A Sampling Method for Intrusion Detection System

Zhuo Ning[1, 2], Jian Gong[1, 2]

[1](School of Computer Science and Engineering, Southeast University, Nanjing 210096, China)
[2](Jiangsu Provincial Key Laboratory of Computer Network Technology, Nanjing 210096, China)
{zhning,jgong}@njnet.edu.cn

Abstract. It is well known that Intrusion Detection System (IDS) does not scale well with Gigabit links. Unlike the other solutions that try to increase the performance of IDS by the distributed architecture, we develop a novel sampling method IDSampling whose sampling rate is adaptive to the memory bottleneck consumption to capture attack packets as many as possible by analyzing characteristics of the attack flow length and its type. IDSampling applies the single sampling strategy based on four traffic feature entropies when large-scale traffic anomaly occurs, and another complicated one instructed by the feedback of the following detection results by default. The results of experiment show that IDSampling can help IDS to remain effective even when it is overloaded. And compared with the other two notable sampling method, packet sampling and random flow sampling, IDSampling outperforms them greatly, especially in low sampling rate.

Keywords: intrusion detection system; sampling; multistage bloom filter; feature entropy

1 Introduction

The Misuse Intrusion Detection System(MIDS) is prevailing in practice for it can provide explicit alerts to users in low false rate. However, it is computationally infeasible to deal with gigantic operations of data storage and analyze in Gigabit link speed. To address this problem, many works have been investigated.

Early works focus on how to augment IDS processing power. [1~3] provide hardware solutions that enhance the performance of IDS by parallel computing, but hardware solutions are hard to be popular due to their high cost. [4,5] propose distributed IDS architectures that distribute traffic to a bunch of IDS detectors using payload balance policy and integrate several IDSs as a whole. However, the distributed IDS architecture also has several cons. Firstly, the distributing policy for IDS is required not only to distribute the traffic as even as possible but also to assign the same session to the same IDS, otherwise the context of attack will be damaged and IDS won't detect it well. So at some special scenarios, such as DDOS with the same source IP and destination IP, the single IDS detector also has the imbalance

problem as before. It indicates that the distributed IDS architecture can not solve the performance-accuracy imbalance problem radically. Secondly, it is rather hard to configure how many IDSes are reasonable to be packed as a whole because either it is suboptimal in low traffic volume or it has the same resource consumption difficulties in high traffic volume. Finally, the maintenance and updating of the distributed architecture are undoubtedly more complicated than that of an IDS.

Compared with the aforementioned work, the following addresses performance limits by sampling, but all focuses on anomaly detection[6-11]. In [6], A.Lakhina proposes a notable anomaly detection method using multiway subspace method to analyze netflow data. In [7], Mai investigates how packet sampling impacts three specific portscan detection methods, TRWSYS, TAPS and entropy-based profiling method. Recently, the work is extended to analyze the impact of other sampling schemes in [8]. It demonstrates further that the random flow sampling is better than packet sampling in anomaly detection. And the findings of [9] suggest that entropy summarizations are more resilient to sampling than volume metrics. However, it is still an open problem whether sampling solutions are sufficient in network-wide intrusion detection.

Unlike the aforementioned sampling methods which apply the traditional packet sampling or flow sampling directly, we propose a novel sampling method IDSampling which is adaptive to the consumption of the memory bottleneck to cover not only anomaly detection but also misuse detection. IDSampling profiles the traffic and applies the single sampling strategy based on four traffic feature entropies when large-scale traffic anomaly occurs, and adopts another complicated one instructed by the feedback of the following detection results by default. The aim is to capture attack packets as many as possible under restricted sampling rate. The results of experiment show that IDSampling can help overloaded IDS remain effective and it outperforms traditional sampling methods greatly.

The rest of this paper is organized as follows. We analyze the characteristics of the attack flow and conclude the sampling strategies for IDS in section 2. IDSamling is proposed in details in section 3. In section 4 experiment results is presented, then we conclude and outline the direction of the future work in section 5.

2 Sampling Strategies for IDS

Unlike the other sampling methods in traffic measurement, IDSampling doesn't care the sampling bias of traffic, but focuses on how to capture the packets belonging to the attack flow as accurately as possible and discard "good" traffic to alleviate the pressure of IDS. So it's critical to filter the packets based on attack features. There are two kinds of attack features. One is the feature in the packet header, the other is the signature lying in packet payload. What can be used in IDSampling is the packet header feature because the payload signature is usually composed of strings whose inspection costs expensively and it will prevent sampling from working in line with the link speed. Moreover, packet header features reflect the traffic feature directly, including the flow type, the flow length, the packet arrival rate and the duration of the flow. Among them the packet arrival rate and the duration are influenced by the

network environment greatly and can not reflect the inherent feature of the attack, so we choose the flow type and the flow length as metrics to exploit the attack traffic feature and propose the sampling strategies according to these features.

Usually there are less than 0.5% attack packets in network, while the figure soars when large-scale anomaly happens. For example, approximately 70% packets are attack packets in some DDOS. The possibility of discarding the attack packets is low in normal traffic, for most of the packets are clean. However, in this case if some attack packets do be discarded, the lost will influence detection rate greatly. When large-scale anomaly happens, the possibility of discarding the attack packets increases, but the traffic feature of high repetition accompanying with large-scale anomaly helps to capture attack packets. So different sampling strategies should be applied in different cases and we will discuss them separately as follows.

Large-scale anomaly bears some distinct traffic features, for example, the volume of packets and flows will soar and exhaust IDS soon. Another distinct feature is that the packets are of high repetition. So in this case it is efficient for the recovery method to make a reasonable approximation though multiplying the results by the reciprocal of the sampling rate. Experiment results of some researches [8] have proven that if the sampling rate is too low and distorts the metrics heavily, the detection rate will be too low to make sense. At this time the sampling strategy should concentrate on the most abnormal flows to guarantee the sampling rate of abnormal flows high enough to detect attacks and ignore the others for saving the limited resources.

Compared with the large-scale anomaly attack, the recovery method of other attacks can't be as simple as multiplying, for their behaviors are no longer of high repetition. These attacks conquer system via various vulnerabilities and we'll conclude the sampling strategy for them by analyzing their typical attack process.

A typical attack is composed of seven phases. ① shielding the source of the attack, ②collecting the information of victims,③exploiting the vulnerability,④breaking into the victim,⑤ clearing the attack trace,⑥launching attack,⑦executing backdoor program. At the beginning ② and ③ are usually carried out by kinds of port scan, so most attack information lies in the short flow. With the attack evolving the long flow gets to contain more and more attack information. In steps ④~⑥ the attacker tend to attempt many different methods to maximize the success possibility. This kind of redundancy makes sampling method promising in detection. Sampling the short flow in high rate at the beginning will help to identify the attack flow as early as possible with the low cost, then we can increase the sampling rate of these attack flows after signing them, and discard the others without signs. As the percent of the attack flow is rather low in this case, the sampling strategy can undoubtedly reduce the pressure of IDS.

To summarize, IDSampling applies different strategies in different cases.①When large-scale anomaly happens, IDSampling should focus on the most abnormal flows to guarantee the sampling rate of abnormal flows high enough to detect attacks. ② When large-scale anomaly doesn't happen, IDSampling should sample the short flow in high rate to guarantee to detect the attack and sign the flow in the beginning, then it can sample subsequent packets of the flow with signs in high possibility and discard the others in high possibility.

3 IDSampling Method

IDSampling profiles the traffic feature to apply different sampling strategy. So the time is divided into measurement bins for traffic statistics. Because the flow feature is of self-similarity and lone-range dependence, it's reasonable to apply sampling in current bin with the statistic results of the former one. As the processing method is the same in all bins, the latter discuss will be limited in one. The rest of this chapter is organized as follows. In section 3.1 IDSampling is introduced. Section 3.2 explains how to adapt sampling rate due to IDS bottleneck. Section 3.3 gives a detailed description of the single sampling method based on four traffic feature entropies which works when large-scale anomaly happens. In section 3.4 another complicated sampling instructed by the feedback of the detection results is applied by default. In section 3.5 the feedback methods are discussed. Finally performance analysis is provided in section 3.6.

3.1 IDSampling

①In the beginning of the bin IDSampling counts the adaptive sampling rate P by equation(1) as discussed in 3.2, then tells whether or not large-scale traffic anomaly occurs using the method discussed in [6].If the answer is yes, then turn to ②, else turn to ③.
②For each arriving packet X in the bin IDSampling applies the single sampling strategy based on 4 traffic feature entropies as discussed in section 3.3,
③For each arriving packet X in the bin IDSampling applies the complicated sampling strategy based on 4 traffic feature entropies as discussed in section 3.4.

3.2 Adapting sampling rate

The sampling rate is restricted by the bottleneck of IDS which lies in either CPU or memory. We suppose that the IDS manufacture will guarantee in its configuration that CPU will not be overwhelmed when memory is exhausted. So under such restriction whether or not the receiving buffer overflows is a signal to tell whether or not the processing rate of IDS can catch up with the input rate. And as long as the buffer overflows, the sampling method will be launched to enable IDS to work efficiently with limited resources. So the sampling rate is determined by equation (1).

P= the packets processing rate of IDS/the arrival rate of network packets **(1)**

3.3 The single sampling strategy of IDSampling

Before the single sampling strategy is proposed, firstly we'll introduce the entropy of the traffic feature as a metric to tell whether or not a flow is abnormal.

Definition1 Given an empirical histogram vector denoted as $\text{feature}_i = \{(x_i, n_i), i=1,2, \ldots N\}$, where feature_i is a traffic feature in the time bin, i.e, source IP or destination port, which means that feaure_i has N different values of x_i and each x_i occurs n_i times, the entropy of the feaure_i is defined as: $\text{H(Feature}_i) = -\sum_{i=1}^{N} (\frac{n_i}{S}) \log_2 (\frac{n_i}{S})$, while $S = \sum_{i=1}^{N} n_i$ is the total number of observations in the bin.

We focus on four feature entropies. Let H(srcIp), H(srcPort), H(dstIp) and H(dstPort) be the entropy of source address, source port, destination address and destination port. The value of them lies in the range(0, $\log_2 N$). The figure takes on the value 0 when the distribution is maximally concentrated, i.e, all observations are the same. The feature entropy takes on the value $\log_2 N$ when the distribution is maximally distributed, i.e, $n_1=n_2=\ldots=n_n$. So the metric provide a convenient summary statistic for a distribution's tendency to dispersed or concentrated. Table 1 lists a set of anomalies commonly encountered in backbone network traffic in which "↑" means the feature is becoming more distributed, while "↓" means more concentrated and "-" means uncertainty. The change of the distribution tendency caused by the abnormal is obvious according to their definitions. It is shown in table 1 that each of the abnormal affects at lest two feature entropies, and that's why the method based on the feature entropy is more accurate than those focus on traffic volume.

Table 1. Qualitative effects on the feature entropy by various anomalies

Abnormal Lable	Defination	H(srcIp)	H(srcPort)	H(dstIp)	H(dstPort)
Alpha Flows	Unusually large volume point to point flow	↓	-	↓	-
DOS/DDos	Denial of Service Attack(distributed or single-source)	↑	-	↓	-
Flash Crowd	Unusual burst of traffic to single destination, from a "typical" distribution of sources	-	↑	↓	-
Port Scan	Probes to many destination ports on a small set of destination address	-	-	↓	↑
Network Scan	Probes to many destination addresses on a small set of destination ports	-	-	↑	↓
Outage Events	Traffic shifts due to equipment failures or maintenance	↓	-	↓	-
Point to Multipoint	Traffic from single source to many destinations, e,g., content distribution	↓	↓	↑	↑
Worms	Scanning by worms for vulnerable hosts(special case of Network Scan)	-	-	↓ ↑	↑ ↓

To tell the feature entropy is normal or not, we introduce the expectation variation of the feature entropy as following.

Definition2 Given X as a random variable, $\exists$ E(X) and δ_X which stands for the expectation and the variance of X separately. Then expectation variance of X, denoted by ζ_X, is defined as $\zeta_X = \frac{|X - \text{E(X)}|}{\delta_X}$, which illustrates how far away X is deviated from E(X).

For a feature entropy i, there are two thresholds ζ_{i_1} and ζ_{i_2} which can get by training data. If $\zeta_i \leq \zeta_{i_1}$, we can tell i is normal, if $\zeta_i \succ \zeta_{i_2}$, then we can tell i is abnormal, or $\zeta_{i_1} \prec \zeta_i \prec \zeta_{i_2}$, and we are uncertain whether or not i is normal.

As discussed in section 2, when large-scale anomaly happens, IDSampling will capture the most "abnormal" traffic. In the following we will show the details and the sampling strategy is called the single one based on the 4 feature entropies.

① Choose the feature whose entropy is the smallest in 4 feature entropies. Denoted as $\text{feature}_{\text{smallest}} = \{(x_i, n_i), i=1,2, \ldots R\}$.

② Sort $\text{feature}_{\text{smallest}}$ by n_i, and compute topN which satisfies equation (2).

$$\zeta_{H(\text{feature}_{\text{smallest}})} > \zeta_{i_2} \tag{2}$$

where $H(feature_i) = -\sum_{i=1}^{N} (\frac{n_i}{S}) \log_2 (\frac{n_i}{S})$, as defined in definition 1.

③For an arriving Packet X, if $\text{feature}_{\text{smallest}}$ of X is included in topN, then samples X using packet sampling method at sampling rate P, else discards it.

3.4 The complicated sampling strategy of IDSampling

When large-scale anomaly doesn't occur, IDSampling will sample the coming packet with different sampling rate according to different flow length to which it is belonged. To find a tradeoff between efficiency and consumption, we divide all flows into three types: the short flow whose length ≤ 10, the long flow whose length ≤ 1000 and the super long flow whose length > 1000. Let P_{short}, P_{long} and $P_{superlong}$ be the sampling rate of them respectively. W_{short} and W_{long} denote the priority of the short flow and that of the long. For the reasons listed in section 2, we will not sample the super long flow any more, so $P_{superlong}$=0 and P_{short} and P_{long} are determined by equation (3) and (4) as the following. In equation(3), P is the adaptive sampling rate determined by equation(1). We adopt notable multistage Bloom Filter to count the flow length[10], and the method is not discussed for brevity.

$$P_{short} + P_{long} = P \tag{3}$$

$$P_{short} / P_{long} = W_{short} / W_{long} \tag{4}$$

In this case the sampling strategy is a complicated one which instructed by the feedback of the following detection results. The method is proposed in details as the following and its flow chart is illustrated in Fig. 1.

①For each arriving packet X, it will pass multistage Bloom Filter to tell which type of the flow it is belonged to.

②If X belongs to the flow which is signed as an attack flow by the previous packets, it will be sampled 100%. If X belongs to the short flow, it will be sampled at high sampling rate P_{short} using packet sampling method. If X belongs to the long flow, it

will be sampled at low sampling rate P_{long} using flow sampling method, or X will be discarded for it belongs to the super long flow.

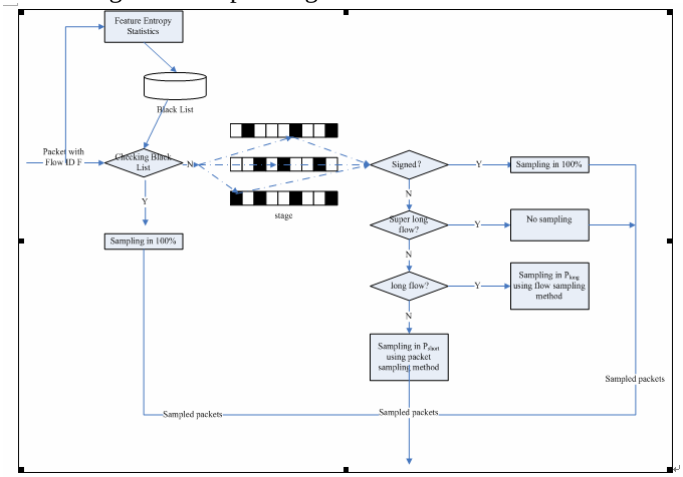

Fig. 1. The flow chart of the complicated sampling strategy

3.5 The feedback method of IDSampling

IDSampling achieves a nice accuracy for it is instructed by the detection results which are reported by the detection engineer. Aiming at communicating with the detection module efficiently, we propose a feedback method which can sign the attack flow in line with the link speed. As the sampling method is different with the flow length, the feedback method also varies with the flow length.

The feedback method of the long flow is to sign the hash counter of the Bloom Filter which works in the same way as counting the flow length. When a packet X is confirmed as an attack packet by the detection engineer, in every stage of the Bloom Filter a hash on its flow ID is computed and the corresponding counter is signed(in bolded). Since all packets belonging to the same flow hash to the same counter, X will be confirmed as a subsequence of the attack flow if all the X's counters are signed and it will be sampled at rate 100%. Fig 1 has illustrated the above feedback method.

However, the aforementioned feedback method is absolutely useless in the short flow. Firstly, the last time of the short flow is very short, usually scales in several million seconds. So the short flow will be ended entirely before the hash feedback method begins to work. Secondly, the amount of the short flow is too large to sign, or the error positive of the multistage Bloom Filter will be too high to work. Finally, IDS can not afford so much communication information between the detection module and the sampling module. Considering the fact that the attack in the short flow is of high repetition, so clustering information is efficient to capture such a characteristic. The reasonable feedback method of the short flow is to put the topN of the $feaure_{smallest}$ into the blacklist in each time bin and the packet with the $feaure_{smallest}$ in the blacklist will be sampled in 100% rate. The maintenance of the blacklist is neglected here.

3.6 Performance Analysis

The memory consumption of IDSampling is composed of two parts mainly. One part is consumed by the multistage Bloom Filter, while the other is done by the statistics of the traffic feature entropy. Denote by b the number of the hash counters of each Bloom Filter stage, let p and n be the number of active packets and flows, and N presents the topN which is decided by equation(2). Thus, the total memory of multistage Bloom Filter is O(bd), and that of the statistics is O(n). As the speed is concerned, the performance of IDSampling is discussed in different cases. When large-scale anomaly doesn't occur, IDSampling will process each packet in O(1), while the preprocess of the statistics costs O(n) in the beginning of each time bin. When large-scale anomaly do occur, IDSampling will process each packet in $O(\log_2 N)$ for each packet will check the black list first. And in this case except for the ordinary statistic processing which costs O(n), the preprocessing will also count the topN which costs O(nlogn), so the preprocessing time of each bin is the sum of the two phases which costs O(n+nlogn).

4 Experimental Results

Our experiments are conducted using Snort as IDS and DARPA 1999, a notable dataset for evaluating IDS, as the data source. We train IDSampling by the inside tcpdump files of the first and the third week, then test it by the file of the fourth week. To evaluate the accuracy, we use Snort to detect the traffic sampled by IDSampling to get alerts denoted as $Result_{sampling}$. In the same way we can get intact alerts denoted as $Resul_{all}$ by detecting the non-sampled traffic. Then $Result'_{sampling}$ and $Result'_{all}$ are generated from $Result_{sampling}$ and $Resul_{all}$ separately after redundancy eliminating. The accuracy of IDSampling can be measured by $accuracy_{dection}$ which equals $Result'_{sampling}$/ $Resul'_{all}$. Another thing worthy to mention is that $accuracy_{dection}$ is lower than the accuracy that IDSampling actually achieves because of the limit detection power of Snort. So we will adopt some recovering methods. Compared to the labeled attack list, we will consider the attack is detected successfully already if the amount of its sampling data is greater than the threshold.

Fig 3 plots the detection rate of every day in the week 4 at different sampling rate. It is shown that the detection rate descends monotonously with the sampling rate. The detection rate(D_r) is rather high at high sampling rate ($\leq$ 1/5), the max D_r reaches 97.8% at 1/2 sampling rate and the lowest is 75.9% at 1/5. As the sampling rate drops, the D_r falls down monotonously, but the dropping speed varies in different day. The attenuation of Dr is quite large in the Monday, Tuesday and Thursday, while that of the Wednesday and the Friday varies little. For example, in the Tuesday the Dr drops from 97.8% at 1/2 sampling rate to 56.5% at 1/100 sampling rate, while the figure of the Friday is 96.8% and 84.1 respectively. By analyzing labeled attacks we confirm that attacks of the Monday, Tuesday and Thursday are scattered ones with a few packets($\leq$100) and their last time are short. So the attack is distorted heavily when the sampling rate drops to 1/100 and the attack will be missed entirely. However, there are a large-scale R2L attack in the Wednesday and a large-scale Probe attack in

the Friday. So the traffic repetition and the clustering information help to maintain D_r even in low sampling rate. To summarize, IDSampling scores pretty good D_r at high sampling rate and it is better suited to the large-scale anomaly.

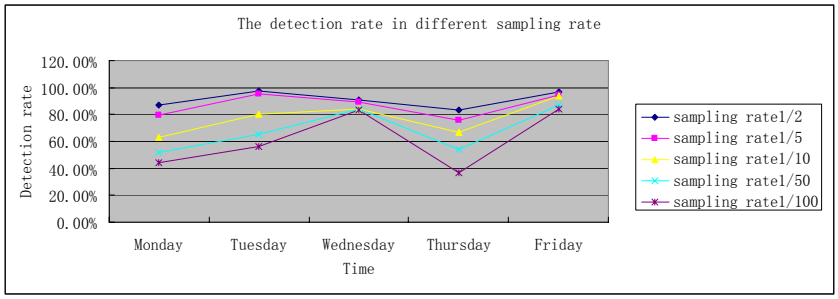

Fig. 2. IDSampling detection rate in different sampling rate of the fourth week

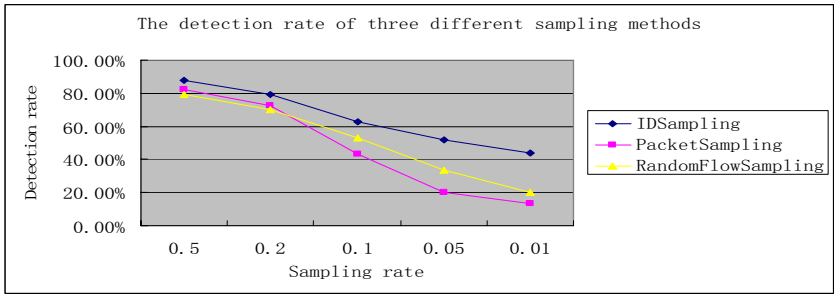

Fig. 3. The **detection rate of three different sampling methods**

To evaluate how efficient IDSampling is, Fig 4 compares the D_r of it to that of the other two notable sampling schemes, packet sampling and random flow sampling, in different sampling rate using the inside tcpdump file of the Monday as a trace. It is indicated that all three sampling methods seem to affect the detection in a similar manner, however, their relative impact on the degradation of D_r is quite different. IDSamling outperforms the other method in all sampling rates and it is the most robust one. It drops by 43.8% from the sampling rate 1/2 to 1/100, and the drops from the other two method are 69.2% and 58.8 separately. The D_r of packet sampling falls to 13.4% and that of random flow sampling is 20.3% when the sampling rate decreases to 1/100, while IDSampling still remains 44%. To summarize, IDSampling is the most efficient one for it is under the help of the feedback of the following detection result and other clustering information of the traffic feature entropy. So the D_r of IDSampling is pretty higher when sampling rate is low. The performance of the packets sampling stands the lowest and the random flow sampling lies in the middle. That's because the random flow sampling samples the flow without bias and so it can get more accurate information about IPs and ports than the packet sampling, but the packet sampling absolutely leans to the long flow.

5 Conclusion

In this paper we employ a novel sampling method IDSampling which samples the packets of the most abnormal flow with the help of the traffic feature entropy when large-scale traffic anomaly occurs, and incorporates dynamic feedback from the detection engine to further maximize the possibility of capturing the attack packet successfully. To address the performance limits of IDS, it is also a cost-effective yet scalable solution which can work in line with Gigabit links. The experiment results show that IDSampling is well suited to extract the attack packet and the detection rate of it in large-scale anomaly is higher than that of in other cases. Anyway it is pretty nice in high sampling rate(<1/10). However, if the sampling rate is relatively low compared with the attack scale, the detection rate of IDSampling will drop due to heavy metric distortion. In another word, IDSampling can not guarantee the performance of IDS.

Our ongoing work is centered on extending IDSampling to be a sampling method with detection accuracy guarantees. In particular, we are studying additional information that can aid in better detecting anomalies by their root-cause, analyzing the mathematics model of it and investigating how much information is necessary for recovering an attack from the sampling data.

Reference

1.H. Bos and Kaiming Huang, Towards Software-Based Signature Detection for Intrusion Prevention on the Network Card. RAID 2005,Vrije University, The Netherlands Xiamen University, China

2. Y. Cho and W. Mangione-Smith. Fast reconfiguring deep packet filter for 1+gigabit network, In IEEE Symposium on Field-Programmable Custom Com[putting Machines, (FCCM), NaPa, CA, April 2005

3. R.Fanklin, D.Caraver, and B. Hutchings. Assisting network intrusion detection with reconfigurable hardware. In Proceedings from filed Programmable Custom Computing Machines. 2002

4. Xun xun Chen, Bing xin fang, The architecture of Intrusion detection system in high-speed network, Computer research development, [J]. 2004 Vol.41 No.9 P.1481-1487

5. I.Charitakis, K.Anagnostakis, and E.Markatos, “An active traffic splitter architecture for intrusion detection,” in Proceedings of 11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS 2003), Orlando, October 2003, pp. 238–241.

6. A.Lakhina, M.Crovella, and C.Diot. Mining Anomalies Using Traffic Feature Distributions. In Proc. ACM SIGCOMM ’05, Philadelphia, PA, USA, Aug. 2005

7. J.MAI, ,SRIDHARDAN, a.,Chuah, C.N, Aang, H., Impack of packet sampling on portscan detection. IEEE Journal on Selected Areas in Communication(2006).

8. J Mai, CN Chuah, A Sridharan, T Ye, H Zang, Is sampled data sufficient for anomaly detection? In Proc. of the 6th ACM SIGCOMM on Internet measurement, Brazil, 2006.

9. D Brauckhoff, B Tellenbach, A Wagner, Impact of packet sampling on anomaly detection metrics**,** In Proc. ACM SIGCOMM’06 Rio de Janeriro, Brazil,2006

10. Estan C, Varghese G. New Directions in Traffic Measurement and Accounting[C].In: SIGCOMM2002, August 2003, Pages 270–313.