



Traffic flooding attack detection with SNMP MIB using SVM[☆]

Jaehak Yu, Hansung Lee, Myung-Sup Kim^{*}, Daihee Park

Department of Computer and Information Science, Korea University, Yeongi-Gun, Republic of Korea

ARTICLE INFO

Article history:

Received 18 March 2008

Received in revised form 10 September 2008

Accepted 10 September 2008

Available online 19 September 2008

Keywords:

Intrusion detection

SNMP

MIB

DoS/DDoS

Support vector machine

ABSTRACT

Recently, as network flooding attacks such as DoS/DDoS and Internet Worm have posed devastating threats to network services, rapid detection and proper response mechanisms are the major concern for secure and reliable network services. However, most of the current Intrusion Detection Systems (IDSs) focus on detail analysis of packet data, which results in late detection and a high system burden to cope with high-speed network traffic. Little or no integration exists between IDS and SNMP-based network management, in spite of the extensive monitoring and statistical information provided by SNMP agents implemented on network devices and systems. In this paper we propose a lightweight and fast detection mechanism for traffic flooding attacks. Firstly, we use SNMP MIB statistical data gathered from SNMP agents, instead of raw packet data from network links. The involved SNMP MIB variables are selected by an effective feature selection mechanism and gathered effectively by the MIB update time prediction mechanism. Secondly, we use a machine learning approach based on a Support Vector Machine (SVM) for attack classification. Using MIB and SVM, we achieved fast detection with high accuracy, the minimization of the system burden, and extensibility for system deployment. The proposed mechanism is constructed in a hierarchical structure, which first distinguishes attack traffic from normal traffic and then determines the type of attacks in detail. Using MIB datasets collected from real experiments involving a DDoS attack, we validate the possibility of our approaches. It is shown that network attacks are detected with high efficiency, and classified with low false alarms.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

With the ever more rapid development of the Internet, the Internet is currently an infrastructure for all kinds of network service. For example, the Web and VoIP service became the most popular and general services for the Internet. Many other new services such as IPTV service are emerging in the Internet. The significant increase of our dependency on Internet-based services in everyday life has intensified the survivability of networks. Because of extensive public availability, the Internet has become the main target of malicious attacks. Both the systems connected to the Internet and the network devices comprising the Internet, can all be severely compromised by intrusions. Recently, network flooding attacks such as DoS/DDoS and Internet Worm have posed devastating threats to network services. Moore et al. [1] reported that the DoS/DDoS attack is the main threat to the entire Internet, and the majority of them (90–94%) are deployed by using TCP. As a re-

sult, rapid detection and fast response mechanisms are the major concern for secure and reliable network services [2–6].

Intrusion Detection attempts to detect network and system attacks by examining various data records obtained from target systems and network. The network and system attacks are categorized into two types: host-based attacks [HAs] and network-based attacks [NAs] [7]. HAs target a machine and try to gain access to privileged services or resources on that machine. The detection systems for HAs usually obtain and analyze the system call data from an audit-process which tracks all system calls made on behalf of each user. On the other hand, NAs make it difficult for legitimate users to access various network services, by intentionally occupying or sabotaging network resources and services. This can be done by sending large amounts of network traffic, after exploiting well-known faults in networking services, overloading network or system, etc. The detection systems for NAs use network traffic data (packet data, flow data, MIB data, etc.) to examine traffic addressed to the machines being monitored. In this paper we focus on the NAs.

There are two general approaches to Intrusion Detection: Misuse Intrusion Detection (MID) and Anomaly Intrusion Detection (AID) [7–10]. Similar to virus detection, MID is based on pattern matching to identify intrusions. These known patterns are referred to as signatures, which are extracted from the known attacks. MID

[☆] This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (KRF-2007-331-D00387) and IT National Scholarship Program of MIC, Korea.

^{*} Corresponding author. Tel.: +82 41 860 1347.

E-mail addresses: dbzzang@korea.ac.kr (J. Yu), mohan@korea.ac.kr (H. Lee), tmskim@korea.ac.kr (M.-S. Kim), dhpark@korea.ac.kr (D. Park).

can detect attacks with high accuracy for pre-learned attacks. However, MID has the intrinsic disadvantage of low flexibility, because the signature must be manually updated for new types of attack. However, AID constructs a normal usage behavior profile, named a historical or long-term behavior profile. And, the analysis model examines deviations of the short-term behavior profile from the norm. The deviations can be treated as the baselines for distinguishing attack activities from normal behaviors. The basic assumption of the AID is that an intruder's behavior will be noticeably different from that of legitimate users. This approach is useful to detect new types of attack, but has the unavoidable limitation that it is difficult to take proper action against the attack, due to the lack of detail information. The main concern of current research in this area is to use the advantages of MIDs and AIDs by overcoming their limitations. In this paper we use a support vector machine (SVM) based AID mechanism for attack detection and in-detail attack type classification.

Most of the current Intrusion Detection Systems (IDSs) investigate packet data to evaluate the security status of the network and system, which results in a significant processing burden and eventually, late detection time. Little or no integration exists between IDS and SNMP-based Network Management Systems (NMS) in spite of the extensive monitoring and statistical information offered by SNMP agents running on network elements. For example, SNMP MIB-II, an IETF standard MIB supported by all the SNMP agents, provides a large number of traffic performance information on different layers and protocols: IP, TCP, UDP, ICMP, etc. SNMP agents are already implemented in most current network elements, thus, MIB statistical data are available to easily collect for security analysis. This can be extended to collect additional data pertinent to network activities and is independent of the operating systems. By enlisting the MIB data, IDS promises a lower processing overhead for analysis, and high flexibility of deployment.

Recently, it has been reported that SVM is one of the most successful classification algorithms in the data mining area and provides good performance for anomaly network intrusion detection [9,11,12]. However, most of the SVM based and other AID approaches assume raw packet data inspection for the input training and testing dataset of the system, because their performance was evaluated by testing with the KDD-Cup99 dataset [13], which is extracted from raw tcpdump packet data. It is worthwhile to apply these approaches to the SNMP MIB dataset for the analysis of network and system security.

Some studies [4,5,14–18] used SNMP MIB data for a intrusion detection. Li et al. [4] developed a system named as MAID which uses SNMP MIB-II data for anomaly detection. They periodically collected 27 MIB variables from 4 MIB-II groups [19] (Interface, IP, TCP, and UDP), and converted them into a probability density function (PDF) to calculate statistical similarity metrics which is the input data of the attack classifier. For the detection mechanism, they used a neural network classifier, a typical backpropagation (BP) network, other than SVM. Cabrera et al. [6,14] also used SNMP MIB for symptom analysis of the DDoS attack. Puttini et al. [15] applied the associated Bayesian classification to the SNMP MIB variables to detect anomalous network traffic behavior in Mobile Ad Hoc Networks (MANET). Ramah et al. [16] developed an anomaly detection system using periodic SNMP data collection which is derived from a PCA (Principle Component Analysis) based unsupervised anomaly detection scheme proposed by Shyu et al. [17]. According to our literature review, these studies focused on the detection of intrusion from normal traffic, but most of them did not consider the determination of attack types, such as TCP-SYN Flooding, UDP flooding, ICMP flooding, etc. These classification mechanisms have their own structural limitations such that BP mechanism should reconstruct a training data from the beginning when a new attack occurs and Bayesian mechanism has difficulty in dealing with the features

having continuous values. Furthermore, SVM-based classification mechanisms are not utilized even though other machine learning approaches (BP, Bayesian, etc.) are selectively used.

In this paper, we propose a lightweight and fast detection mechanism for the traffic flooding attacks such as DoS/DDoS and Internet Worm. Firstly, we use the SNMP MIB statistical data gathered from SNMP agents, instead of the raw packet data from network links. The involved SNMP MIB variables are selected by an effective feature selection mechanism named correlation feature selection (CFS) [27]. The SNMP MIB data are effectively retrieved from the target system as soon as the MIB variables are updated at the target system. Here our MIB update time prediction mechanism is applied for the fast detection. Secondly, we use a machine learning approach based on a Support Vector Machine (SVM) for attack classification. Our overall objective is to integrate IDSs with NMSs by attempting to construct SVM-based IDSs working on SNMP MIB data. Using the MIB data and SVM, we achieved fast detection with high accuracy, minimization of the processing burden, and flexibility of system deployment. The proposed mechanism is constructed as a hierarchical two-level structure. At the first level, a one-class SVM distinguishes attack traffic from normal traffic. At the second level, a multi-class SVM identifies the type of attacks in detail: TCP-SYN flooding, UDP flooding, ICMP flooding, etc. Using MIB datasets collected every 15 seconds during 10 days from real experiments involving DDoS attack, we tested the possibility of our approaches. Our experimental results showed that the detection accuracy of the proposed mechanism approaches 99.40%, with a false positive rate (FPR) and false negative rate (FNR) of 1.8% and 0.6%, respectively. It is shown that network attacks are detected with high efficiency, and classified with relatively low false alarms.

The paper is organized as follows. Section 2 describes three considering points for the SNMP-based attack detection process. In Section 3 we present the proposed SVM-based hierarchical two-level structure for traffic flooding attack detection. Section 4 describes the experiments and results. Section 5 closes this paper with our conclusion and possible future work.

2. Considering points for SNMB-based traffic flooding attack detection

SNMP provides a universal method of exchanging data for purposes of monitoring systems that reside on a network. The use of SNMP is most dominant in the modern industry. But, to utilize SNMP for traffic flooding attack detection, we need to consider the following three points in the use of the SNMP MIB variables which affects the performance and accuracy of the detection system: (1) Proper selection of SNMP MIB variables for attack detection, (2) Determination of the detection timing about when and how often, (3) Algorithm for attack detection using the selected MIB variables.

The first, we need to select proper SNMP MIB variables for attack detection. This selection should be done to meet that the number of SNMP MIB variables involved is minimized and the range of attack types covered is maximized. Yoo et al. [30] utilized `tcpInErrs`, `udpNoPorts`, and `icmpOutEchoReps` SNMP MIB variables for the purpose of detecting DoS/DDoS attacks. However, most of modern attack tools get matured and generate error-free packets to a valid port in a victim system. Attackers scan vulnerabilities of a victim host first before sending a large flood of packets targeting the victim host, which sabotage both systems resources and network resources. Therefore, the MIB variables used in [30] may not contribute much as before on detecting modern attacks. We solve this MIB selection problem by the correlation based feature selection algorithm (CFS) [27].

Identifying a representative set of features from which to construct a classification model for a particular task has been a central problem in machine learning area [22–27]. Feature selection is the problem of selecting a subset of d -features from a set of D -features based on some optimization criterion. Oh et al. [24] proposed a hybrid genetic algorithm for feature selection. Fleuret et al. [25] suggested a fast feature selection technique based on conditional mutual information. Li et al. [26] proposed an interactive RELEAF for feature weighting. But these modern feature selection mechanisms are not suitable because of their high computational complexity and RELAF bias. In this paper, we utilized the popular CFS algorithm whose performance is widely accepted and is implemented as a component of the WEKA [28]. For the experiment, firstly we selected 13 MIB variables from empirical observation and literature review, which are mostly affected by network flooding attacks. Among 13 MIB variables we selected 7 MIB variables by CFS.

For the second consideration, we need to determine the detection timing: when and how often the detection system is triggered to retrieve SNMP MIB variables from a target system and analyze them to decide whether the system is normal or abnormal state. Short detection interval gives a fast detection while high processing and traffic overhead occurs as a side-effect, while long detection interval results in late detection with low burden in system and network. We have to determine a suitable detection time and interval.

Fig. 1 illustrates the change of *ifInOctets* MIB value in the interval of 1 second which is gathered from a Net-SNMP agent running on a typical Linux system. Fig. 1 says that SNMP MIB variables are periodically updated with a certain time interval. The *ifInOctets* MIB variables of Net-SNMP agent is updated in about every 15 sec. We figured out that the update interval is different from system to system and from agent to agent. If we do not consider the periodic update of SNMP MIB values, the MIB values retrieved at a certain time is the values at 15 sec before in the worst case. The best scenario is to retrieve MIB values just after the MIB values are updated in a SNMP agent. By triggering detection system whenever the MIB variables are updated, we can achieve the requirement of fast detection. But detection of update time is another problem to be solved. System load and network overhead must be considered in order to manage multiple target systems. We solved this problem by predicting the next MIB update time using exponential averaging scheme. In this scheme, we calculate the exponential average of past MIB update intervals and predict the next MIB update time. By sleeping the system until the next MIB update time is reached, we can reduce system and network

burden and activate the detection system just after the MIB data are updated.

Thirdly, we have to develop an algorithm for attack detection using the SNMP MIB data gathered from a target system. This is the problem of how to utilize MIB data for attack detection. Several machine learning mechanisms, such as BP, C4.5, and Bayesian networks, have been considered for the classification of attack traffic from normal traffic [6,14,15]. But the SVM based mechanism, which is widely accepted as the most successful classification mechanism in the pattern recognition area, has not been utilized in the SNMP MIB-based attack classification until now, while it was utilized in the packet-based attack detection. The existing machine learning based approaches concentrated on distinguishing attack traffic from normal traffic, but gave little attention to identify the types of attack. In this paper, we propose a SVM-based two-level hierarchical structure for traffic flooding attack detection, which detects attacks from normal traffic as well as identifies the types of attack. The two-level hierarchical structure is very flexible and extendable. When a new attack type is known, the only thing we have to do is to add the corresponding SVDD module for the new attack in the second level of our structure.

3. SVM-based hierarchical two-level structure

In this Section, we introduce the basic concept of SVM and the proposed SVM-based hierarchical two-level structure for traffic flooding attack detection, in which an one-class SVM distinguishes attack traffic from normal traffic at the first layer, then, at the next layer, a multi-class SVM classifies the type of attacks in detail: TCP-SYN flooding, UDP flooding, ICMP flooding, etc. The overall architecture of our proposed system is given in Fig. 2.

Recently, the support vector learning method has emerged as a promising tool in the area of intelligent systems, since it has shown an excellent performance for pattern classification and function approximation, by ensuring the global optimum for a given problem. However, it has the intrinsic structural limitation of the binary classifier. According to a recent literature review, there are three major known types of approaches for multi-class SVM: one-against-all, one-against-one, and DAGSVM in the form of combining many binary SVMs [9]. In this paper, instead of adopting one of the previous works, we construct a series of SVDDs operating in parallel at the second layer of the proposed system, as illustrated in Fig. 2, in order to determine the type of attacks in detail.

In general, the number of datasets necessary for training varies according to the amount of normal traffic and attack traffic. Hence, the resulting training may not be independent of other classes, due

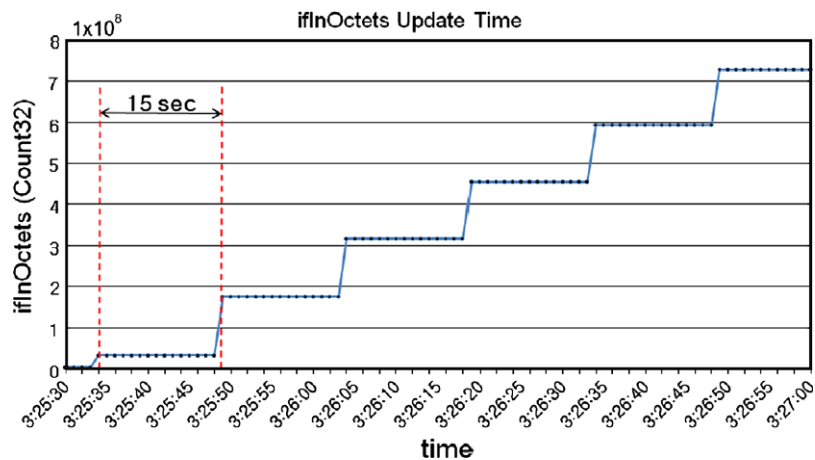


Fig. 1. Periodic update of *ifInOctets* MIB variable in a typical SNMP agent.

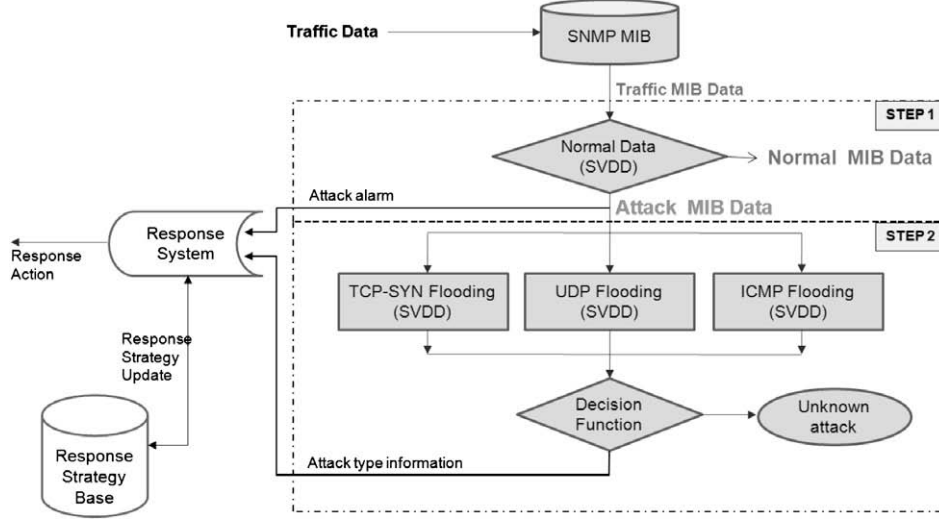


Fig. 2. The overall architecture of the proposed system.

to the unbalanced size of training data. In addition, it may not be true that the current training data represents the whole classes, since new types of attack are increasingly emerging. Thus, the binary classifier SVM may suffer from misclassification of novel attack data, by creating a decision boundary including an unobserved area. Accordingly, it is preferable to select a decision boundary function using a one-class SVM which expresses the corresponding class independently (one of the most well-known one-class SVMs is a support vector data description (SVDD)). The multi-class SVM based on SVDD is described as follows [9]:

Given a k -data set of N_k patterns in ad -dimensional input space, $D_k = \{x_i^k \in R^d \mid i = 1, \dots, N_k\}; k = 1, \dots, K$, the multi-class SVM based on SVDD is defined as the problem of obtaining a hypersphere which contains as many training datasets as possible, while keeping the radius small. It is formalized as the following mathematical optimization problem:

$$\begin{aligned} \min \quad & L_0(R_k^2, a_k, \zeta_k) = R_k^2 + C \sum_{i=1}^{N_k} \zeta_i^k \\ \text{s.t.} \quad & \|x_i^k - a_k\|^2 \leq R_k^2 + \zeta_i^k, \zeta_i^k \geq 0, \forall i. \end{aligned} \quad (1)$$

Where a_k is the center of the sphere that expresses the k -th class, R_k^2 is the square value of a sphere radius, ζ_i^k is the penalty term that shows how far i -th training data x_i^k deviates from a sphere, and C is the trade-off constant.

By introducing a Lagrange multiplier for each inequality condition, we obtain the following Lagrange function:

$$\begin{aligned} L(R_k^2, a_k, \zeta_k, \alpha_k, \eta_k) = & R_k^2 + C \sum_{i=1}^{N_k} \zeta_i^k + \sum_{i=1}^{N_k} \alpha_i^k [(x_i^k - a_k)^T (x_i^k - a_k) \\ & - R_k^2 - \zeta_i^k] - \sum_{i=1}^{N_k} \eta_i^k \zeta_i^k \end{aligned} \quad (2)$$

where $\alpha_i^k \geq 0, \eta_i^k \geq 0, \forall i$.

From the saddle point condition, Eq. (2) has to be minimized with respect to R_k^2, a_k , and ζ_i^k , and maximized with respect to α_k and η_k . The optimal solution of (1) should satisfy the following:

$$\begin{aligned} \frac{\partial L}{\partial R_k^2} = 0 : & \sum_{i=1}^{N_k} \alpha_i^k = 1. \\ \frac{\partial L}{\partial \zeta_k^2} = 0 : & C - \alpha_i^k - \eta_i^k = 0, \alpha_i^k \in [0, C], \forall i. \\ \frac{\partial L}{\partial R_k^2} = 0 : & a_k = \sum_{i=1}^{N_k} \alpha_i^k x_i^k \end{aligned} \quad (3)$$

After substitution of the above into the Lagrange function L , we obtain the following dual problem:

$$\begin{aligned} \min \quad & \sum_{i=1}^{N_k} \sum_{j=1}^{N_k} \alpha_i^k \alpha_j^k \langle x_i^k, x_j^k \rangle - \sum_{i=1}^{N_k} \alpha_i^k \langle x_i^k, x_i^k \rangle \\ \text{s.t.} \quad & \sum_{i=1}^{N_k} \alpha_i^k = 1, \alpha_i^k \in [0, C], \forall i. \end{aligned} \quad (4)$$

A sphere can express a more complex decision boundary in feature space, F . We can map an input space into a feature space using the kernel function, K . Therefore, the training can be performed by solving the following convex quadratic problem

$$\begin{aligned} \min \quad & \sum_{i=1}^{N_k} \sum_{j=1}^{N_k} \alpha_i^k \alpha_j^k k_k(x_i^k, x_j^k) - \sum_{i=1}^{N_k} \alpha_i^k k_k(x_i^k, x_i^k) \\ \text{s.t.} \quad & \sum_{i=1}^{N_k} \alpha_i^k = 1, \alpha_i^k \in [0, C], \forall i. \end{aligned} \quad (5)$$

When the Gaussian function is chosen for the Kernel function, we always have $k(x, x) = 1$ for each $x \in R^d$. Thus, the above problem can be further simplified as follows:

$$\begin{aligned} \min \quad & \sum_{i=1}^{N_k} \sum_{j=1}^{N_k} \alpha_i^k \alpha_j^k k_k(x_i^k, x_j^k) \\ \text{s.t.} \quad & \sum_{i=1}^{N_k} \alpha_i^k = 1, \alpha_i^k \in [0, C], \forall i. \end{aligned} \quad (6)$$

Note that in this case, the decision function of each class can be summarized as follows:

$$\begin{aligned} f_k(x) = & R_k^2 - \left[1 - 2 \sum_{i=1}^{N_k} \alpha_i^k k_k(x_i^k, x) + \sum_{i=1}^{N_k} \sum_{j=1}^{N_k} \alpha_i^k \alpha_j^k k_k(x_i^k, x_j^k) \right] \\ & \geq 0 \end{aligned} \quad (7)$$

Since the output $f_k(x)$ of a one-class SVM defined in different feature spaces represents the absolute distance between corresponding data and decision boundary, determining the pertaining class by comparing absolute distances in different feature spaces is not feasible. Accordingly, we calculate the relative distance $f_k(x) = f_k(x)/R_k$, and decide the class having maximum relative distance as the one to which the input data x pertains.

$$\begin{aligned} \text{Class of } x &\equiv \arg \max_{k=1, \dots, K} \hat{f}_k(x) \\ &\equiv \arg \max_k \left\{ \left[R_k^2 - \left(1 - 2 \sum_{i=1}^{N_k} \alpha_i^k k_k(x_i^k, x) + \sum_{i=1}^{N_k} \sum_{j=1}^{N_k} \alpha_i^k \alpha_j^k k_k(x_i^k, x_j^k) \right) \right] / R_k \right\} \end{aligned} \quad (8)$$

4. Experiment

4.1. Testbed network structure

We constructed a testbed network to carry out an actual attack experiment. Fig. 3 illustrates the network topology, which consists of one victim system, two attack agent systems, one attack handler system, and one dataset collector system. The OS of the victim system is Linux Fedora 7. Linux Fedora 8 is used for the OS of other systems. Several servers (Web server, SSH/SFTP server, VNC server, Samba server, and MRTG server, etc.) are working on the victim system. The testbed network is connected to the campus network, so normal traffic is generated between the victim host and other hosts outside the testbed network during the experiment period.

We used Stacheldraht [20], a distributed denial of service attack tool, to generate attack traffic. The Stacheldraht was selected because it is a more mature attack tool compared to other attack tools, such as TFN, TFN2K, or Trinoo. The Stacheldraht is composed of handler (master) and agent (daemon) programs. The handler system scans vulnerabilities of the victim host before sending an attack command to the corresponding multiple agent systems. Agent systems produce a large flood of packets targeting the victim host, which sabotage both systems resources and network resources. For the experiments we conducted three types of network flooding attacks: TCP-SYN flooding, UDP flooding, and ICMP flooding attacks.

4.2. SNMP MIB variables

During the attack experiment, the dataset collector system gathered SNMP MIB data from the victim system using SNMP query messages. Firstly, we investigated 66 MIB variables from five MIB-II groups: Interface, IP, TCP, UDP, and ICMP. The data types of the 66 MIB variables are Counter, which is a non-negative 4-byte integer that may be incremented but not decremented. These MIB variables are continuously updated as the outgoing and incoming traffic from/to a system occur, which could possibly be used for attack detection. We selected 13 MIB variables among 66 MIB variables, which are likely to be affected by the attack traf-

fic by a comprehensive but not exhaustive investigation. Most of the 13 MIB variables are used in [29] for traffic flooding attack detection. In the first phase of our experiment, we used these 13 SNMP MIB variables for the proposed detection mechanism. The 13 MIB variables and their corresponding MIB groups are shown in Table 1.

In the second phase of our experiment, we selected 3 MIB variables and 5 MIB variables among the 13 MIB variables for each stage of our attack detection mechanism by the aforementioned feature selection method CFS [27]. The first 3 MIB variables are used for the first-level attack determination from normal traffic. The next 5 MIB variables are used for the second-level attack type classification among attack traffic. Table 2 illustrates the SNMP MIB variables selected by CFS feature set selection method. The `udpInError` MIB variable was selected in both levels by CFS. Only 7 distinct MIB variables among 13 MIB variables are used in the second phase of our detection mechanism.

4.3. MIB data collection

Most SNMP agents update MIB variables with a certain periodic interval as aforementioned in Section 2. For fast attack detection the best scenario is to gather SNMP MIB data just after the corresponding MIB variable is updated and apply them to the detection algorithm. We developed a MIB data collection system which uses a simple prediction mechanism to determine the next update time of SNMP MIB variables in a target system, by which we can achieve fast detection and reduce network traffic overhead.

The MIB data collection system uses `ifInOctets` MIB variable to determine whether the SNMP MIB variables in target system are updated or not. This system consists of two phases of MIB data collection. In the initial phase, the system collects `ifInOctets` MIB data every 1 second for a certain time period (30 min in our system), and determines the minimum update interval (U_{\min}) and the last update time (t_{old}) which are used as initial input data for the subsequent prediction-based MIB data collection.

In the next phase, the system predicts the next MIB update time and the system sleeps until the next predicted update time is reached. In order to decide the next update and sleep time, we used the following equations using the exponential average. U_n is the n -th real MIB update interval. P_n is the n -th exponential average of the MIB update interval U_n which is used as the next predicted MIB update time. The constant value α is set to 0.5 in the equation. Finally, the next sleep time (S_n) is calculated as $P_n - 1$ as follows.

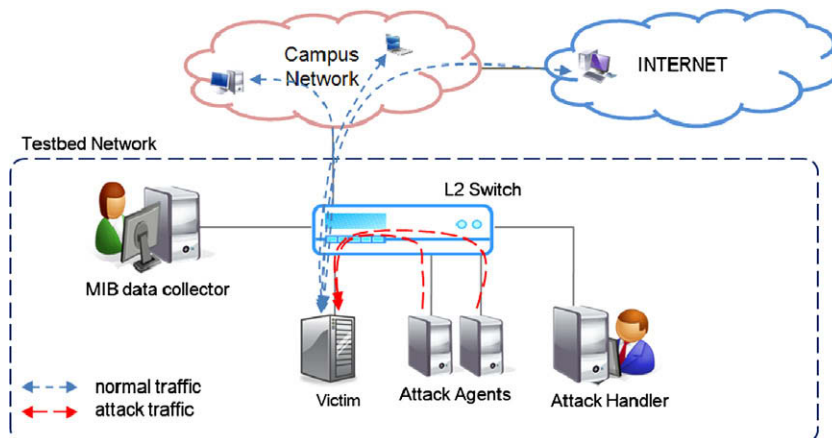


Fig. 3. The testbed network for the DDoS attack experiment.

Table 1
The SNMP MIB variable used for the attack detection mechanism

MIB Group	SNMP MIB variables
IP	ip.ipInReceives (3) ip.ipInDelivers (9) ip.ipOutRequests (10) ip.ipOutDiscards (11)
TCP	tcp.tcpAttemptFails (7) tcp.tcpOutRsts (15)
UDP	udp.udplnErrors (3)
ICMP	icmp.icmplnMsgs(1) icmp.icmplnErrors(2) icmp.icmplnDestUnreachs(3) icmp.icmpOutMsgs(14) icmp.icmpOutErrors(15) icmp.icmpOutDestUnreachs(16)

$$P_n = \alpha \times P_{n-1} + (1 - \alpha) \times U_n$$

$$S_n = P_n - 1 \tag{9}$$

The system calculates the first sleep time (S_1) at the time t_{old} with the following initial values: $P_0 = 0$ and $U_1 = U_{min}$. After n -th MIB data collection, the system calculate the next sleep time P_{n+1} and S_{n+1} , and sleeps for that amount of time. After waking up, the system checks the ifInOctets MIB variable every 1 sec to detect the next MIB update of the system. When ifInOctets MIB variable is updated, the system retrieves all the MIB values used for our detection mechanism.

Thirteen MIB variables listed in Table 1 are collected during 10 days of the attack experiment. The average update interval of the MIB variables in the target system was 15 sec. We collected MIB data every 15 sec of interval in average. At the same time, we recorded the time when attack traffic was sent from the two distributed attack agents. Based on the time log data, we determined their security state for each set of MIB data: normal, TCP-SYN flooding, UDP flooding, and ICMP flooding. We collected a total of 57640 MIB datasets including 2526 TCP-SYN flooding, 2769 UDP flooding, and 2613 ICMP flooding attacks during 10 days of the experiment. Among 57640 MIB datasets we randomly selected 5000 MIB datasets: 2000 for a normal state and 1000 for each attack type (TCP-SYN, UDP, and ICMP flooding attack) for the validation of the proposed detection mechanism.

4.4. Attack identification result

Firstly, we performed an identification test of the proposed mechanism between attack traffic and normal traffic. We performed a SVDD training with 1000 normal MIB records of a training dataset, then, tested with 2500 MIB records of a testing dataset: 1000 normal MIB records and three 500 MIB records for three attacks types: TCP-SYN, UDP, and ICMP flooding attacks. The MIB records in each dataset are selected randomly among all 5000 MIB records.

We used three important formulas [21] to evaluate the performance of the first step of identification: the attack detection rate,

false positive rate (FPR), and false negative rate (FNR), which are as follows:

$$\text{Attack Detection Rate(ADR)} = \frac{\sum_{i=1}^n T_i}{\sum_{i=1}^n I_i} \tag{10}$$

$$\text{False Positive Rate(FPR)} = \frac{\sum_{i=1}^n P_i}{\sum_{i=1}^n N_i} \tag{11}$$

$$\text{False Negative Rate(FNR)} = \frac{\sum_{i=1}^n F_i}{\sum_{i=1}^n I_i} \tag{12}$$

In the above equation, I is an individual attack traffic MIB record, while N is a MIB record for normal traffic. In our experiment, the number of attack traffic records and normal traffic records are 1500 and 1000, respectively. T is an attack traffic record which is classified as an attack by the system. P indicates a normal traffic record which is misclassified as attack traffic. F is an attack traffic record which is misclassified as normal traffic.

We performed the training and testing process with the two different MIB variable sets for the same MIB records: all of the 13 MIB variables and CFS-selected 3 MIB variables. In this experiment, the trade-off constant C in the Gaussian Kernel function is set to $C = 0.1$ in both phases. The parameter σ in the Gaussian Kernel function are chosen as 0.04 and 0.05 in the first and second phase of our experiment, respectively. The experimental results are shown in Table 3. We used the attack detection rate, false positive ratio (FPR), and false negative ratio (FNR) as the performance criteria.

The overall attack detection rates of the two phases are 97.07% and 99.40%, respectively. The second phase with CFS-selected MIB variables outperforms the first phases; this means that the attack identification is highly affected by the selection of MIB variables as well as the attack detection mechanism, and CFS is a good choice for the MIB variable selection.

The FPR is the rate of misclassified normal traffic, as attack traffic over total normal traffic. The FNR is the rate of misclassified attack traffic, as normal traffic over total attack traffic. It is generally accepted that the FNR is more important than the FPR for evaluation of the performance of an intrusion detection system. As shown in Table 3, the value of FNR and FPR were 0.60 and 1.80, respectively, in case the second phase of experiment, which indicates that the proposed attack identification mechanism gives good performance when $\sigma = 0.05$ and $C = 0.1$.

4.5. Attack type classification result

Secondly, we performed a classification test into three types of attacks from attack traffic. For this test, we selected three sets of MIB records for the SVDD training of the three TCP-SYN, UDP, and ICMP flooding attacks. Each set consists of 500 MIB data records which are randomly selected among the total 1000 traffic records. The three SVDDs were used to train three types of different attack. The correctly determined MIB records in the first step of attack identification are used for the performance test of the attack classification. In the first step, the number of misclassified MIB records was 7, 2, and 0 for each type of attacks: TCP-SYN, UDP and ICMP attack, respectively. So we performed the attack-type

Table 2
The SNMP MIB variables selected by CFS

First-level attack identification		Second-level attack-type classification	
MIB Group	SNMP MIB variables	MIB Group	SNMP MIB variables
IP	ip.ipInReceives (3)	IP	ip.ipInDelivers (9)
TCP		TCP	tcp.tcpOutRsts (15)
UDP	udp.udplnErrors (3)	UDP	udp.udplnErrors (3)
ICMP	icmp.icmpOutMsgs(14)	ICMP	icmp.icmplnMsgs(1) icmp.icmplnErrors(2)

Table 3

The performance of the proposed system in the attack identification

Phase	Number of MIBs	σ value	Attack detection rate	FPR	FNR
First	13 (all features)	0.04	97.07 %	2.90 %	2.93 %
Second	3 (selected features)	0.05	99.40 %	1.80 %	0.60 %

Table 4

The performance of the proposed system in the attack classification

Phase	MIBs	σ value	TCP-SYN flooding	UDP flooding	ICMP flooding	Classification accuracy
First	13 MIBs	TCP-SYN	93.38 %	97.95 %	100.00 %	97.18 %
		UDP				
		ICMP				
Second	5 MIBs	TCP-SYN	98.78 %	99.80 %	100.00 %	99.53 %
		UDP				
		ICMP				

classification test with 1491 MIB records: 493, 498 and 500 for each attack type, respectively. For the performance evaluation, we used the classification accuracy shown below:

$$\text{Classification Accuracy} = \frac{\sum_{i=1}^n T_i}{\sum_{i=1}^n I_i} \quad (13)$$

In the above equation, I is an individual attack traffic record in the corresponding attack class. T is the correctly classified attack traffic record. Like in the previous attack identification test, we performed the attack-type classification test with two different set of MIB variables for the same MIB records: 13 MIB variables in the first test and CFS-selected MIB variables in the second test. We used 5 MIB variables in the second test as shown in Table 2. The trade-off constant C was chosen to be 0.1 for all of the six SVDD in the first and second test. But the parameter, σ , of the Gaussian Kernel function was selected as to different values for each attack type and each test, which is show in Table 4. The classification result is also shown in Table 4.

As shown in Table 4, the overall classification accuracy was 99.53% in the test with CFS-selected MIB variables, which outperforms the first test with 13 MIB variables. Only 7 traffic records among 1491 records are misclassified as other types of attack. The result indicates that the proposed mechanism classifies attacks into the detailed attack types with acceptable accuracy. The proposed mechanism identified ICMP flooding attacks with no errors, while 7 attack records are misclassified as TCP-SYN attacks and UDP attacks.

In this experiment, we tested a total of 1500 attack traffic records: 500 attack records for each attack type. The proposed hierarchical flooding attack detection mechanism correctly identified 1491 attacks among 1500 attacks in the first step of the hierarchical mechanism and only 9 traffic records were misclassified. This means the proposed attack detection mechanism gives excellent identification result. The undetected attack traffic records were 7 TCP-SYN flooding attacks and 2 UDP flooding attacks. In the second step of detail attack-type classification, all of the ICMP flooding attacks are precisely classified. Only 6 of the 493 TCP-SYN flooding attacks are misclassified as UDP flooding attacks, and 1 UDP attack is misclassified as TCP-SYN flooding attack. This result also we believe gives good classification result.

5. Conclusion

In this paper we proposed a lightweight and fast attack detection mechanism using an SVM-based hierarchical structure. The proposed mechanism consists of a two-level structure. At the first level a one-class SVM called SVDD identifies attack traffic from

normal traffic with high accuracy. At the second level the attack traffic is separated into several sets according to the corresponding attack types. The proposed mechanism can detect new and unknown attacks at the first level without any additional effort. Also, the system can be easily adapted to newly emerging attacks. By adding an SVDD entity for a new type of attack at the second level of the hierarchical architecture, the proposed mechanism can determine the identity of the newly added attacks.

The proposed mechanism analyzes the security status of a network and system using SNMP MIB traffic records collected from SNMP agents. This makes the detection system lightweight, fast, and flexible. We achieved fast detection by gathering the SNMP MIB data as soon as the MIB variables are updated at the target system, which is achieved by predicting the next MIB update time using an exponential average of past MIB update interval. In the experiment using the MIB variables collected every 15 sec in average we showed that the proposed mechanism can identify attack traffic in less than 15 sec with reliable performance. By selecting more important MIB variables by the CFS mechanism we improved the detection accuracy. Also we proposed more suitable set of MIB variables for attack detection. The selection of MIB variables as well as the attack detection mechanism highly affects the accuracy and performance of the SNMP-based attack detection.

For the future work, we are going to add more SVDD modules to our two-level structure for other types of traffic flooding attacks. Furthermore, we are planning to develop a prototype system based on the proposed mechanism for real-time attack traffic detection.

References

- [1] D. Moore, G. Voelker, S. Savage, Inferring internet denial-of-service activity, in: Proceedings of the Usenix Security Symposium, 2001, pp. 401–414.
- [2] M. Kim, H. Kang, S. Hong, S. Chung, J.W. Hong, A flow-based method for abnormal network traffic detection, in: Proceedings of NOMS 2004, Seoul, Korea, April 2004, pp. 559–612.
- [3] E. Duarte, A.L. Santos, Network fault management based on SNMP agent groups, in: Proceedings of ICDCSW, Phoenix, AZ, USA, April 2001, pp. 51–56.
- [4] J. Li, C. Manikopoulos, Early statistical anomaly intrusion detection of DOS attacks using MIB traffic parameters, in: Proceedings of IEEE Information Assurance Workshop, 2003, pp. 53–59.
- [5] L.P. Gaspary, R.N. Sanchez, D.W. Antunes, E. Meneghetti, A SNMP-based platform for distributed stateful intrusion detection in enterprise networks, IEEE J. Selected Areas Commun. 23 (10) (2005) 1973–1982.
- [6] J.B.D. Cabrera, L. Lewis, X. Qin, C. Gutierrez, W. Lee, R.K. Mehra, Proactive intrusion detection and SNMP-based security management: new experiments and validation, in: Proceedings of IM, 2003, pp. 93–96.
- [7] L. Khan, M. Awad, B. Thuraisingham, A new intrusion detection system using support vector machines and hierarchical clustering, VLDB J. 16 (4) (2006) 507–521.
- [8] S. Noel, D. Wijesekera, C. Youman, Modern intrusion detection, data mining, and degrees of attack guilt, in: Applications of Data Mining in Computer Security, Kluwer Academic Publisher, 2002, pp. 1–31.

- [9] H. Lee, J. Song, D. Park, Intrusion detection system based on multi-class SVM, in: Proceedings of RSFDGrC, LNAI, vol. 3642, 2005, pp. 511–519.
- [10] J. Zheng, M. Hu, Intrusion detection of DoS/DDoS and probing attacks for web services, in: Proceedings of WAIM, LNCS, vol. 3739, 2005, pp. 333–344.
- [11] T. Ambwani, Multi class support vector machine implementation to intrusion detection, in: Proceedings of the International Joint Conference on Neural Networks, vol. 3, 2003, pp. 2300–2305.
- [12] X. Xu, X. Wang, An adaptive network intrusion detection method based on PCA and support vector machines, in: Proceedings of ADMA, 2005, pp. 696–703.
- [13] KDD CUP DATA, 1999. Available from: <<http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>>, <<http://www-cse.ucsd.edu/users/elkan/kdresults.html>>.
- [14] J.B.D. Cabrera, L. Lewis, X. Qin, W. Lee, R.K. Mehra, Proactive intrusion detection and distributed denial of service attacks – a case study in security management, *J. Netw. Sys. Manag.* 10 (2) (2005) 225–254.
- [15] R. Puttini, M. Hanashiro, F. Mizziara, R.D. Sousa, L.J. García-Villalba, C.J. Barenco, On the anomaly intrusion detection in mobile ad hoc network environments, in: Proceedings of PWC, LNCS, vol. 4217, 2006, pp. 182–193.
- [16] K.H. Ramah, H. Ayari, F. Kamoun, Traffic anomaly detection and characterization in the tunisian national university network, in: Proceedings of Networking, LNCS, vol. 3979, 2006, pp. 136–147.
- [17] M. Shyu, S. Chen, K. Sarinnapakorn, L. Chang, A novel anomaly detection scheme based on principal component classifier, in: Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop, Florida, USA, 2003, pp. 172–179.
- [18] P. Barford, D. Plonka, Characteristics of network traffic flow anomalies, in: Proceedings of ACM SIGCOMM IMW, San Francisco, CA, November 2001.
- [19] IETF RFC 1213, Management information base for network management of TCP/IP-based internets: MIB-II, Available from: <<http://www.rfc-editor.org/rfc/rfc1213.txt>>.
- [20] D. Dittrich, Distributed denial of service (DDoS) attacks/tools, Available from: <<http://staff.washington.edu/dittrich/misc/ddos/>>.
- [21] Y. Liao, V.R. Vemuri, Use of k-nearest neighbor classifier for intrusion detection, *Comput. Secur.* 21 (2002) 439–448.
- [22] Y. Wu, A. Zhang, Feature selection for classifying high-dimensional numerical data, in: Proceedings of IEEE Conference on Computer Society, CVPR, vol. 2, 2004, pp. 251–258.
- [23] N. Williams S. Zander G. Armitage A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification, in: *ACM SIGCOMM Computer Communication Review*, 36 (5) 2006, pp. 5–16.
- [24] I. Oh, J. Lee, B. Moon, Hybrid genetic algorithms for feature selection, in: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26 (11) 2006, pp. 1424–1437.
- [25] F. Fleuret, Fast binary feature selection with conditional mutual information, *J. Mach. Learn. Res.* 5 (2004) 1531–1555.
- [26] Y. Sun, J. Li, Iterative RELIEF for feature weighting, in: Proceedings of the Twentythird International Conference on Machine Learning, 2006, pp. 913–920.
- [27] M. Hall, Correlation-based feature selection for machine learning, in: PhD Diss., Department of Computer Science, Waikato University, Hamilton, NZ, 1998.
- [28] WEKA: Data mining software in java, Available from: <<http://www.cs.waikato.ac.nz/ml/weka/>>.
- [29] J. Park, M. Kim, Design and implementation of an SNMP-based traffic flooding attack detection system, in: Proceedings of the Asia-Pacific Network Operations and Management Symposium (APNOMS) 2008, Beijing, China, October 2008.
- [30] D.-S. Yoo, C.-S. Oh, Traffic gathering and analysis algorithm for attack detection, *KoCon 2004 Spring Integrated conference*, vol. 4, 2004, pp. 33–43.