

# Hardware Supports for Network Traffic Anomaly Detection

Dae-won Kim and Jin-tae Oh

Electronics and Telecommunications Research Institute in Korea

**Abstract** - Modern network systems are plagued with unknown attacks every time. To detect these attacks, network security systems monitor the anomalous status of network traffics. Most of researches have focused on the software-based anomaly detection and, thus, they have suffered from huge performance declines in high speed and bandwidth networks over one giga bps on one side. Due to the potential performance overhead, they are difficult to frustrate network attacks in real-time. To solve this problem, the researches about network anomaly detection must change from the only software-based to the combination-based of hardware and software. In this paper, we present the hardware designs for supporting the traffic anomaly detection of network security systems, which are IDS and IPS, in the high speed network. First advantage of these designs is that to some degrees, they overcome limitations of hardware memory and gather effectively and quickly network traffics which require for the software anomaly detection. Second, in hardware level, they can determine whether the statuses of current networks are DoS attacks or Worm spreading.

**Keywords:** Network Security, IDS, IPS and Anomaly Detection.

## 1 Introduction

Networks have rapidly grown up and for the security their domains have been complicated [1, 11]. However, attacks have also been more elaborated and the damages occurred by the attacks targeted on these networks also have been rapidly increased [6, 7, 10]. The typical damages are various kinds of service hindrances and the typical attacks that cause the damages exhaust CPU and memory of victims and network bandwidths. To prevent these attacks, many researches have been studied and some of them [2, 3] are practically effective on the well-defined attacks and some abnormal traffics.

In network research fields, many researchers have interested in the traffic anomaly analysis. These fields have important means on the two sides of network management and security. Particularly, on the side of security, researchers have tried to find the anomalous and new attacks that can evade the existing security systems. However, to avoid these systems, the attacks have been

gotten more complex and various [4]. Therefore, the existing systems have some detecting limitations about new attacks and to overcome these limitations, the network anomaly detection has been actively researched.

The more networks grow up, the more the number of portal sites with many users increase. Therefore, it is sure that the new attacks targeted on these sites will enlarge the scope of damages. DoS attacks [5] and Worms [9, 12] are most threatened to the point of network security among many types of traffic anomalies. To prevent these attacks, many researches have focused on the traffic anomalies and have developed security systems. The administrators of large scale sites are seriously interested in these attacks and hope the network security systems to protect their servers in real-time [8]. Because there are the continuous demands about the unknown attacks from the network operators, a variety of the academic world and laboratory has researched the systems for detecting traffic anomaly.

Main direction of their researches have been processed to increase the detection ability and to decrease the false positive [16]. As these reasons, most of them have too many complex algorithms. Because of these complexities, the software-based researches have become the main parts of anomaly detection and researches to combine hardware and software were not the main trend of anomaly detection. However, the hardware support for anomaly detection has important roles to enhance the detection performance in the high speed network.

In this paper, we propose the hardware designs for supporting the network anomaly detection of security systems (IDS and IPS) in the high speed network. Our approaches have two main advantages. First, within the limited hardware memory, it gathers effectively network traffics which are required to detect anomalies as software. Second, in the hardware level, it can inform statuses of current network threats (DoS attacks and Worm spreading) to the software. Its information could be used to control detection sensitivities of software.

This paper is organized as follows. In Section 2, we present the hardware design concepts to overcome the software-based anomaly detection. Then, in Section 3, we introduce the overview of hardware designs and, in Section 4, 5, in

detail; we describe the methods to manage the hardware memory and to detect in advance the anomalous status of network in hardware. Finally, we conclude in Section 6.

## 2 Hardware Supports for anomaly detection

To run anomaly detection using network traffics in the software, the information should be periodically collected. The computing power of cost-effective network security system is not so strong. Because analyzing traffic anomalies is complex, the software requires considerable computing power. Moreover, if it is operated in high speed network, the detection software can not apply detection algorithms to every incoming packet because of its performance [14, 15]. In that network, if the algorithms are processed about all incoming packets, some packets are disappeared in buffers of NICs or kernel sockets due to the full of buffer. If the software runs detection with incorrect traffic information, the detection results will be also incorrect. Therefore, to solve the problems of more correct traffic information and software overhead reduction, the network security system gathers periodically all incoming traffic information and should operate detection algorithms with the information.

To collect all traffics without the packet-missing, the network security system should do it through hardware. If the software has huge memory and uses the array indexing through some hashing algorithms for searching the desired entry of a packet, the packet information may be gathered correctly on the software memory without the packet-missing. However, in actuality, the construction of such powerful system is not easy due to the costs. Moreover, if the detection processes and traffic collection are simultaneously processed by softwares, the probability of packet-missing can not be predicted by the detection software. In this case, like above paragraph, the detection results can not be trusted. The traffic collection through hardware does not occur uncertain the packet-missing like the case of software. Because the performance of hardware is determined from the design stage of system, its performance is constant and, therefore, can be absolutely trusted.

When the traffic information is collected through the hardware, the hardware memory should be managed by considering normal features of network attacks because the size of hardware memory has even more constraints than the software. Network traffics should be periodically gathered to reduce the software overhead and to overcome the memory limitation. If this period is short, the traffic information is not useful and if long, the hardware memory is full in that period. The period to collect at least correct TCP session information should be maintained if the information gathered by the hardware is useful [13]. In high speed networks, if the hardware memory is used as

simple structures to gather traffic information in a short period, the memory will be full in many periods. To solve the problem of memory full, analyzing the analysis of attack features is required. The one important feature of network attacks is that excessive traffics are generated when attacks are happened. As analyzing this feature, at least attack traffics should be managed not to disappear on hardware memory every period. To accomplish this goal, the hardware memory to gather network traffics requires to be controlled by using LRU (Least-Recently Used).

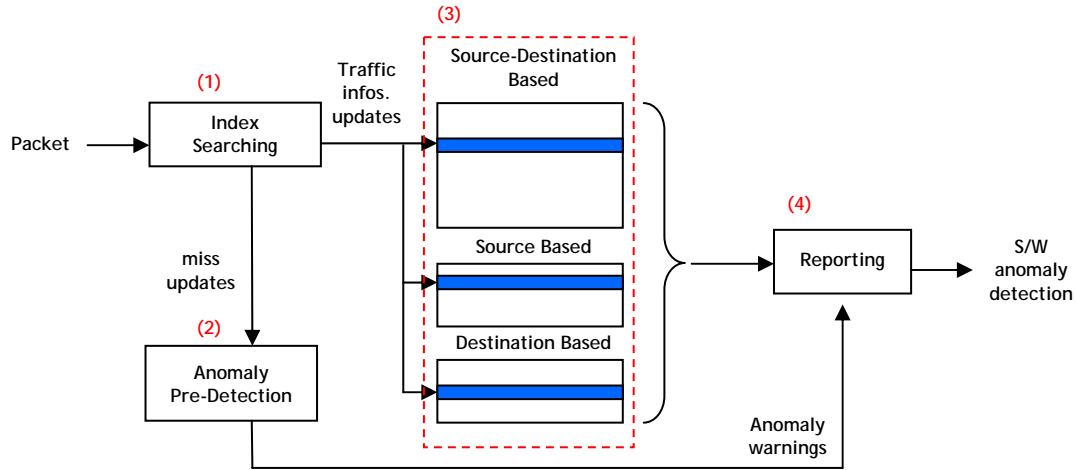
To enhance the performance of real-time responses about network attacks, traffics of the source-based and the destination-based, which are the 3-tuple-based, and the source-destination-based, which is the 5-tuple-based, may be required to be gathered. The 5-tuple-based information is most basic data to detect anomalies as softwares. When network attacks are happened, if the software detects anomalies using the information, it can know exact relations between attackers and victims. Therefore, the system can applies best effective defense policies according to the results. The reason that the 3-tuple-based should be also collected is to enhance the performance of real-time response. Normally, to run anomaly detection algorithms, the software extracts the 3-tuple-based from the 5-tuple-based. This is the software overhead to reduce the response time of system. To collect this information through the hardware decreases the software overhead and, additionally, in hardware level the initial status of DoS attacks and Worm spreading can be detected.

The three partly duplicated information of traffic information in the limited hardware memory decreases the number of traffics that can be collected every period. It is sure that in this case, the fast response of system and the number of traffics to be gathered are the relation of trade-off. However, if LRU to manage the hardware memory is used like the above paragraph, although some parts of collected information are replaced in memory entries, the attack traffic information can be maintained on the memory. Therefore, to collect the traffic information of three types is more valuable at the point view of system.

## 3 Design overview

In Chapter 2, we asserted the followings for the supports of hardwares to detect traffic anomalies from the software.

- To support the traffic anomaly detection of software, the hardware should periodically gather the incoming traffic information.
- If the network security system collects traffic s through the hardware, that information is trusted because the hardware is designed not to miss packets from the initial design.



**Fig. 1. The design overview**

- For the fast responses of security system, both the 3-tuple-based and the 5-tuple-based are required.

In this Chapter, on the base of above sentences, we describe the system overview to support the anomaly detection of software. It is presented as processes from the time that a packet was come into the hardware to the report to the detection software.

In Fig. 1, if a packet comes into the system, the logic (1) to search the position of hardware memory about the packet is processed. As the results, the memory indexes of the 3-tuple-based and 5-tuples-based are searched (hit), or the new index is allocated (miss). The miss information is accumulated to the logic (2) to detect, in advance, anomalies in the hardware level. If the desired memory is hit, the information of current packet is added, if miss, the information of current packet is newly written at the new position of memory. Whenever packets come into the system, these processes are repeated.

In the case of logic (2), from the start time that the hardware logic is operated, every period it sends total miss count during a period to the report logic (4). Additionally, if the miss count is over a specific threshold, which can be dynamically determined, the logic (2) sends the anomaly warning signals (DoS attacks or Worm spreading) to the report logic (4). This miss count is initialized every period and the threshold count is updated to new one.

The collected traffic information on hardware memory is read by the report logic (4) on each entry every following time.

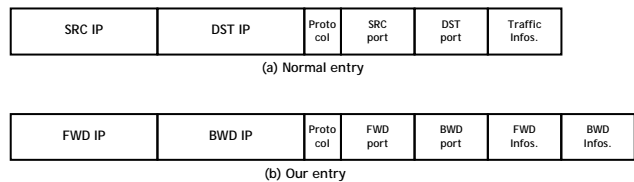
$$\frac{\text{Sum of the 3-tuple and 5-tuple entries}}{\text{Period}}$$

After then, the reported entry is initialized, the information of all entries is read by the report logic (4) during a period.

## 4 Management of hardware memory

When network traffics are collected through the hardware, the best serious problem is the shortage of hardware memory. Although the memory is insufficient, the reason that traffics should be gathered as three kinds was represented in Chapter 2. In this Chapter, using the limited memory, the methods which are efficiently collect and manage traffics are described.

### 4.1 Bi-directional entries for collecting a pair of packets



**Fig. 2. Comparison of an entry to collect traffic information on the hardware memory**

In Fig. 2, in the case of 5-tuple-based collection, normally one entry is managed like as (a) and, (b) proposed in this paper is more efficient than (a). (b) does not identify source and destination, and it identifies them only as forward and backward. Because network traffics are transferred between two hosts, (a) requires two entries to collect traffic information and (b) requires one. At the point view of network security, the identification of source and destination is not required, and only the identification of attacker and victim is. It means that the detection software

requires that only traffic information transferred between two hosts is reflected to the algorithm. Finally, (b) has more available memory as well as the size of one 5-tuple than (a). In an entry when H is the 5-tuple size, P is the information size and K is the number of host pairs, the more available memory is the follow.

$$\begin{aligned} & \{2_{entries} \cdot K \cdot (H + P)\}_{(a)} - \{K \cdot (H + 2P)\}_{(b)} \\ & = K \cdot H \end{aligned}$$

To collect the 5-tuple information of packets, if the header of current packet and the 5-tuple of (b) are matched, its information is collected in FWD infos. If the header of current packet that the sequence of source and destination is reversed and the 5-tuple of (b) are matched, its information is collected in BWD infos. To use the management method of (b), entry identification rule is required to avoid entry collision. Simply, the IP of lower value between source and destination IPs of current packet can be determined as forward. If two IPs are same, they are certainly attack packets. Therefore, they are not required to be collected and should be alarmed and dropped.

## 4.2 Management of hardware memory

The hardware memory is always insufficient and, therefore, it can not collect all traffics through network security system because of the memory cost and the available space on the system board. If the security system decreases the collection period, it can collect traffics as many as possible using the insufficient memory. However, if anomalies are detected with the traffic information collected during short period, in Chapter 2 we presented that its results can not be trusted.

If the hardware memory is managed as simple arrays, when all entries are full, the gathered attack traffic information is replaced with other new traffics. The goal of network security system is to find network attack traffics and to find them the attack traffic information is required, not normal traffics. When attacks are occurred, in current period the amount of attack packets increases rapidly. It means that the frequency of attack packets through the security system also increases. Therefore, we manage the memory to collect traffics as LRU to maintain them for a long time on the memory. Finally, when all entries are full and a new packet comes into the system, an entry of the lowest frequency is replaced with new traffic information. It is because the lowest frequency means that the relation with attack traffics is the lowest. In Fig. 3, head is the entry of lowest frequency. It shows that the entry of lowest frequency 1 is replaced with new 5.

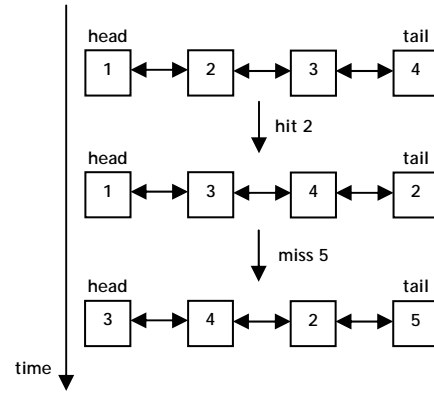


Fig. 3. Examples of entry management

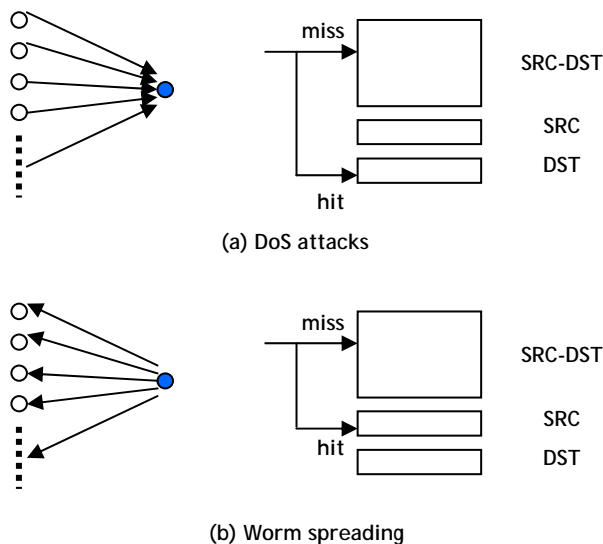
## 5 Anomaly pre-detection

In the network security, the best threats are DoS attacks and Worm spreading. On the side of network security, these attacks have common features. Excepting for special attacks, in the case of DoS attacks, attacks are accomplished through the source IP randomization and the generation of a great quantity of source IPs. As well, in the case of Worm spreading, the spreading is achieved through the destination IP randomization and the generation of many destination IPs. The reason that massive random IPs should be generated is because attackers do not know IP statuses of networks that are targeted to attack. Especially, if the target hosts and networks are selected randomly, these features are stronger. Finally, in a period, when network attacks are happened, the dispersion level of IPs and flows is heightened.

The algorithms to detect network anomalies are too various and the detection parameters are also. According to the goals of algorithms, those structures can reflect dispersion parameters to the detection or not. However, an important fact is that without the kinds of detection algorithms, the dispersion parameters can be components which require detecting anomalies. In softwares, to find the dispersion level produces many overheads. In Fig. 1, this logic (2) can not precisely analyze dispersion status as the level like softwares. However, results from the logic (2) are sufficient to show missing the right status of network.

Normally, the sensitivity of detection algorithm has influence on the amount of false positive on alarm reports of security system. The main reason that these false positives are occurred is because without grasping whether the network is right status or not, the detection algorithm of same sensitivity is processed. To reduce these false positives, the software runs the detection of appropriate sensitivity in normal network status and, if it receives warnings generated from the logic (2), they can run more sensitive detection in suspicious network status.

Fig. 4 shows the results of the index search logic (1) when the DoS attacks and Worm spreading are happened. The DoS attacks [18] in (a) are achieved as numerous sources transfer a large number of packets to a few destinations. In that case of DoS attacks, if the logic (1) tries to get indexes to find the desired entry of current packet, in the case of destination entry the hit probability is high and in the case of source-destination entry the miss probability is high. Therefore, the status that the dispersion level of source increases can be chased in hardware level. However, because this status is the same with the case of flash crowds that normal accesses are concentrated on specific servers, the exact result should be determined from the analysis of software.



**Fig. 4. The search of desired entry in the anomalous status of network**

The Worm spreading [17] in (b) is achieved as a few sources transfer packets to a great number of destinations. In the case of Worm spreading, if the logic (1) tries to get indexes to find the desired entry of current packet, in the case of source entry the hit probability is high and in the case of source-destination entry the miss probability is high. Therefore, the status that the dispersion level of destination increases can be chased in hardware level.

The logic (2) accumulates and analyzes the miss results received from the logic (1), and if the analyzed results are out of thresholds that are dynamically determined through some algorithms, the logic (2) sends warning signals to the report logic (4). Because the algorithm that determines dynamically thresholds is not main topic of this paper, it is not mentioned in this paper. Finally, the hardware-based anomaly pre-detection reduces software overheads and can immediately detect the anomalies of network.

## 6 Conclusions

In this paper, we introduced the entry management on hardware memory and the pre-detection of network anomaly, two hardware logics for supporting network anomaly detection of software. These logics are complementary to the software-based anomaly detection, which has large overheads to collect traffic information. By supporting traffic collections and early warnings, the logics provide the enhancement of performance and detection quickness of software.

The first part of this paper provided motivations with regard to the software-based problems and solutions. We motivated two approaches using hardware logics for solving them. The second part described an overview of hardware supporting functions. It presents important processes from the point that a packet came into the security system to the point that results are reported to software. The third part described the management methods of information entry and the network anomaly pre-detection that analyzes the miss count of information entry.

We hope that we were able to introduce a comprehensive overview and proposes, which are different ways with techniques that are currently used. However, we also feel the limitation of researches because in the field of network security, some absolute solutions can not be existed due to the various features of network. Instead, it is sure that more challengeable researches always exist because the attack and defense are the relation to chase and to be chased.

## 7 References

- [1] B. Alarcos, M. Sedano, and M. Calderon, "Multidomain Network Based on Programmable Networks: Security Architecture," *ETRI Journal*, vol.27, no.6, Dec. 2005, pp.651-665.
- [2] Bro Intrusion Detection System, <http://bro-ids.org>, 2006.
- [3] Snort, <http://www.snort.org>, 2006.
- [4] J. Mirkovic and P. Reiher, "A Taxonomy of DDoS Attack and DDoS Defense Mechanisms", In *Proceedings of the 2004 ACM Conference on Computer and Communications*, Washing-ton D.C., U.S.A., Oct. 2004.
- [5] A. Hussain, J. Heidemann, and C. Papadopoulos. "A Framework for Classifying Denial of Service Attacks". In *ACM SIGCOMM*, Karlsruhe, Aug. 2003.
- [6] D. Moore, C. Shannon, and J. Brown. *Code-Red: A Case Study on the Spread and Victims of an Internet Worm*.

- In Proceedings of the ACM Internet Measurement Workshop, Nov. 2002. Conference and Exposition (DISCEX), pages 303–314, Apr. 2003.
- [7] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. The Spread of the Sapphire/Slammer Worm. *IEEE Security and Privacy*, 1(4), July 2003.
- [8] D. Moore, C. Shannon, G. Voelker, and S. Savage. Internet Quarantine: Requirements for Containing Self-Propagating Code. In *IEEE Proceedings of the INFOCOM*, Apr. 2003.
- [9] S. Staniford, V. Paxson, and N. Weaver. How to own the Internet in Your Spare Time. In *Proceedings of the USENIX Security Symposium*, Aug. 2002.
- [10] Computing Research Association. CRA conference on grand research challenges in Information Security and Assurance, Nov. 2003.
- [11] V. Yegneswaran, P. Barford, and J. Somesh. Global Intrusion Detection in the DOMINO Overlay System. In *Proceedings of the ISOC Network and Distributed Systems Security Symposium (NDSS)*, 2004.
- [12] Z. Chen, L. Gao, and K. Kwiat. Modeling the Spread of Active Worms. In *Proceedings of IEEE INFOCOMM*, volume 3, pages 1890 – 1900, 2003.
- [13] K.-A. Kim and B. Karp. Autograph: Toward Automated Distributed Worm Signature Detection. In *Proceedings of the USENIX Security Symposium*, Aug. 2004.
- [14] J. W. Lockwood, J. Moscola, M. Kulig, D. Reddick, and T. Brooks. Internet worm and virus protection in dynamically reconfigurable hardware. In *Proceedings of the Military and Aerospace Programmable Logic Device Conference*, Sept. 2003.
- [15] J. W. Lockwood, “Evolvable Internet Hardware Platforms”, *Evolvable Hardware Workshop*, Long Beach, CA, USA, July 12-14, 2001, pp. 271-279.
- [16] A. Lakhina, M. Crovella, and C. Diot. Diagnosing Network-Wide Traffic Anomalies. In *ACM SIGCOMM*, Portland, Aug. 2004.
- [17] S. Schechter, J. Jung, and A. Berger. Fast Detection of Scanning Worm Infections. In *Seventh International Symposium on Recent Advances in Intrusion Detection (RAID)*, Sophia Antipolis, France, Sept. 2004.
- [18] L. Feinstein, D. Schnackenberg, R. Balupari, and D. Kindred. Statistical Approaches to DDoS Attack Detection and Response. *DARPA Information Survivability*