An Improved NPCUSUM Method with Adaptive Sliding Window to Detect DDoS Attacks

Degang Sun, Kun Yang, Weiqing Huang, Yan Wang(✉), and Bo Hu

Institute of Information Engineering,
Chinese Academy of Sciences (CAS), Beijing, China
{sundegang,yangkun,huangweiqing,wangyan,hubo}@iie.ac.cn

Abstract. DDoS attacks are very difficult to detect, researches have been in the pursuit of highly efficient and flexible DDoS attacks detection methods. For this purpose, we put forward an improved Non-parametric CUSUM method (NPCUSUM), which combined with adaptive sliding windows (ASW), to detect DDoS attacks. In order to evaluate our method, we do experiments on 2000 DARPA Intrusion Detection Scenario Specific Data Set (DARPA 2000 Dataset). The results show that the proposed method improves the detection efficiency and has good flexibility.

Keywords: NPCUSUM · Sliding window · Conditional entropy · DDoS attacks detection · Darpa 2000 dataset

1 Introduction

With the rapid development of Internet, it is an exponential growth on network attacks. Due to the availability of many tools for novices, and the difficulty of tracking attackers, DDoS attacks have become a major threat to Internet [1]. Attackers employ Botnets to launch DDoS attacks for specific targets (Victims), which can lead victims to be paralyzed and cause huge economic losses [2,3].

Currently, there are plenty of detection methods [4–6] against DDoS attacks. Their main idea is almost the same that firstly extracted features which can be used to represent DDoS attacks, then found the abnormalities in network traffic.

In the article [7], the authors proposed to build a database of normal sequences by sliding a window of length n on hosts. However, the size of window is fixed, which does limit the scalability of the algorithm. The authors [8] employed statistical approaches, entropy and Chi-Square Statistic, to detect DDoS attacks. In the paper, the threshold won't be changed, so it is not a smart and wise method. In the article [9], the authors put forward a fast entropy method combined with several sliding windows. But it needs to set the initial thresholds, and it uses the pcap data which means to need much more time on calculation. The authors in the paper [10] proposed a multi-attributes CUSUM

S. Qing et al. (Eds.): ICICS 2015, LNCS 9543, pp. 303–310, 2016.
DOI: 10.1007/978-3-319-29814-6_25

method based on conditional entropy [16,17]. Nevertheless, the thresholds cannot be updated automatically and the size of sliding windows are not variable.

Based on the above reasons, this paper proposes an improved Non-parametric CUSUM method (NPCUSUM) method based on adaptive sliding windows to detect DDoS attacks. The main improvements of the detection method are as follow, (1) calculating conditional entropy based on netflow flow data rather than pcap packets data [13–15], (2) employing an improved NPCUSUM method based on an adaptive sliding window (ASW) to detect DDoS attacks, (3) employing tolerance factor to reduce false positive rate.

In this paper, the adaptive sliding window we designed, which could adjust its size automatically and auto update thresholds without human intervention, according to the network traffic. Our method do not need to set the initial thresholds, it can be auto updated. In order to reduce false positive rate, a tolerance factor does be employed. Only the number of attacks detected beyond the tolerance factor, then produce one alert.

The rest of this paper is organized as follows. In Sect. 2, we present the improved detection algorithm and elaborate each part in detail. In Sect. 3, we show our experiments and analysis the result. Finally, we conclude the paper in Sect. 4 and give the next step of our research.

2 Proposed Detection Algorithm

In this section, we elaborate the proposed detection approach of DDoS attacks. Table 1 lists mainly algorithm's steps, Fig. 1 displays the sketch map of our algorithm, and Table 2 gives some notions.

Table 1. Proposed algroithm' steps

(1) Transform pcap packets data to netflow flows data,
(2) Select features for detection,
(3) Chose interval time and compute the features values,
(4) Compute cumulative sum by the improved NPCUSUM method,
(5) Compare cumulative sum to thresholds and detect attacks,
(6) Updated thresholds and SASW's size.

2.1 Transform Pcap Packets Data to Netflow Flows Data

In order to reduce computation time and achieve good performance in real time, we transform pcap packets data to netflow flows data [18,19]. Generally, Fig. 2 shows a typical output of a NetFlow command line tool-nfdump.

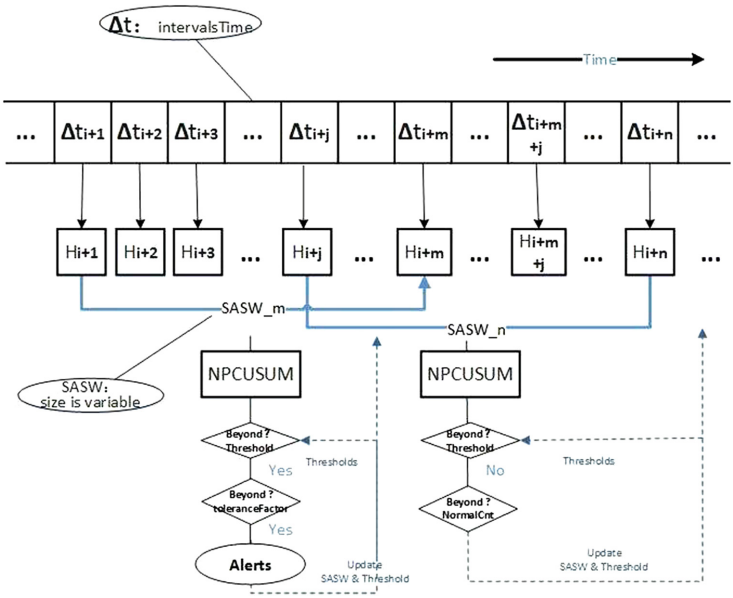

Fig. 1. Proposed algorithm's flowchart

Table 2. Notions

Δt: intervals time. The minimums calculating unit for conditional entropy
Hi: conditional entropy
$SASW$: simple adaptive sliding window. Its size is variable : $SASW\_m$ and $SASW\_n$ represent different SASW's sizes
$NPCUSUM$: Non-parametric CUSUM algorithm
$toleranceFactor$: tolerance factor
$normalCnt$: continuous SASW number of normal level
$Threshold$: estimate whether the flow is a DDoS attack or not

```
Date flow start          Duration Proto   Src IP Addr:Port      Dst IP Addr:Port     Packets    Bytes Flows
2010-09-01 00:00:00.459    0.000 UDP     127.0.0.1:24920   ->  192.168.0.1:22126         1       46     1
2010-09-01 00:00:00.363    0.000 UDP     192.168.0.1:22126 ->  127.0.0.1:24920           1       80     1
```

Fig. 2. Netflow's flow data

2.2 Select Features for Detection

In this paper, considering the efficiency of the algorithm, we only chose two features, conditional Entropy ($H(srcIP|dstIP)$) and the number of flows per interval time ($flowCnt$) [11,12]. We do not choose any ports (source/destination ports) information, because ports usually are uncertain, and any other information, such as, the number of packets per interval Time (packets), bits per second (bps), packets per

second (pps) and Bytes per package (Bpp) information, because they and $flowCnt$ have the same properties.

2.3 Choose Interval Time and Compute the Features Value

In this paper, interval time is the minimize unit in our detection algorithm, and we set 1 s as the default interval time.
Conditional Entropy ($H(srcIP|dstIP)$) is computed,

$$\begin{aligned} H(srcIP|dstIP) &= \sum_{dstIP \in \Delta t} p(dstIP) H(srcIP|dstIP) \\ &= -\sum_{dstIP \in \Delta t} p(dstIP) \sum_{srcIP \in \Delta t} p(srcIP|dstIP) log H(srcIP|dstIP) \end{aligned} \tag{1}$$

FlowCnt is computed,

$$flowCnt = log2(sum(flows)/intervalTime) \tag{2}$$

2.4 Compute Cumulative Sum by the Improved NPCUSUM Method

In this paper, we calculate cumulative values between the current SASW and the last normal SASW. ByondnormalLevelSum represents the cumulative sum which beyond averages (normalMeanPre) in the last normal SASW, LessnormalLevelPre represents the cumulative sum which less than averages in the last normal SASW.

$$\begin{aligned} ByondNormalLevelSum &= sum(\{diffValues | diffValues = currentValues \\ &\quad - normalMeanPre \& diffValues > 0\}) \\ LessNormalLevelSum &= sum(\{diffValues | diffValues = currentValues \\ &\quad - normalMeanPre \& diffValues < 0\}) \end{aligned} \tag{3}$$

2.5 Compare Cumulative Sum to Thresholds and Detect Attacks

In traditional methods, they just compare the cumulative sum to thresholds [16,17], which will cause too much alerts, so we introduce tolerance factor-toleranceFactor. Our improved method is to cumulate the number of attacks-AttackCnt. If AttackCnt > toleranceFactor, then produce alerts. In this way it will reduce many alerts. Pseudocode 1 shows our improved idea.

2.6 Updated Thresholds and SASW's Size

In this section, we will update the thresholds and SASW's size. If there is a DDoS attack, we employ the average of three previous thresholds to update the thresholds, and SASW's size substract 1, otherwise, use the previous values. This avoids just use one single threshold to update the new thresholds.

Pseudocode 1. Estimating Improved NPCUSUM

```
if (ByondnormalLevelSum > thresholds||LessnormalLevelSum < thresholds) then
    warning : It is maybe some abnormality, please look out!
    AttackCnt = AttackCnt + 1
    if AttackCnt > toleranceFactor then
        alerts : Attack is Happened!
    end if
else
    normal values
end if
```

3 Experiment and Analysis

The 2000 DARPA dataset is a typical dataset of DDoS attack traffic, which includes a DDoS attack run by a novice attacker (MIT Lincoln Lab, 2000) [20]. In this section, we do experiments on Darpa 2000 dataset, and analysis the results. The total test time is about 3 h and the initial parameters are as follow, initial threshold is [0,0], intervals time is 1 s, toleranceFactor=4, SASW: [SASWSizeMin,SASWSizeMax]=[3,10].

3.1 Results

In this part, we show the results. Employing formulas (1) and (2), we can compute the conditional entropy ($H(srcIP|DstIP)$) and entropy (flowCnt) at each interval time, which can be seen in Fig. 3, the maximum entropy values represent that there is a DDoS attack.

Figure 4 shows that the thresholds are changed continuously in order to adapt the status of network traffic. At the beginning, the initial threshold is [0,0]. However, when the status of network traffic has changed, the threshold also changed automatically. When DDoS attack has happened at peak points in Fig. 4, thresholds employed the previous values and can still detect the attack.

Figure 5 is cumulative sum at each SASW, which shows that a huge change appeared when DDoS attack happened. And we can compare this value to thresholds. If this value beyond thresholds at toleranceFactor times, then produce alerts.

Figure 6 gives the change of SASW. When there are not DDoS attacks, the SASW will increase, in order to increase cumulative sum, otherwise, the SASW will decrease, because of DDoS attacks happened. At this time, a DDoS attack happened, so SASW has decreased.

Compared with [13], the authors employ many conditional entropy to detect DDoS attacks which spends several hours. But in our paper, we just use $H(srcIP|dstIP)$ and flowCnt to detect, our method spends less time, which is 1993.3560 s – about half hour, to detect DDoS attacks on Darpa 2000 and can automatic adjust parameters to adapt the status of network traffic.

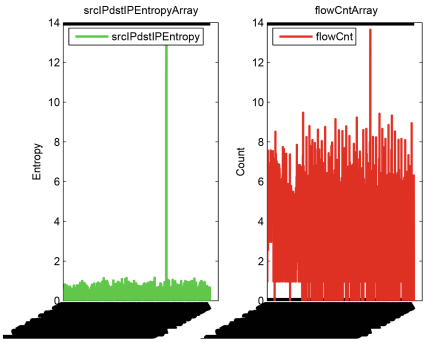

Fig. 3. Entropy

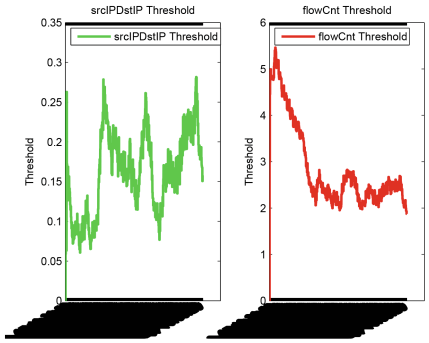

Fig. 4. Threshold

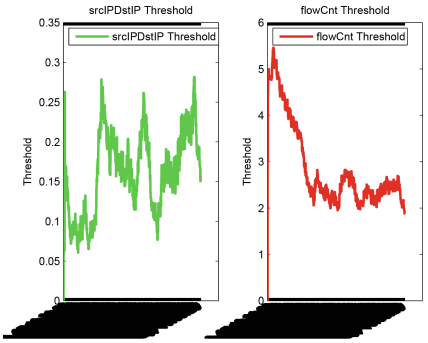

Fig. 5. NPCusum

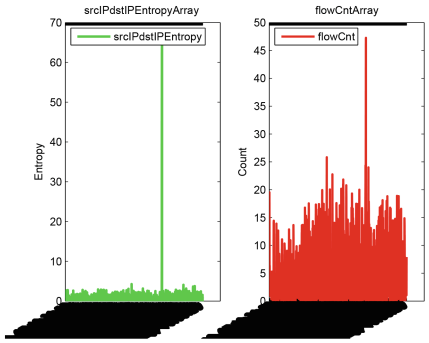

Fig. 6. SASWSize

Through the experiments, our method can adjust the algorithm's parameters, such as, automatic to learn detection thresholds, dynamic to update detection thresholds and auto-adjust sliding windows size, according to the status of network. Our method improves the detection efficiency and has good flexibility. It can be used in practice for real-time detection.

Notes: Due to lots of values, x-axis can not be seen clearly, so we give the explanation that x-axis represents the start time of each interval or SASW.

4 Conclusion

In this paper, we propose an improved Non-parameter CUSUM method (NPCUSUM) based on an adaptive sliding window to detect DDoS attacks. According to the status of network, it is able to automatically adjust the parameters of NPCUSUM, such as, automatic to learn detection threshold, dynamic to update detection thresholds and automatic to adjust the sliding window size.

In order to evaluate our method, we do experiments on DARPA 2000 Dataset, the results show that its flexible and effective to detect DDoS attacks.

But this method also has several deficiencies, (1) it can't distinguish between Flash Contest and DDoS attacks traffic, (2) it will causes a certain delay. In this paper, we introduce toleranceFactor to reduce the number of alerts. ToleranceFactor controls the detection sensitivity. It will cause a certain delay. In practice, it is always a tough task for researchers.

In view of the above reasons, we need to constantly improve our method and make it more reasonable and efficient.

References

1. Neustar.biz (2014). http://www.neustar.biz/resources/whitepapers/ddos-protection/2014-annual-ddos-attacks-and-impact-report.pdf
2. Kaspersky Report, Statistics on botnet-assisted DDoS attacks in Q1 2015
3. Cloudflare.com (2013). http://blog.cloudflare.com/the-ddos-that-knocked-spamhaus/-offline-and-ho/
4. Bhuyan, M.H., et al.: Detecting distributed denial of service attacks: methods, tools and future directions. Comput. J. **57**(4), 537–556 (2014)
5. Prasad, K.M., Reddy, A.R.M., Rao, K.V.: DoS and DDoS attacks: defense, detection and traceback mechanisms-a survey. Global. J. Comput. Sci. Technol. 14(7) (2014)
6. Murtaza, S.S., Khreich, W., Hamou-Lhadj, A., et al.: A host-based anomaly detectionapproach by representing system calls as states of kernel modules. In: 2013 IEEE 24th International Symposium on Software Reliability Engineering (ISSRE), pp. 431–440. IEEE (2013)
7. Forrest, S., Hofmeyr, S., Somayaji, A., et al.: A sense of self for unix processes. In: 1996 IEEE Symposium on Security and Privacy, pp. 120–128. IEEE (1996)
8. Feinstein, L., Schnackenberg, D., Balupari, R., Kindred, D.: Statistical approaches to DDoS attack detection and response. In: Proceedings of DARPA Information Survivability Conference and Exposition, vol. 1, pp. 303–314. IEEE, April 2003
9. No, G., Ra, I.: Adaptive DDoS detector design using fast entropy computation method. In: 2011 Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), pp. 86–93. IEEE (2011)
10. Zhao, X.H., Xia, J.B., Guo, W.W., Du, H.H.: Detection DDoS attacks based on multi-dimensional entropy. J. Air Force Eng. Univ. (Natural Science Edition) **3**, 015 (2013)
11. Conditional Entropy. https://en.wikipedia.org/wiki/Conditional_entropy
12. Cover, T.M., Thomas, J.A.: Elements of Information Theory, 1st edn. Wiley, New York (1991). ISBN 0-471-06259-6
13. Bereziski, P., et al.: An entropy-based network anomaly detection method. Entropy **17**(4), 2367–2408 (2015)
14. Thapngam, T., Yu, S., Zhou, W., Makki, S.K.: Distributed Denial of Service (DDoS) detection by traffic pattern analysis. Peer-to-Peer Networking Appl. **7**(4), 346–358 (2014)
15. Bhuyan, M.H., Bhattacharyya, D.K., Kalita, J.K.: An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection. Pattern Recogn. Lett. **51**, 1–7 (2015)

16. Page, E.S.: Continuous Inspection Scheme. Biometrika 41 (1/2): 100C115(1954). doi:10.1093/biomet/41.1-2.100.JSTOR2333009
17. Bassevilleand, M., Nikiforov, I.V.: Detection of Abrupt Changes: Theory and Application. Prentice-Hall Inc., Upper Saddle River (1993)
18. Cisco. http://www.cisco.com/c/en/us/tech/quality-of-service-qos/netflow/index.html
19. Hofstede, R., Celeda, P.: Flow monitoring explained: from packet capture to data analysis with NetFlow and IPFIX. IEEE Commun. Surv. Tutorials (IEEE Communications Society) **16**(4), 28 (2014). doi:10.1109/COMST.2014.2321898
20. Darpa2000. http://www.ll.mit.edu/IST/id/data/2000/LLS_DDOS_1.0.html