A New Network Flow Grouping Method for Preventing Periodic Shrew DDoS Attacks in Cloud Computing

ZengGuang Liu*, XiaoChun Yin**, Hoon Jae LEE***

* IP Platform Dept of MGC, Alcatel-Lucent, 169, SongLing Road, QingDao, China
** WeiFang University of Science & Technology, 1299, JinGuang Street, ShouGuang, China
*** Division of Computer and Engineering of Dongseo University, Sasang-Gu, Busan 617-716, Korea
sterling.liu@alcatel-lucent.com, Spring.yin@163.com, hjlee@dongseo.ac.kr

Abstract— Based on the investigation of periodic shrew distributed DoS Attacks among enormous normal end-users' flow in cloud computing, this paper proposed a new method to take frequency-domain characteristics from the autocorrelation sequence of network flow as clustering feature to group end-user flow data by BIRTH algorithm, and re-merge these clustering results into new groups by overcoming the deficiency of BIRTH algorithm. At last, the result of simulation proves the proposed method distinguishes abnormal network flows with higher detection accuracy and faster response time, and prevents abnormal network flow groups with less impaction.

Keywords— Cloud Computing, Periodic Shrew Distributed DoS, Network Flow Grouping, Clustering Feature, Detection Accuracy, Response Time

I. INTRODUCTION

Traditional DDoS attacks, characterized by brute-force, sustained high rate or specifically designed to explore the protocol limitations or software vulnerabilities in services, are well-known and can be detected with many methods. But distributed low-rate DoS attacks, as a new category, are becoming a serious threat to Internet, especially to cloud computing with enormous normal end-users. Compared with traditional DDoS attacks, they have three mainly characteristics: hard to detect because it has same flow classified features with normal flow; low-cost because the attacks can be finished in single node with small flow data; long term attacked-target insensitive attacks because attacked-target has self-adapt mechanism (treat them as normal flow) to adjust network flow. Thus, DDoS attacks can not only finish all kinds of attacks, but also hard to be detected. Recently, Yu Chen, Barbhuiya FA etc gives their owned detecting or preventing methods to address above issues, but they are hard to implement in real cloud environment. Thus, the aim of this paper is to have a method with high detection accuracy, fast response time and light-weight implementation to detect and prevent periodic TCP targeted Shrew DDoS Attacks.

II. TCP TARGETED SHREW DDoS ATTACKS IN CLOUD COMPUTING

In cloud computing, the TCP targeted shrew DDoS attacks launched by multiple zombies could lower their individual traffic rates further, compared with single shrew attack stream. Since the distributed attack sources could decrease its average traffic either by lowering the peak rate (known as Synchronous DDLSoS Attacks, in Fig 1-1) or by using longer attack periods (known as Asynchronous DLDSoS Attacks, in Fig 1-2). Thereby it makes detection using existing traffic volume analysis method at time domain ever harder.

Figure 1. Periodic Characteristics of Shrew Attack Streams from Multiple Sources

As shown in Fig2-1, we have a shrew attack with the peak rate of 200Kb/sec and the attacking period of 1000ms. And we can see that the shrew attack stream hides itself among normal traffic by making its peak rate even lower than the normal traffic rate. Thus, before the link is saturated, the traffic volume analysis scheme may not be able to detect such a stealthy attack. However, the autocorrelation sequence will amplify the influence the periodical pattern of shrew attack stream has after it is converted into frequency domain. In Fig2-2, what exactly happens is that more power of the
autocorrelation function is distributed in the lower frequency band if there is shrew stream contained in the traffic.

Thus, this paper presents network flow grouping method based on the analysis above, and chooses the normalized cumulative amplitude spectrum (NCAS) value as its clustering feature (CF).

III. NETWORK ABNORMAL FLOW GROUPING METHOD

In the designed flow grouping model, the network flow data is sampled per user and his real-time flow every 1 ms. And then, the flow data is directly converted from the time series to its frequency-domain representation using Discrete Fourier Transform (DFT), and take its \( N \text{CAS}_{F, K} \), \( F \) is a constant Hz at K-Point as clustering feature. Grouping mothed takes use of BIRCH algorithm to group network flow into different user group based on above feature. The group merging algorithm overcame the case that same user belongs to different group, and yields the final grouping result.

A. Network Flow Clustering Feature Extraction

The core router starts to sample incoming packets per user flow and starts one timer as Fig3. Once the timer is expired, the router converts the time-domain series into it frequency domain representation using DFT, and the NCAS at K-point will pass to BIRTH algorithm as CF.

01: IF sampling is not done THEN
02: Continue sampling packets number per 1 ms;
03: ELSE
04: Convert the time-domain series into frequency domain;
05: Calculate the NCAS value at K-Point;
06: IF NCAS < Threshold THEN
07: Mark the flows as legitimate, routing it;
08: ELSE
09: Start BIRTH algorithm with NCAS value;

B. Grouping Network Flows by Clustering Feature

This paper is using BIRCH to group mass flow users into different groups. Given a set of \( N \text{CAS}_{F, K} \) data points, the clustering feature of the set is defined as the triple \( CF = (N, \sum (\text{CAS}_{F, K}) \cdot \sum (\text{CAS}_{F, K}^{-1}) \) , where \( \sum (\text{CAS}_{F, K}) \) is the linear sum and \( \sum (\text{CAS}_{F, K}^{-1}) \) is the square sum of data points. Each non-leaf node contains at most B (Branching Factor) entries of the form \( \{ CF_i, Child \} \), where \( Child \) is a pointer to its \( i^{th} \) child node and \( CF \) representing the associated subcluster. The tree size depends on the parameter T (Threshold).

The algorithm scans all the leaf entries in the initial CF tree to rebuild a smaller CF tree, while removing outliers and grouping crowded subclusters into larger ones. And then, an existing clustering algorithm is used to cluster all leaf entries. Here an agglomerative hierarchical clustering algorithm is applied directly to the subclusters represented by their CF vectors. After this step a set of clusters is obtained that captures major distribution pattern in the data. At last, the centroids of the clusters produced in above step are used as seeds and redistribute the data points to its closest seeds to obtain a new set of clusters.

C. Group Merging

Thinking about the deficiency of BIRCH, which fails to handle the case of “same user, different group”, the clustering result has to be revised. Luckily, the user with same clustering feature always continues for a bit long time. That is, the possibility of grouping the user into the same group is pretty high. Thus, the following merging group schema is proposed: Firstly, all the users are divided into a group (known as current group) when starting to cluster. And every user is given a “Time to Live” (TTL, says \( T_i \)) and a initialized value \( 0 \).

Secondly, once a new set of groups is generated by BIRCH algorithm, each of groups executes intersection operation with current group. The one who has max intersection is merged into current group. In this new current group, if the user belongs to both group, its TTL is updated to \( T_i + 1 \), which should be less than \( T_{max} \). If the user only exists in the last current group, its TTL is down to \( T_i - 1 \), when \( T_i = 1 \), the user is cleaned from this group. If the user only exists in new merged group, its TTL equals to \( T_i \). After above steps, a new current group is here. Taking use of this method, if a user...
doesn’t belong to some group, it is cleaned out from these group after several clustering’s.

IV. SIMULATION EXPERIMENTS AND RESULTS

![Simulated Cloud Data Center Network Topo](image)

Figure 4. Simulated Cloud Data Center Network Topo

In order to verify the method having higher detection accuracy and faster response time, and less impaction to normal network flow groups, the following simulation scenario is designed. In cloud data center of Fig4, three kinds of business flow types are used by 1000 users, and 300 users of them are abnormal ones. They start shrew attacks with the peak rate of 200Kb/sec and the attacking period of 1000ms, which can lead to whole data center network traffic jam.

A. Detection Accuracy of Flow Grouping Method

In order to explain that the proposed method has well detection accuracy, we define the detection accuracy ($\alpha$) as the ratio between abnormal users and all users in the abnormal group.

\[
\alpha = \frac{\text{Detected number of abnormal users containing shrew streams}}{\text{Total number of users in the abnormal group}}
\]

Fig5 presents the simulation results. Before shrew attacking, the ratio is almost the same with the pre-defined ratio. While shrew attacking starts, the ratio bumps to 70%, even 100% after a little learning time. That is, the group method has a high detect accuracy.

![Detection Accuracy Line of the Abnormal Group](image)

Figure 5. Detection Accuracy Line of the Abnormal Group

B. Response Time of Flow Grouping Method

The response time is a critical parameter to evaluate the performance of the flow grouping method. Here, we define it as the time when grouping method detects whether malicious flows exists or not. When the shrew attacking starts, the abnormal user number is increasing in the abnormal group. And in simulation 5 ms, the detected number is larger than 200 as Fig6.

![Abnormal User Number Line During Shrew Attacks](image)

Figure 6. Abnormal User Number Line During Shrew Attacks

V. CONCLUSIONS

The presented network flow grouping method in cloud computing based frequency-domain feature can resolve the deficiency of BIRCH’s lacking of soft clustering. And it is capable of blocking malicious shrew flows with accuracy greater than 70% and response time less than 5ms. But the threshold $T$, time to live $T_t$ and branching factor B are all needed to well tune manually per real network environment. Thus, we plan to investigate an efficient method to help work out above values in continued work.

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(grant number: NRF-2011-0023076). And it also supported by the BB21 project of Busan Metropolitan City.

REFERENCES


[8] Babhuiya FA, Gupta V, Biswas S. Detection and Mitigation of Induced Low Rate TCP-Targeted Denial of Service Attack, the 2012 IEEE Sixth International Conference on Software Security and Reliability, 2012

ZengGuang Liu
He received the B.S. and M.S. from Dept. of computer engineering, University of ShangHai for Science and Technology, China in 2005 and 2008 respectively. He is a senior software engineer at IP platform dept. of Alcatel-Lucent, QingDao, China from 2008. His research interests include Operation System, real-time communication and network security area.

XiaoChun Yin
She received the B.S. degree in education and technology from Qufu Normal University, Qufu, China in 2004, and received the M.S. degree in education and technology from Nanjing Normal University, Nanjing, China in 2007. She had been working as a lecturer in Weifang University of Science & Technology, China from 2008 to 2012. Currently she is a doctoral candidate in cryptography and network security at Dongseo University, Korea. Her research interests include network security, cloud security, authentication protocol and real-time communication.

Hoon Jae Lee
He received the B.S., M.S. and Ph.D. degree in Electrical Engineering from Kyungpook national university in 1985, 1987 and 1998, respectively. He had been engaged in the research on cryptography and network security at Agency for Defense Development from 1987 to 1998. Since 2002 he has been working for Department of Computer Engineering of Dongseo University as an associate professor, and now he is a full professor. His current research interests are in security communication system, side-channel attack, USN & RFID security. He is a member of the Korea institute of Information security and cryptology, IEEE Computer Society, IEEE Information Theory Society and etc.