Contemporary Engineering Sciences, Vol. 9, 2016, no. 14, 647 - 653
HIKARI Ltd, www.m-hikari.com
http://dx.doi.org/10.12988/ces.2016.6453

A Method of the First-N Packets Storage by Using the Advanced-PCA

Seung Min Ryu[*], Si Young Lee[**], Chang woo Ryu[**] and Seong Gon Choi[*1]

[*] Department of Radio and Communication Engineering
Chungbuk National University, Korea
[1] Corresponding author

[**] Xabyss Inc., Korea

Abstract

This paper focuses to reduce the search speed of the packet and the storage capacity in the network detection. The Packet Capture Appliance (PCA) has the storage problems because of the overload of pre-processing and the storage of BigData. We propose a method of storing the First-N packets. So it improves the search speed and the storage capacity efficiency by storing the first n packets of the session. This method shows an efficiency of about 99.319% storage space compared with the full packet capture method.

Keywords: Full packet, Advanced-PCA, Storage Efficiency

1 Introduction

In the recent, there is a growing interest in network security with the rapid development of network environment. The IDS (Intrusion Detection System) is a system for monitoring events, detecting intrusion and counteracting. The IPS (Intrusion Prevention System). Monitors the network to block the threat of network in the real time. The two methods are difficult to detect as history data. It is necessary for the storage of information in the history data. It cannot correspond to the current overflowing data because the general data storage method stores full packet. So we need a new way to store the data.

A high-performance software solution does not lost the packet capture and the optimum CPU usage in the multi-gigabit networks. Although this paper proposes the processing of the network packet in the gigabit network, it did not suggest a way to store the method [1].

The embedded network intrusion detection systems achieve the packet processing rate with 1.48mpps. But it was written only with respect to the possible current intrusion detection. It did not propose the method with a way to intrusion of unknown new network method [2].

A custom full packet capture system is described for the benefits of full packet storage. The problem is difficult to secure sufficient storage space because it stores all the details of information [3].

The exiting method is difficult to prevent invading the network and the new network threats. Although it has a way to the turning past time to store the full packet, the full packet storage method is the difficulty to store a huge amount of network traffic in the current network.

In the existing paper, the method stores the First-N packets to develop the performance of traffic classification system. It is sufficient with the 5 packets [4]. We need how to store the proper packets.

To solve the storage problem, we propose a method to store some packet having session and flow information in some packet data or streaming data according to the requirements of the application.

The proposed method has the characteristic stored differently depending on the user's needs. The result shows the 99.319% efficiency compared to the existing method by utilizing the proposed method.

This paper is organized as follows. Section II describes the problem of the PCA. Section III describes the proposed method. Section IV compared the proposed method and the existing method. Section V is conclusion.

2 Related work

The DashCap architecture is a method of directly processing the application on the user's NIC Card via DMA_MAP of libDash Cap and Kernel Space in the user space. It connects the address space of a couple with a direct connection to the Tx ring and Rx ring and DMA_MAP. This proposed method disappears the addressing connection CPU from the kernel space and it reduces the overhead of CPU to enable multicore processing [1].

The embedded network intrusion detection systems propose the flow ring architecture consisting of flow controller and ring buffer. It makes the ring by flow in the kernel space and is assigned the core by ring. The packet is disposed to the core because it is treated with the same core. This proposed method has the effect which is the CPU usage reduction and disappearing the drop packet. In this paper, it can use the network of 10Gbps, but there no proposal for a secured storage space and storage method [2].

The full packet capture is the best way to secure an already transmitted packet analysis. It can confirm the generation information of problem at any parts and

any point because all the packet is captured. However, it is very difficult to secure sufficient storage space [3]. In particular, it needs the time consuming to process the large data as the storage of capacity from about 30TB by one day in the case of 10Gbps.

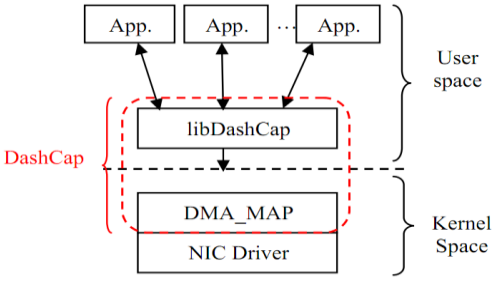

Figure 1. The architecture of DashCap [2]

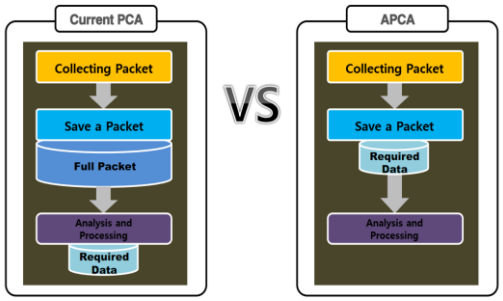

Figure 2. The system architecure of Current PCA and APCA

3 Advanced PCA

We need the same benefits such as the full packet capture method and the method which can improve a processing time to solve the storage space problem. An Advanced-PCA can improve the processing speed to reduce the storage capacity in the existing PCA.

The existing PCA collects the packets and gets the necessary data after the capturing full packet via the packet analysis and processing. After we check each method about the proposed architecture.

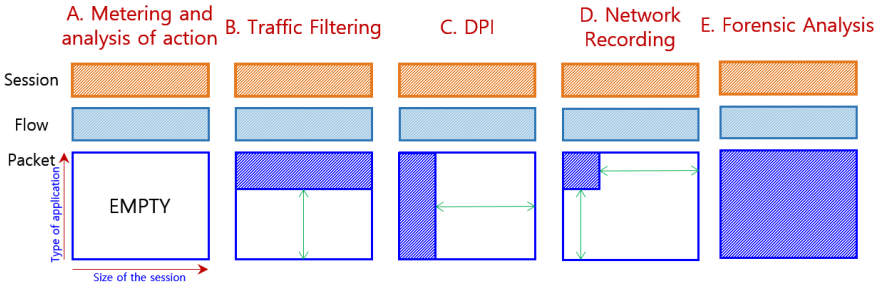

Figure 3. The proposed 5 types

In proposed method, user requirements regarding data storage are classified with five type.

- type A: It is suitable for using the abstract network data such as Metering / Charging / Billing. Traffic Pattern Analysis and Activity Analysis.
- type B: It is suitable for no storing all the application data such as Cloud-based Web service.
- type C: It is suitable for no storing all the application data such as the application type analysis via the DPI.
- type D: The optimizing network big data is stored (Type B + Type C)

- type E : The PCA

We propose the type D which is type B + type C. The type D apply to the DPI (Deep Packet Inspection) technology and DFI (Deep Flow Inspection). It recognizes the First-N packets to apply the DPI technology and the DFI technology in the real time. It changes the amount of variable data depending on the application.

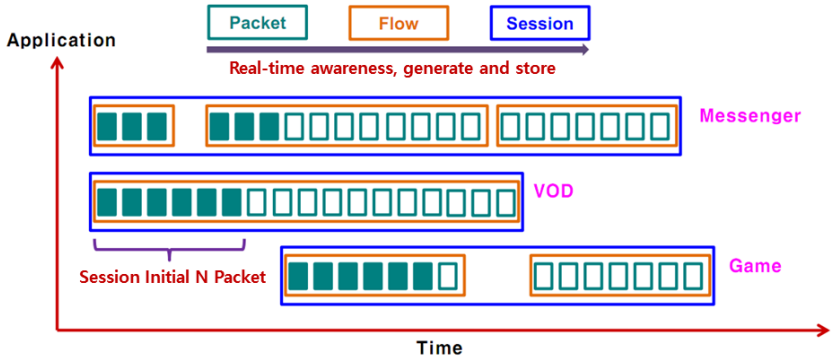

Figure 4. The packet capture method

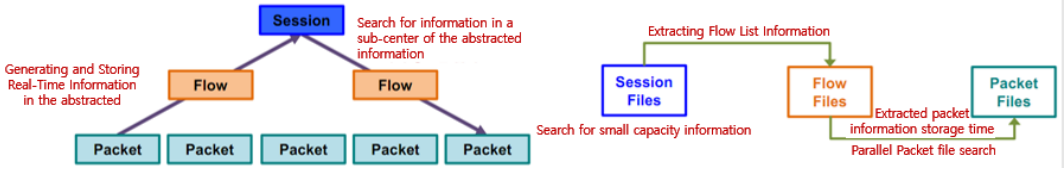

Figure 5. The methods of the packet capture and search

Figure 4 shows the flow of the message. The flow has the different sizes in the session. We consider to store the first 6 packet of the session. The packets are sent by the 3 packets of the first flow, the 11 packets of the second flow and 7 packet of the third flow. It stores the initial packets of the session with the 3 packet of the first flow and the 3 packets of the second flow.

We store the First-N packets of the session without session relationship. It is best way to store the number of packets in the initial 5 packets of the session. If it checks the initial 5 packet of the session, it can check the network signature [4]. However, we evaluate the first n packets of the session because the stored packets are many cases exist in the network.

The flow information is created and stored on the basis of a number of the received packets. This flow is a set of the continuously delivered IP packets. By comparing the 5-tuple (source address, destination address, source port, destination port, protocol), if it has the same value, it is continually updated. If there is no same value, it generates the new flow information. Based on the plurality of the flow, it generates the session information. This information is created and stored in real time.

The packet search method uses the stored information retrieves the correspondent session. And it searches the correspondent flow in the extracted

flow list information. So it takes the parallel file searching and it searches the proper packet by using the researched flow.

4 Implementation Result

The computer was used for a total of n. We implement each of the computer on downloading the files, playing the game and watching the stream in the general network and confirms the data storage efficiency.

Table 1. The Initial Session Packet Storage Size(Kb) of N

File name	Transmitted packet	Full Packet	8 packet	5 packet
Filedown1	2960166	2960166	2871	2273
Filedown2	2764320	2764320	2640	2065
Game1	63389	63389	954	776
Game2	37919	37919	832	692
Stream1	2021184	2021184	9622	8065
Stream2	1601969	1601969	9328	7815

Table 2. A comparison of the percentage of existing methods and proposed methods

File name	Full Packet	8 packet	5 packet
Filedown1	100	0.097	0.077
Filedown2	100	0.096	0.075
Game1	100	1.505	1.224
Game2	100	2.194	1.825
Stream1	100	0.476	0.399
Stream2	100	0.582	0.488

Table 1 shows the downloading file, playing game and watching stream by each two computers. By all of the files, each packet transmission capacity shows the same capacity by full packet capture method. In cases of Filedown1 and Filedown2. If it saves the 8 packets of initial session, the filedown1 is stored with the 2871KB. The compared to the previous packet, it is able to secure a storage space of about 99.904%. The filedown2 can secure a storage space of about 99.924% by storing 2273KB compared to previous packets.

In cases of Game1 and Game2. If it saves the 8 packets of initial session, the Game1 is stored with the 2871KB. The compared to the previous packet, it was able to secure a storage space of about 98.776%. The Game2 can secure a storage space of about 98.175% by storing 692KB compared to previous packets.

In cases of stream1 and stream2. If it saves the 8 packets of initial session, the stream1 is stored with the 9622KB. The compared to the previous packet, it was able to secure a storage space of about 99.524%. The stream2 can secure a storage space of about 99.418% by storing 9328KB compared to previous packets.

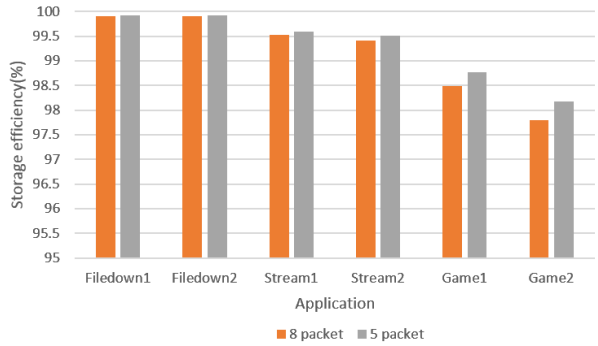

Figure 6. The storage capacity efficiency of proposed method

Figure 6 shows the storage capacity efficiency of proposed method. It considers the existing packet capacity to 100%. The file down saves the storage space with an average of about 99.9% in contrast to the existing packet. The stream is more saving an average of about 99.5%. The game is more saving an average of about 98.2%.

5 Conclusion

We utilize the First-N packets storage method to improve searching speed packet. In the exiting method performed to full packet capture by determining the network information through the pre-treatment, it had the overhead problem for the analysis and the problem of the storage space. In the proposed method, it can confirm the network signature of session by storing the n packets of the initial session (optimal n: 5) and can be capable of high-speed search of the packet according to the proposed storage method. The result according to the proposed method is secured with an additional space of about 99.319%.

We need to confirm the network signature by utilizing the stored data that must be verified.

Acknowledgments. This work was supported by "Human Resources Program in Energy Technology" of the Korea Institute of Energy Technology Evaluation and Planning (KETEP), granted financial resource from the Ministry of Trade, Industry & Energy, Republic of Korea. (No. 20144030200450) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF-2015R1A2A2A03004152.

References

[1] Mahdi Dashtbozorgi, Mohammad Abdollahi Azgomi, A High-Performance Software Solution for Packet Capture and Transmission, 2009 2nd IEEE *International Conference on Computer Science and Information Technology*, (2009), 407-411. http://dx.doi.org/10.1109/iccsit.2009.5234920

[2] Chia-Hao Hsu, Sheng-De Wang, Embedded Network Intrusion Detection Systems with a Multi-core Aware Packet Capture Module, *2011 40th International Conference on Parallel Processing Workshops*, (2011), 207-213. http://dx.doi.org/10.1109/icppw.2011.37

[3] D. Banks, Custom Full Packet Capture System, 2013.

[4] Jun-Sang Park, Sung-Ho Yoon, Myung-Sup Kim, Performance Improvement of Signature-based Traffic Classification System by Optimizing the search, Space, *Korea Society for Internet Information*, **12** (2011), no. 3, 89-99.

Received: April 25, 2016; Published: June 2, 2016