack region in SVM

6 Conclusions

In this paper, we utilized a TRA (Traffic Rate Analysis) method proposed in earlier research [7], [8]. In addition, we adopted multiple SVMs (Support Vector Machine) model instead of single SVM model in order to compile detection rules. As a result, multiple SVMs model shows slightly higher detection accuracy and lower false positive rate. We expect that our approach will be useful in providing early detection of DDoS attacks against the Internet infrastructure. However, our machine learning scheme does not have unsupervised feature but supervised feature. It may mean sometimes if our scheme meet a kind of unexpected situation, it is difficult it can work well or not. Thus, we need additional work using unsupervised learning method without pre-existing knowledge. In our future work, we will use other kernel function methods of SVM and various machine learning methods. Moreover, we are going to focus on detecting and defending against other types of attacks like worms.

References

1. Garber, L.: Denial-of-Service Attacks Rip the Internet, IEEE Computer, vol. 33(4), (2000) 12-17
2. Houle, J.K., and Weaver, M.G.: Trends in Denial of Service Attack Technology, CERT Coordination Center, (2001)
3. Moore, D., Voelker, G.M., and Savage, S.: Inferring Internet Denial-of-Service Activity. In Proceedings of the 10th USENIX Symposium, (2001) 9-22
4. Gil, T.M, and Poletto, M.: MULTOPS: a data-structure for bandwidth attack detection, In Proceedings of the 10th USENIX Security Symposium, (2001) 23-38
5. Kulkarni, A.B., Bush, S.F., and Evans, S.C.: Detecting Distributed Denial-of-Service Attacks Using Kolmogorov Complexity Metrics. Technical report 2001CRD176, GE Research and Development Center, (2001)
6. Wang, H., Zhang, D., and Shin, K.G.: Detecting SYN Flooding Attacks, In Proceedings of IEEE INFOCOM – The Conference on Computer Communications, vol. 21, no. 1, (2002) 1530-1539
7. Lee, C., Noh, S., Choi, K., and Jung, G.: Characterizing DDoS Attacks with Traffic Rate Analysis, In Proceedings of the IADIS e-Society, vol. 1, (2003) 81-88

8. Noh, S., Lee, C., Choi, K., and Jung, K.: Detecting Distributed Denial of Service (DDoS) Attacks through Inductive Learning, Lecture Notes in Computer Science(LNCS), Springer-Verlag, vol. 2690, (2003) 286-295
9. Seo, J., Lee, C., and Moon, J.: Defending DDoS Attacks Using Network Traffic Analysis and Probabilistic Packet Drop, In Proceedings of the Third International Conference on Grid and Cooperative Computing, (2004) 390-397
10. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines, Cambridge University (2000)
11. Li, M., and Vitanyi, P.: An Introduction to Kolmogorov Complexity and Its Applications, Springer-Verlag, Section 7.6, (1997) 506-509
12. Ruping S.: mySVM – a Support Vector Machine, University of Dortmund (2004)
13. Burges. C.: LA Tutorial on Support Vector Machines for Patter Recognition, Data Mining and Knowledge Discovery, Boston, 1588