International Journal of Security and Its Applications
Vol. 10, No. 12 (2016), pp.53-68
http://dx.doi.org/10.14257/ijsia.2016.10.12.06

Research on A structure of the Multimedia list Oriented Network Intrusion Detection System

Xu Zhao[1,2], Jin Jiang[3] and Max Stinnett[4]

[1]*Department of Computer Science, Xi'an Polytechnic University, Xi'an, China*
[2]*School of Law, Criminal Justice and Computing, Canterbury Christ Church University, Canterbury, UK*
[3]*Department of Humanities, Xi'an Polytechnic University, Xi'an, China*
[4]*The Teacher College, Emporia State University, Emporia, Kansas, USA*
37274679@qq.com

Abstract

There always appears high packet loss rate in Network Intrusion Detection System (NIDS), especially when the network traffic is high. To address this problem, we have proposed the methods to identifying multimedia packets and processing them in a particular way thus received good results. On this basis, this paper propose a solution that uses a multimedia list structure based on the original list in NIDS. This multimedia list structure can let NIDS reduce the matching times to multimedia packets significantly by shortening the average searching length of OTN nodes dramatically. In addition, this paper also introduce the method of dynamic sorting to the multimedia list in order to shorten the time of rule matching. So this solution can further improve the detection efficiency of NIDS by speeding up the processing efficiency of NIDS to multimedia packets. Various experiments have shown that the packet loss rate of NIDS can be reduced on a large scale and the security of NIDS is not reduced by using the multimedia list.

Keywords: *Network Intrusion Detection System (NIDS); multimedia packets; multimedia list structure; rules*

1. Introduction

1.1. Background

NIDS is used to monitor network traffic and detect attack attempts. As network speeds increase, the requirements of the NIDS's processing efficiency also increase. How to improve the NIDS in the processing capacity per unit time becomes a research hotspot in the field.

In order to solve this problem, many researchers have studied it in different ways. Here are several common research directions.

1) Particle Swarm Optimization [1-2]

LI Zhengjie [2] proposed an intrusion detection model based on immune agent and particle swarm optimization immune principle by means of combining mobile agent and quantum-behaved particle swarm optimization. This system can improve the low detecting speed and high false positive rate of traditional NIDS. However, this approach is easy to fall into local optimum.

2) Clustering algorithm [3-5]

ISSN: 1738-9976 IJSIA

ZHAI Guangqun [5] proposed a new intrusion detection algorithm based on the combination of K-prototypes and fuzzy evaluation. This algorithm has many advantages, but it has some disadvantages. For instance, it might mistake dubious data for normal one.

3) Support Vector Machine [6-7]

LIU Ming-zhen [7] put forward a network intrusion detection model based on the Chaotic Particle Swarm Optimization (CPSO) algorithm and Least Squares Support Vector Machine (LSSVM). This model can select the optimal feature subset and LSSVM parameters. The detecting speed and network intrusion detection accuracy are improved by using this model. However, the detection rate of this model is not ideal for U2R and R2L.

4) Improved pattern matching algorithm [8-12]

Some researchers have improved the efficiency of intrusion detection system by using improved pattern matching algorithm which is an indispensable part in NIDS. For example, Songtan [8] presents an improved multi-pattern matching algorithm which is based on deterministic finite-state automaton. However, this algorithm is only suitable for finding the small character sets pattern string in large character set text string.

5) Rule list [13-16]

As Samaneh Rastegari [13] indicated, because the tree structure of rule list is simple, there exist problems of oversize RTN rule list and overlength average match length in the rule list of NIDS. This fact will result in increased resource consumption and overload at NIDS.

Since the multimedia packets occupy a larger proportion in network flow, the method of particular processing on them can greatly improve the efficiency of NIDS. Among various studies on NIDS, we started from the study of multimedia data in network flow and proposed an identifying method and two separate processing methods [17] for multimedia packets to raise the efficiency of the NIDS. Based on these studies, a multimedia list structure is proposed according to the thought of Breadth-First Search. With its help, NIDS can speeds up the processing efficiency to multimedia packets.

1.2. Contributions of This Paper

To summarize, we mainly make the following contributions in this paper:

a. Among many studies on NIDS, we start from the study of multimedia packets in the network. Based on the previous study, we propose a multimedia list structure which can let NIDS reduce the matching times to multimedia packets significantly by shorten the average searching length of OTN nodes dramatically.

b. We introduce the concrete implementation of the multimedia list structure on NIDS and give the calculation of the average searching length for the pattern match in NIDS.

c. We describes the implementation of dynamic sorting to the multimedia list and give the calculation of the time complexity and space complexity.

d. We demonstrate the performance of our solution on the packet loss rate, the alarm number and other indicators of NIDS by two publicly available intrusion detection datasets, KDD99[18] and CDMC2012[19].

1.3. Paper Organization

The rest of the paper is organized as follows: First, related works are discussed in Section 2; Section 3 presents the design of the multimedia list structure; Section 4

describes the working process in NIDS installed with the multimedia list structure; Section 5 gives the dynamic sorting method of multimedia nodes; the experiment and result analysis for contrast the differences before and after using the multimedia List structure is in Section 6; Section 7 summarizes the whole paper and presents some directions of the future work.

2. Related Works

In the NIDS, the conventional detection method performs the pattern match on all packets under thousands of rules. It does not distinguish between the multimedia packets and non-multimedia packets. Actually, multimedia packets account for a larger proportion in network traffic and are relatively safe [17]. Thus, this makes multimedia packets consume a lot of resources of NIDS.

2.1. The Methods to Processing Multimedia Packets

We has proposed and designed an identifying method and two separate processing methods [17] for multimedia packets to solve this problems.

These two processing methods are as follows:

(1)The releasing method [17]: When the multimedia packet is found in the net flow, this method identify multimedia packets and allows them to bypass the conventional detection method in NIDS. Though this method is simple and efficient, its security is lower.

(2) The corresponding media type detection method (CMTDM) [17]: This method is a safer and more efficient method than the abovementioned method. To initiate this method, a multimedia rule base is created. The multimedia rule base stores the rules which are specifically collected for multimedia packets. The CMTDM can be used to choose the corresponding multimedia rules according to the specific multimedia type that packets carry in order to pre-detect aggressive characteristics. If no aggressive characters are identified in the multimedia packet, the packet is released. However, if aggressive characters are identified by The CMTDM, the packet will be put into the conventional detection method in NIDS. Because the number of multimedia rules contained in the CMTDM is far less than the rules for conventional detection method in the NIDS, this method can significantly improve the detective efficiency for most of the safe multimedia packets. The security of this method is also higher than that of the releasing method.

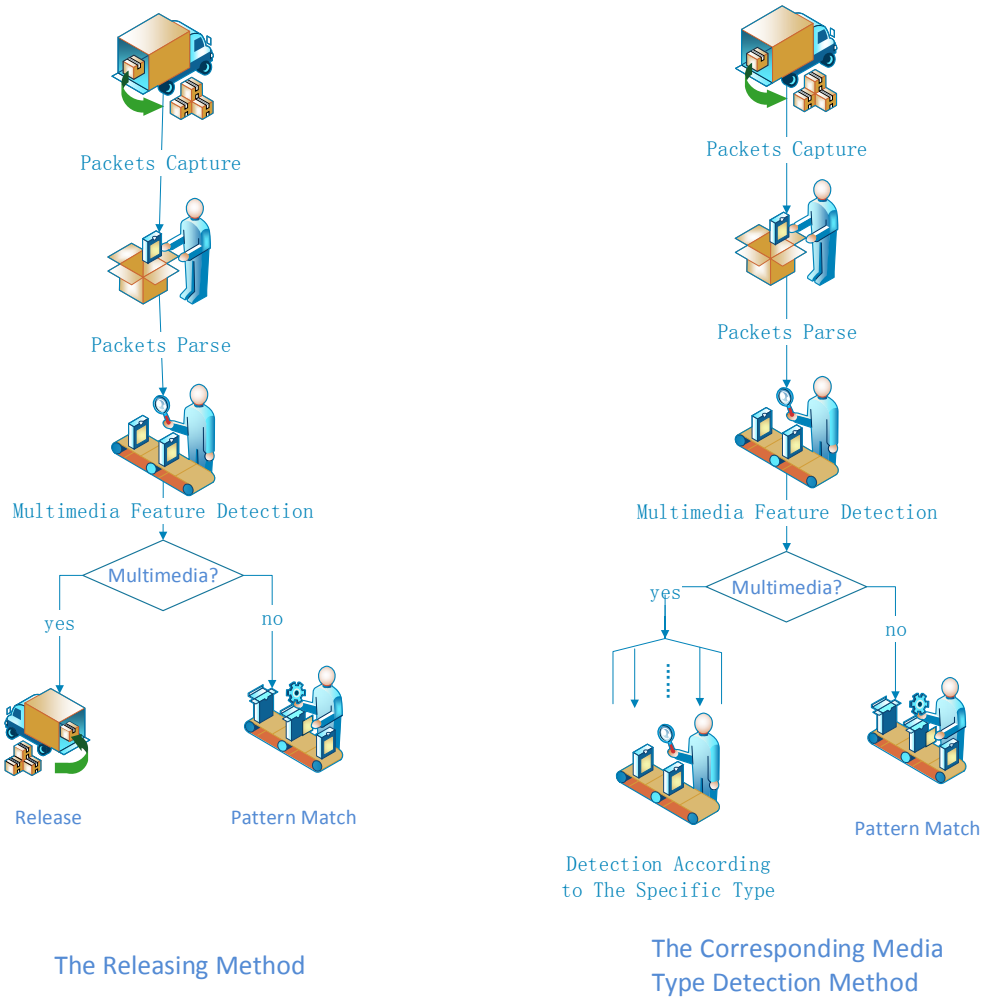

Figure 1. The Work Flow of Two Methods

2.2 The Current Rule List Structure in NIDS

Snort is a well-known open source software of NIDS. Next, Snort is taken as an example to analyze its rule list structure. All of the known intrusion characteristics are extracted and summarized as rules into the rule base in Snort. Snort rules are divided into two logical sections, the rule header and the rule options. The rule header contains the rule's action, protocol, source and destination IP addresses and netmasks, and the source and destination ports information. The rule option section contains alert messages and information on which parts of the packet should be inspected to determine if the rule action should be taken.

The following example illustrates a sample Snort rule:

alert tcp any any -> 192.168.1.0/24 111 \ (content:"|00 01 86 a5|"; msg:"mountd access";)

The text up to the first parenthesis is the rule header and the section enclosed in parenthesis contains the rule options.

The first item in rule header is the rule action. The rule action tells Snort what to do when it finds a packet that matches the rule criteria. There are 5 available default actions in Snort, alert, log, pass, activate, and dynamic. In order to be retrieved a bit faster, these rules are organized to the list. In the list, the rule header is stored in RTN nodes, the rule

options is stored in OTN nodes. According to the classification of the rule action and protocol, the list structure is as follows:

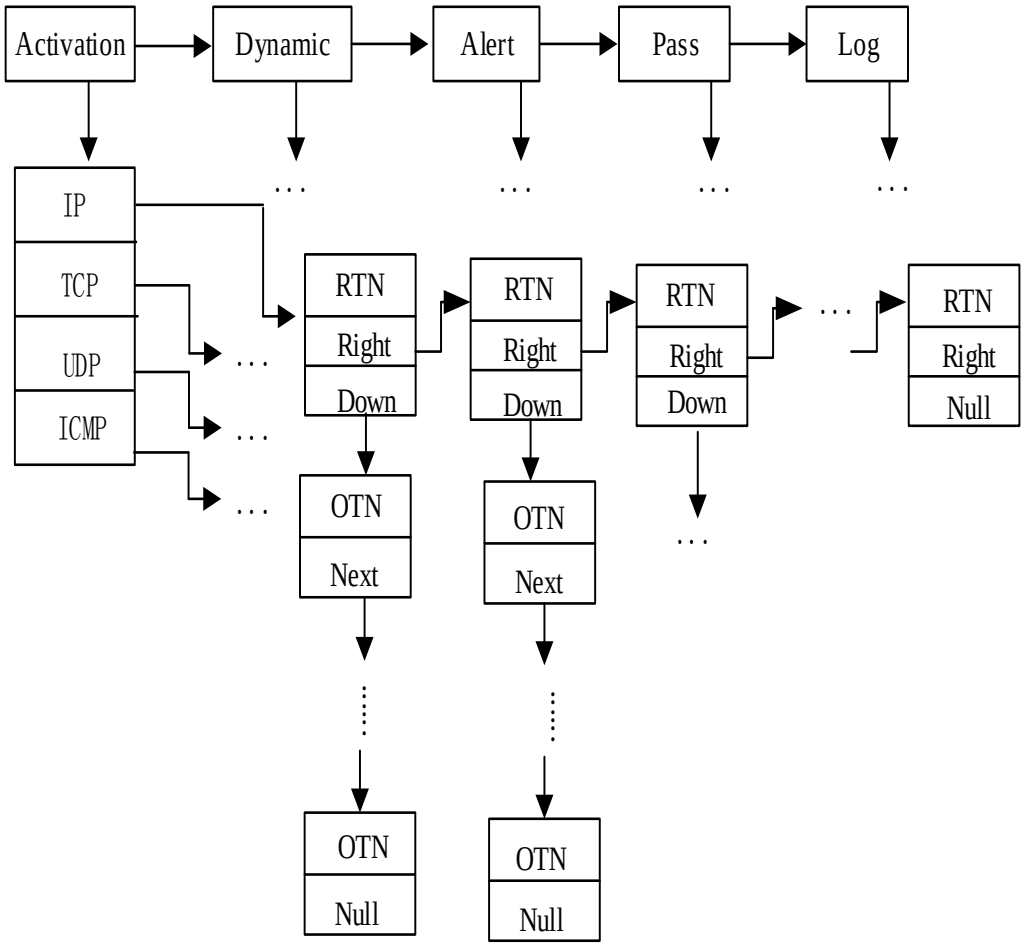

Figure 2. The Rule List Structure of Snort

2.3. Why to Improve

As shown in Figure 2, the tree structure of the rule list of Snort is still relatively simple. This will cause the problems of oversized RTN rule list and overlong average matching length of OTN nodes. These problems will further reduce the efficiency of NIDS because NIDS needs to process too many packets. To address this problem, we propose the multimedia rule list which can dramatically shorten the average matching length of OTN nodes by adding multimedia type nodes and direction nodes in rule list structure of NIDS.

3. The Design of the Multimedia List Structure

Some of the rules in Snort rule base aim at specific media types. For example, the rules in the following describes "if the GIF image data packets from the server to the client contain "! | FF 0B | NETSCAPE2.0" content, is considered "GIF stack overflow."”

alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"WEB-CLIENT Mozilla GIF single packet heap overflow - NETSCAPE2.0"; flow:from_server,established; content:"image/"; pcre:"/^Content-Type\s*\x3a(\s*|\s*\r?\n\s+)image\x2fgif/smi"; content:"GIF"; distance:0; content:"!|FF 0B|NETSCAPE 2.0"; ……)

According to the direction of flow in rule header, the rules of the same kind of multimedia files are divided into four categories (from_server, to_server, from_client and to_ client). OTN nodes of the same categories follow their direction nodes in the rule list structure as Figure 3 shown.

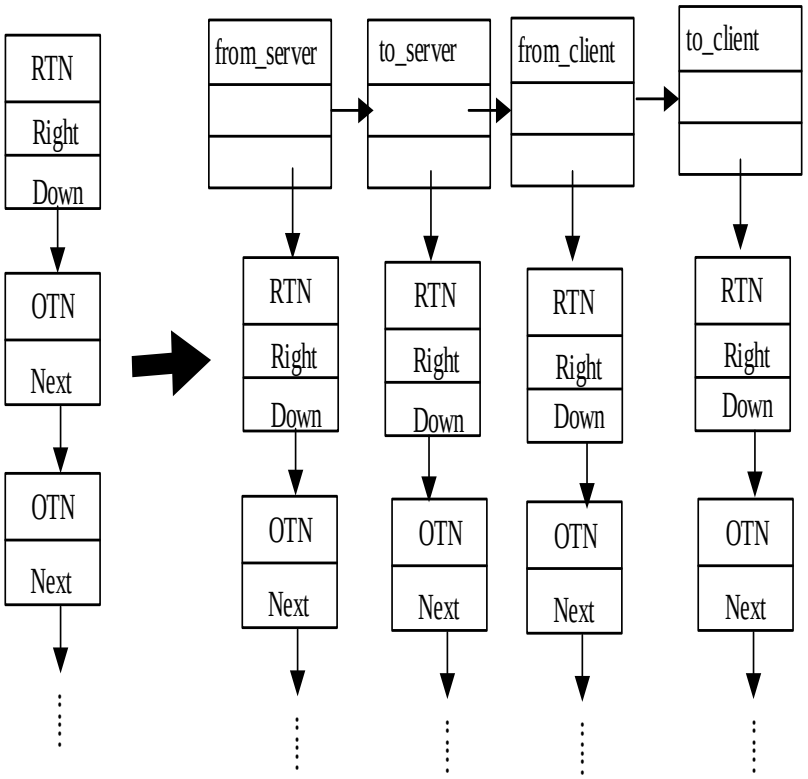

Figure 3. OTN Nodes List is Divided into Four OTN Nodes List

By this way, the average search length (ASL) to each OTN nodes can be dramatically shortened. Let

$$ASL = \sum_{i=1}^{n} P_i C_i \tag{1}$$

P_i is the probability of finding the ith OTN node below the RTN node, and $\sum_{i=1}^{n} P_i = 1$. Ci is the times of comparisons with previous OTN nodes when the ith OTN node is found in list. If the probability of finding every node is equal, that is $P_i = \frac{1}{n}$, then in this case, the ASL to each OTN nodes is following formula:

$$ASL = \sum_{i=1}^{n} P_i C_i = \frac{1}{n} \sum_{i=1}^{n} i = \frac{1+n}{2} \tag{2}$$

When the OTN nodes of the same kind of multimedia file are divided into four categories according to the direction of flow, Let us suppose that the number of OTN nodes is equal, then

$$ASL = \frac{1}{4}\sum_{i=1}^{n} P_i C_i = \frac{1}{4n}\sum_{i=1}^{n} i = \frac{1+n}{8} \tag{3}$$

So ASL is 4 times smaller than it was before.

The next step is to add the multimedia type nodes to the rule list structure. First, the characteristic string extracted from the "content" and "pcre" in each rule for multimedia file, is stored in OTN nodes (as is shown in Figure.4). The left side of the two-dimensional list is the 119 multimedia nodes storing different types of multimedia, the right side of each multimedia node is the horizontal list constituted by four directions nodes. Below each direction node is a vertical list constituted by many OTN nodes in which characteristic strings are stored. By this decomposition method, the length of the list below TCP or UDP protocol node is reduced significantly. As an extreme case, if the following 3 assumptions are established, then the ASL to each OTN nodes can be reduced 476 times as shown in the formula 4.

A. non-multimedia nodes below TCP or UDP protocol node are ignored

B. the number of OTN nodes below each multimedia nodes is equal

C. the number of OTN nodes below each direction nodes is equal

$$ASL = \frac{1}{119} \times \frac{1}{4} \times \sum_{i=1}^{n} P_i C_i = \frac{1+n}{952} \tag{4}$$

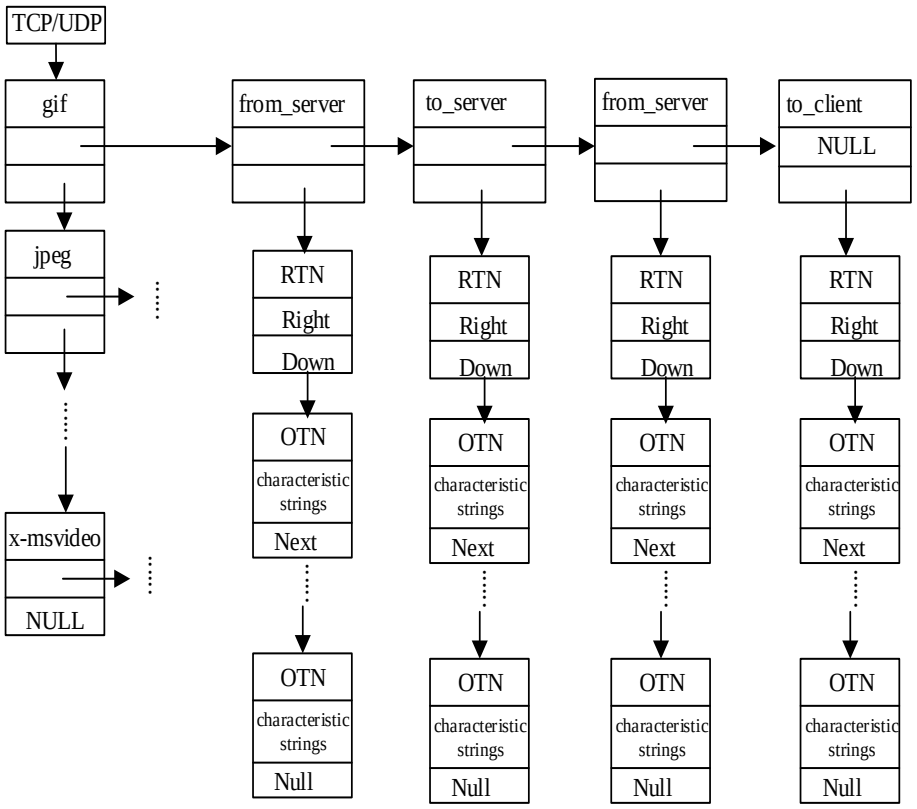

Figure 4. The Two-Dimensional List Structure of Multimedia Rules

The data structure of multimedia nodes is shown as follows:

```
struct _MediaAndCharacter
{
```

```
    int  mediatype;          /* storing the code of multimedia type */
    struct _netflow *next;   /* point to direction nodes*/
    int charnum;                    /* the number of characteristic string */
    ……..
    struct _MediaAndCharacter * down  /* point to the next multimedia node */
};
```

The data structure of direction nodes is shown below:

```
struct _DirectionNode
{
int head_node_number;
int type;
int direction_flag;                 /* the flag for the direction of network flow */
……
struct _ Direction *right;          /* point to the next direction node */
```

CharacterstringNode *down; /* point to the OTN nodes in which characteristic string is stored*/

} DirectionNode;

When the NIDS installed with the multimedia rule list starts, the CreateMediaType function which we add to NIDS is called cyclically to build the multimedia type nodes in two-dimensional list, ProcessFlowNode and ParseCharacterNode functions are also used to construct four direction nodes and the following OTN nodes respectively.

4. The Working Process in NIDS Installed with the Multimedia List Structure

NIDS parse packets after capturing them, then NIDS identify whether they are multimedia packets. If they are, the corresponding multimedia type detection method (CMTDM) can be used to pre-detect intrusion characteristics in multimedia packets with the multimedia rule list. If intrusion characteristics exist, that means the packets may be dangerous, and then the multimedia packets will be handed over to NIDS for conventional detection for safety's sake. If intrusion characteristics don’t exist, just let it go. The whole process will continuously cycle.

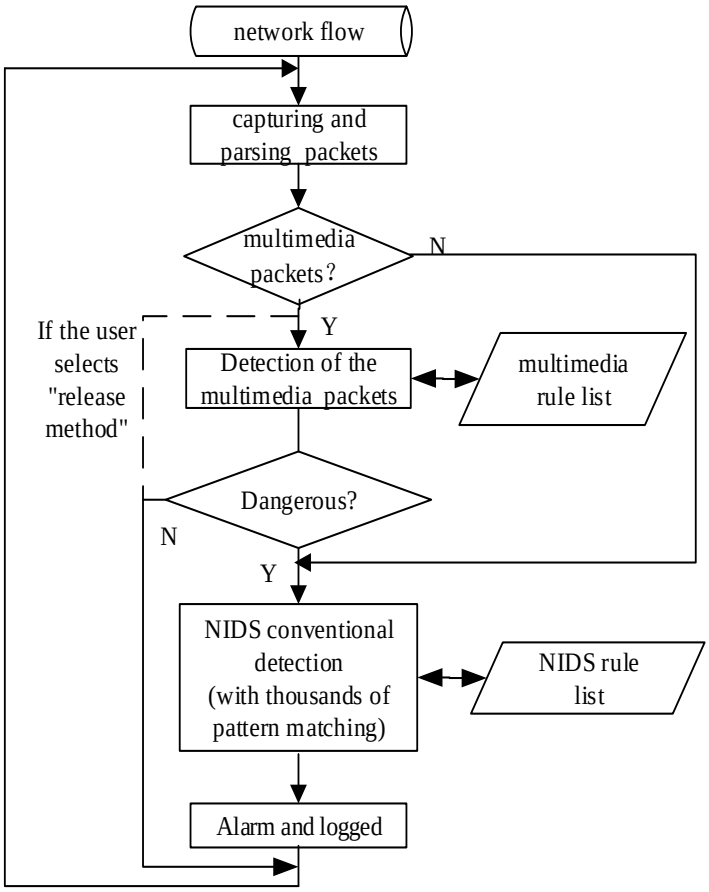

Figure 5. The Working Process of the NIDS Installed with the Multimedia Rule List

In addition, users can also release all the identified multimedia data packets, or only release the multimedia packets from the server (or the client) side by means of setting different parameters on the NIDS Modified. Although this method is the efficient approach of reducing the packet loss rate and it can significantly improve overall detection efficiency of NIDS because these multimedia packets can cross the complicated conventional detection with thousands of pattern matching in NIDS.

5. The Dynamic Sorting in the Multimedia List

5.1. The Dynamic Sorting to Multimedia Nodes and OTN Nodes

The type of multimedia packets in the network traffic will be the same for a period of time through the observation of network traffic. For example, a large number of pictures and texts will appear in the network traffic when the client is browsing the web site. In order to improve the search speed of the NIDS to multimedia packets, we add dynamic sorting function to the multimedia list structure. The principle of dynamic sorting is to allow multimedia types nodes which often appear recently to move forward in the list in order that the second search time for this multimedia type node can be reduced. The specific implementation of the method is that if a certain type of multimedia file is detected, the node of this multimedia type and the above node swap position in the multimedia list. Since nodes are in the list, changing their order only needs to change their head and tail pointer as shown in Figure 6.

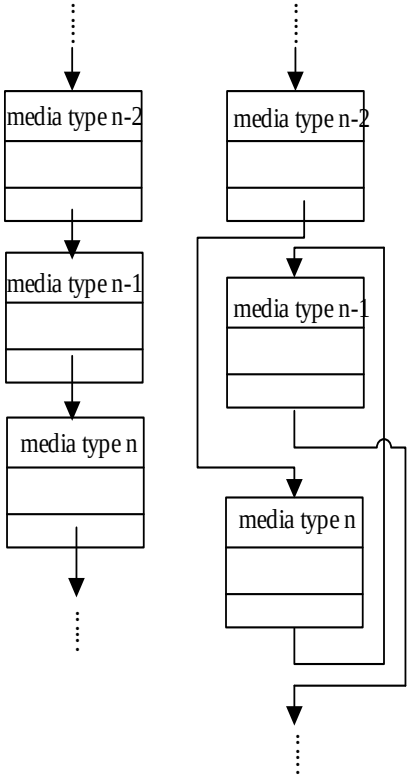

Figure 6. Changing Multimedia Type Nodes' Order by Changing their Head and Tail Pointer

For example, when the page is being downloaded, some gif files are continuously detected, so the position of the gif node moves forward so that the subsequent gif files can be detected quickly (as shown in Figure 7).

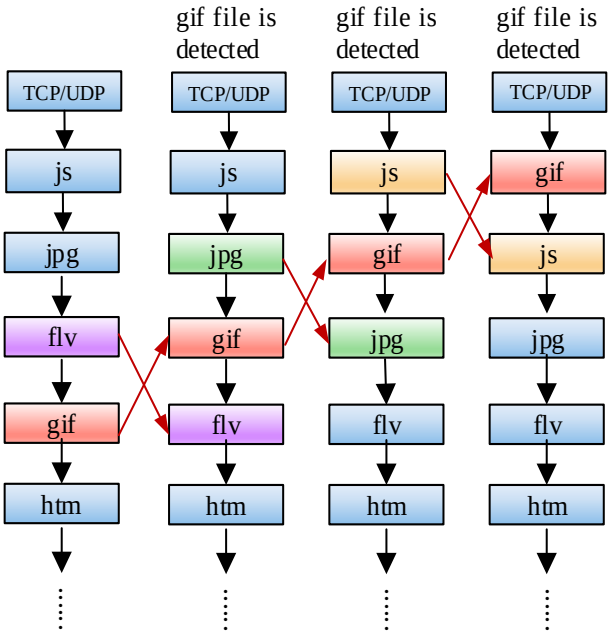

Figure 7. Dynamic Sorting with GIF as an Example

Since multimedia packets with some kind of attack information will appear in a period of time frequently, dynamic sorting to OTN nodes will also reduce the match time to them. The specific method of dynamic sorting is the same as above. If a certain characteristic strings is detected, this node and the above node will swap position by changing their head and tail pointer as shown in Figure 8 in the multimedia list.

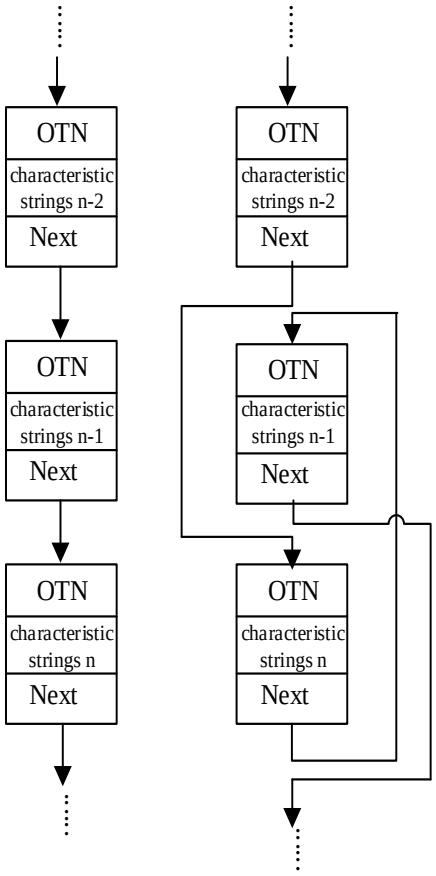

Figure 8. The Dynamic Sorting of Characteristic String Nodes

5.2. The Time Complexity and Space Complexity of Dynamic Sorting

If NIDS continuously captures some kind of multimedia file and this kind of multimedia node is just right on the top of the multimedia list, so both the search times (ST) and move times (MT) achieve the minimum value:

$$ST_{\min} = 1, MT_{\min} = 0 \tag{5}$$

So the best time complexity of dynamic sorting is $O(1)$.

If this kind of multimedia node is just right on the bottom of the multimedia list, in order to move this node to the top, n-1 times search is needed to find it in the first round (n is the number of multimedia nodes). In the second round, n-2 times is needed, and so on. In each round, pointers need to be moved three times in order to swap positions .In this case, the search times and move times achieve the maximum value:

$$ST_{\max} = \frac{n(\mathrm{n}-1)}{2} = \mathrm{O}(\mathrm{n}^2), MT_{\max} = 3n = \mathrm{O}(\mathrm{n}) \tag{6}$$

So the worst time complexity of dynamic sorting is $O(n^2)$.

Since the space complexity does not change with the size of n obviously, so it can be expressed as $O(1)$.

6. Experiments and Result Analysis

6.1. Experimental Environment

The experiments reported here demonstrate a variety of changes in performance before and after using the multimedia rules list on NIDS. Experimental environment consists of three computers (OS: WIN 10, CPU: Intel Core i7 5960X, Memory: 8GB DDR4).

In the experiment, the network traffic captured before is sent by the first computer, which is used as real background traffic, contains a large number of multimedia packets. As the attacker, the second computer uses Lincoln Laboratory KDD CUP 99 data set and IDS Informer to generate attack traffic. Both mixed flows are sent to test NIDS installed on the third computer, as shown in Figure 9.

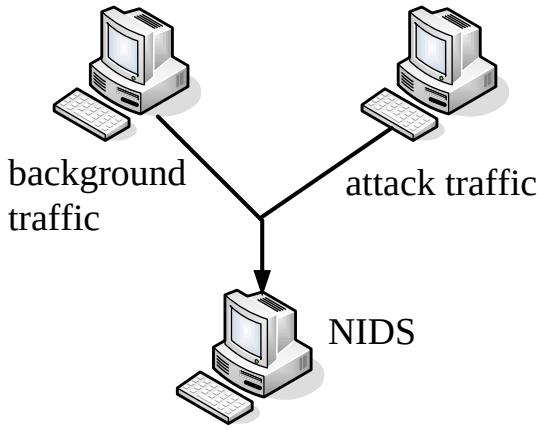

Figure 9. Experimental Environment

In the attack traffic, including the four types of network attacks [18-19]: DoS,R2L,U2R and PROBE, the types and quantities are shown as follows:

Table 1. The Types and Quantities in the Attack Traffic

	DoS	R2L	U2R	PROBE	Total
number	229853	16137	228	4166	250384

6.2. The Packet Loss Rate

The first step in the experiment is to compare the packet loss rate before and after using the multimedia rule list, the results are shown in Figure 10. The packet loss detection threshold increases significantly when the multimedia rule list is added in the NIDS, it has been improved from 40Mbps to 84Mbps after improvement. The packet loss rate decreases by 19%-25% when the bandwidth is 20 ~ 123Mbps and decreases by 16% to 31% when the bandwidth is 215 ~ 305Mbps.

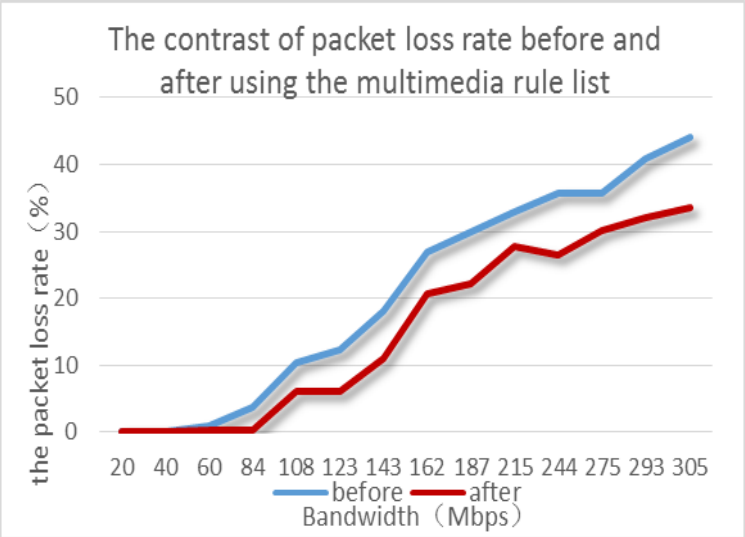

Figure 10. The Contrast of Packet Loss Rate before and after using the Multimedia Rule List

6.3. The Number of the alarm

The second step is to test security before and after using the multimedia rule list. Security can be determined by the number of the alarm record in the log file, the results are shown in Figure 11.

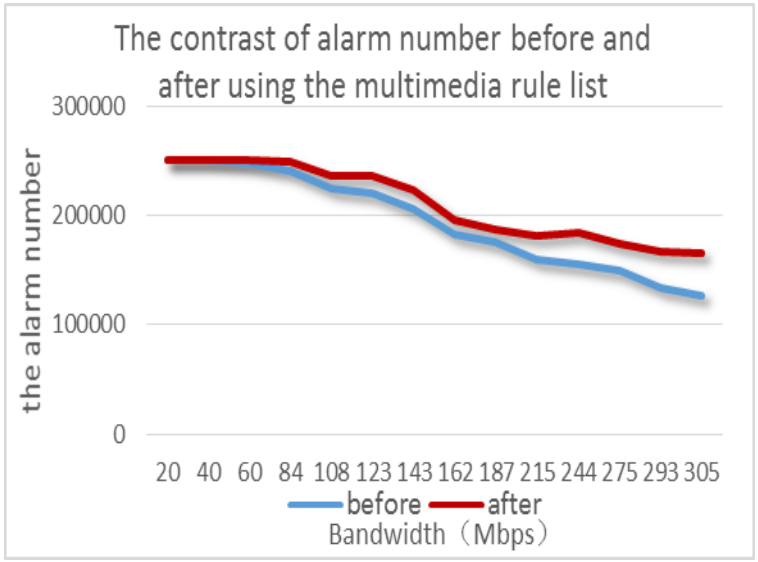

Figure 11. The Contrast of Alarm Number before and after using the Multimedia Rule List

The figure above indicates that when using the multimedia rule list, the alarm number increases with the packet loss rate's growth. The greater the difference in packet loss rate is, the greater the difference between the numbers of alarms is. The difference between the numbers of alarm even can reach up to 23% when the bandwidth is 305Mbps.

Analysis of those numbers has revealed that because the packet loss rate decreases by using the multimedia rule list, the number of packets detected increases, the alarm number also increases with it.

7. Conclusion and Future Work

Among many studies on NIDS, this article studies from multimedia file in the network, designs and implements multimedia rule list. This method can identify various types of multimedia packets in network traffic and detect it in advance, so that secure multimedia packets can go across the conventional detection process of NIDS to save detection time. Experiments show that the packet loss rate decreases obviously and the security of the system is improved when the multimedia rule list is used in NIDS.

As for future work, firstly, we will continue to improve the multimedia list, for example by using a more rapid dynamic scheduling algorithm. Secondly, I will compare continually the advantage with disadvantage of the multimedia list proposed in the paper with those of other experiments to get more objective evaluations in the higher speed network environment.

Acknowledgment

This research was supported by Scientific research program of the Education Department of Shaanxi Provincial Shaanxi Province (16JK1347), Xi'an Beilin District Science and Technology Bureau (GX1509),Social Science Foundation of Shaanxi Province(2016R030)

References

[1] W. Qingtao, C. Jibang and Z. Ruijuan, "Intrusion feature selection algorithm based on Particle Swarm Optimization", Computer Engineering and Applications, vol. 49, no. 7, **(2013)**, pp. 89-92.
[2] L. Zhengjie, L. Yongzhong and X. Lei, "Research of intrusion detection method based on particle swarm optimization and immune Agent", Computer Engineering and Applications, vol. 48, no. 1, **(2012)**, pp. 102-104.
[3] J. Shen and W. Dawei, "An Improved Ant Colony Clustering Method for Intrusion Detection", Computer Technology and Development, vol. 23, no. 12, **(2015)**, pp. 139-142.
[4] L. Feng-zhu, "A Clustering Method for Anomaly Intrusion Detection", Computer Security, vol. 15, no. 2, **(2013)**, pp. 156-161.
[5] Z. Guangqun, "Intrusion detection algorithm based on fuzzy evaluation and clustering analysis", Computer Engineering and Applications, vol. 48, no. 21, **(2012)**, pp. 99-102.
[6] Z. Guangping and A. Shrestha, "Efficient Intrusion Detection Scheme based on SVM", Journal of networks, vol. 8, no. 9, **(2013)**, pp. 2128-2134.
[7] L. Ming-zhen, "Network Intrusion Detection Based on CPSO-LSSVM", Computer Engineering, vol. 39, no. 11, **(2013)**, pp. 131-135.
[8] S. Tian and L. Dong-Ni, "Memory Efficient Algorithm and Architecture for Multi-Pattern Matching", Journal of Software, vol. 24, no. 7, **(2013)**, pp. 1650-1665.
[9] L. Linlin and T. Ye, "Research on Proving Multi-pattern Matching Algorithm Based on Deterministic Finite-state Automation", Computer Applications and Software, vol. 30, no. 7, **(2013)**, pp. 321-323.
[10] L. Wei-guo, H. Yong-gang and DHSWM, "An improved multi-pattern matching algorithm based on WM algorithm", Journal of Central South University, vol. 4212, **(2011)**, pp. 3765-3771.
[11] Y. Hongwen, "Research on improved BMH single-pattern matching algorithm based on Snort", Computer Engineering and Applications, vol. 48, no. 31, **(2012)**, pp. 78-81.
[12] D. Shibo, L. Xungen and Y. Zhenzhen, "Improved string matching algorithm", Computer Engineering and Applications, vol. 49, no. 8, **(2013)**, pp. 133-137.
[13] S. Rastegari, Evolving statistical rulesets for network intrusion detection", Applied Soft Computing vol. 33, **(2015)**, pp. 348-359.
[14] Y. Shu-ting, "Analysis and improvement of structure of snort rule chain", Journal of Yanshan University, vol. 30, no. 3, **(2009)**, pp. 1380-1387.
[15] K. Sravani and P. Srinivasu, "Comparative Study of Machine Learning Algorithm for Intrusion Detection System", Advances in Intelligent Systems and Computing, vol. 247, **(2014)**, pp. 189-196.
[16] M. Arun and A. Krishnan, "Functional verification of signature detection architectures for high speed network applications", International Journal of Automation and Computing, vol. 9, no. 4, **(2012)**, pp. 301.
[17] Z. Xu and W. Changshan, "The Improvements to Snort Intrusion Detection System", Journal of Xi"an Polytechnic University, vol. 21, no. 6, **(2007)**, pp. 859-863.
[18] M. Mahoney and P. K. Chan, "An analysis of the 1999 DARPA/Lincoln laboratory evaluation data for network anomaly detection", RAID, **(2003)**, pp. 220-237.

[19] A. Hesham and K. Shahbar, "(WHASG) Automatic SNORT Signatures Generation by using Honeypot", Journal of Computers, vol. 8, no. 12, **(2013)**, pp. 3280-3286.
[20] W. Ren, L. Hu and K. Zhao, "Intrusion Classifier based on Multiple Attribute Selection Algorithms", Journal of Computers, vol. 8, no. 10, **(2013)**, pp. 2536-2543.
[21] Z. Shi, Y. Xia, F. Wu and J. Dai, "The Discretization Algorithm for Rough Data and Its Application to Intrusion Detection", Journal of networks, vol. 9, no. 6, **(2014)**, pp. 1380-1387.
[22] Z. Xu, "Research on Dynamic Self-Adapting Multimedia Data Processing Method Based on Snort", Computer Systems & Applications, vol. 20, no. 4, **(2011)**,pp. 211-213.

Authors

Xu Zhao is an associate professor in the Department of Computer Science, Xi'an Polytechnic University, Shannxi, China. The visiting scholar of School of Law, Criminal Justice and Computing, Canterbury Christ Church University. He received the M.S. degree from Xi'an Electronic Technology University, Xi'an City, Shannxi Province, China in 2007. He has developed several methods to deal with multimedia packets for network intrusion detection systems and is currently working on new optimization method with the help of artificial intelligence. He has some projects in research supported by provincial funds. His research interest is Network Security.

Jin Jiang is a Lecturer in the Xi'an Polytechnic University, Shannxi, China. She received the M.E. degree from Xi'an Technological University, Xi'an City, Shannxi Province, China in 2010. She has some projects in research supported by provincial funds. Her research interest is Network Security.

Max Stinnett is an instructor in the Emporia State University, Emporia, Kansas, USA. His research interest is multimedia communication.