DOI: 10.5604/20830157.1121390

MODIFICATION OF TCP SYN FLOOD (DOS) ATTACK DETECTION ALGORITHM

Tomáš Halagan, Tomáš Kováčik
Slovak University of Technology in Bratislava

Abstract. *This work focuses onto proposal and implementation of modification of SYN flood (DoS) attack detection algorithm. Based on Counting Bloom filter, the attack detection algorithm is proposed and implemented into KaTaLyzer network traffic monitoring tool. TCP attacks can be detected and network administrator can be notified in real-time about ongoing attack by using different notification methods.*

Keywords: DoS, TCP, SYN, flood attack, network security, notification messages, detection module

MODYFIKACJA ALGORYTMU WYKRYWANIA ZMASOWEGO ATAKU TCP SYN FLOOD (DoS)

Streszczenie. *Praca koncentruje się na propozycji implementacji modyfikacji algorytmu wykrywania zmasowanego SYN flood (DoS) ataku. Bazując na filtrze typu Counting Bloom, zaproponowano algorytm wykrywania ataku i zaimplementowano w narzędziu monitorowania ruchu w sieci - KaTaLyzer. Ataki TCP mogą być wykrywane i administrator sieci może być powiadamiany w czasie rzeczywistym o przeprowadzanym ataku dzięki różnym metodom powiadamiania.*

Słowa kluczowe: DoS, TCP, SYN, zmasowany atak, zabezpieczenia sieci, komunikaty powiadamiania, moduł wykrywania

Introduction

Computer network security and privacy is currently on very high level due to various detection algorithms or protection mechanisms implemented on various network layers, network devices and directly in operating systems. Despite of all these, the most important factor remains in information about currently ongoing attack which administrators need to have in adequate time. Hence, this area is a space to create new effective solutions to detect and provide such information.

In this work we focus on proposal of modification of SYN flood attack detection algorithm and its implementation into KaTaLyzer network traffic monitoring tool which has been developed at STUBA [1]. Based on Counting Bloom Filter (CBF) mechanism, our contribution is in modification and usage of the CBF of its tables which are used for storing counters of half-open connections (TCP). After detecting ongoing attack, network administrator is notified by chosen notification methods.

It is assumed familiarity with the issue of DoS SYN flood attack. In case you are not familiar with DoS SYN flood attack, you can find more information at this link [2].

1. Bloom filter and its modification

1.1. Bloom Filter algorithm

Mathematics behind the Bloom Filter data structure is following: consider a set of *m* elements, in our case a set of *m IP* addresses, $IP = \{ ip_1; ip_2; ip_3; \dots ip_m\}$. The set will be described by a vector *V* which is *n* bits long, it is initially set to *n* zeros. Next, consider *k* independent hash functions which are used by the Bloom filter to generate *k* hash values from each of elements of the *IP* set. The hash functions output values are from range $\{1, \dots, n\}$ and represent index in the vector *V*. If i^{th} hash function $h_i()$, $1 \le i \le k$, applied to members of *IP*, $ip_j \in IP$, $1 \le j \le m$ results in value *K*, i.e. $K = h_i(ip_j)$, K^{th} bit of the *V* vector is set to 1. For each element $ip_i \in IP$, $1 \le i \le k$, K^{th} bit of *V* vector is set to 1, while $K = h_i(ip_j)$, for each $1 \le j \le m$, $1 \le i \le k$.

According to relations among the hash functions and overlapping of their results, bits in *V* vector can be set to 1 multiple times. However, only the first setting of K^{th} bit to 1 changes the value of the bit.

1.2. Counting Bloom Filter

Assume a situation in which the elements of *IP* set change periodically and thus they are being inserted to and deleted from the data structure. Inserting elements is a simple process which we described above. When we want to delete an element from Bloom filter data structure, we need to set the corresponding bits to zero. However, it is possible that this operation will affect bits which were set to 1 by hash function for different element of *IP* set. In this situation the Bloom filter no longer provides correct representation of the elements of the *IP* set.

This problem was solved in work [2] which outlined new data structure called Counting Bloom Filter (CBF). In the CBF data structure, bits in vector *V* are replaced by integers which are used as counters. If we want to save a track of element ip_x in the data structure, each counter corresponding value of independent hash functions will be incremented. During deletion of an element appropriate counters are decremented.

1.3. Modified CBF

Simplification of the CBF structure is one of our contributions. The solution consists of one table of counters used for inbound and one table of counters used for outbound TCP connections. Such simplification can be done as in defined short time interval the tables of counters are re-initialized to 0. Increment and decrement of the counters are designed to fit the proposed solutions to detect SYN flood attack and these will be described in later section S-Orthros detection algorithm.

1.4. Independent Hash Functions

Finding well designed set of hash functions is as important for the correct storage and distribution of elements in the CBF data structure as well as for the performance of hash functions. In a comparative study performed by Chen and Yeung in paper [1], there are independent hashes functions designed which have low probability of collision. 32-bit *IP* address is used as a key for the hash functions.

The hash functions are defined as follows:

$$hi(IP) = (IP + IP \bmod pi) \bmod n, \; 1 \le i \le k,$$

where mod denotes the modulus operation, n is the row length of the hash table, and pi is a prime number less than n.

Following table shows comparison of the proposed hash function with other known hash functions.

Our examination resulted in setting variables as follows: n=1024, k=4.

Table 1. Table tested hash functions [1]

Hash function	Consumed time (s)	Number of collisions
Our function	0.187	0
Robert's 32-bit function	0.188	3838
Robert's 96-bit function	0.250	0
Cruth's function	0.031	977
Hybrid function	0.328	0

2. Detection algorithm S-Orthros

As mentioned above, the algorithm for detecting SYN Flood attack uses CBF data structure which is modified for our attack detection purposes. Modification of the data structure has yielded to simplification and clarification of the solution and also the algorithm itself. The intention is therefore the evaluation of the conditions and thus attack detection for a given constant time interval.

Consider the case where the detection algorithm cooperates with a measuring tool which is used to capture and analyze network traffic in real time. At the beginning, the network administrator starts the process of capturing network traffic which runs as the main program. Other processes are invoked and run as separate threads. At the end of the main process of capturing network traffic also other running threads are closed.

After a defined time, another process is run to analyze the measured network traffic information. In this process, the data which are important for the detection algorithm S-Orthros are being stored. In this case, the relevant data are source and destination IP addresses and these are saved through the independent hash functions to the modified CBF. The modified CBF consists of two tables containing N counters.

The first table is used to store information about the source *IP* address while the second table provides space to store destination *IP* addresses. During the analysis, the detection algorithm S-Orthros collects information about initiated connection, i.e. SYN packet and information of confirmation of the connection, i.e. ACK packet. If analysis detects SYN packet, the data structure is incremented - both tables of counters are modified. In case of ACK packet confirming SYN packet, data structures in both tables are decremented. If there is no flood attack, the handshake is correct, and data structures remain empty - a pair of *IP* addresses in the first SYN packet is stored, and after confirming ACK packet they are cleared.

In case of flood attack, values in CBF structure are very rising. Threshold of acceptable half-open connections has been empirically set to 50. Verification process compares this threshold value with the measured SYN-SENT value and according to the result either starts or does not starts the process of creating an alert message.

3. Evaluating proposed solution

Through Modified CBF we are able to distinguish different variations of SYN flood attacks. We have identified the following possible TCP SYN attacks:

- Random SYN flood – generating a random source spoofed IP address for each packet
- Subnet SYN flood – generating source spoofed IP address from specific subnet for each packet
- Fixed SYN flood - several IP addresses from which the attack is launched

Each one of these attacks can be identified thanks to the typical arrangement of IP addresses stored in the used data structure. This arrangement of IP addresses has been verified by table editor on sample of 250.000 IP addresses and the results are shown in the following figure 1 shown below.

We can clearly see that the system is being attacked as the values are much higher than 0 level which describes usual traffic.

4. Conclusion

We proposed and implemented detection module S-Orthros based on modified Counting Bloom filter, which provides information about currently ongoing TCP SYN attack to network administrator. Detection module has been tested for several TCP SYN attacks and its functionality has been verified and evaluated. Within the detection module has been proposed and implemented various methods for the timely sending of information about the ongoing attack. For its extensiveness they have not been described in this paper.

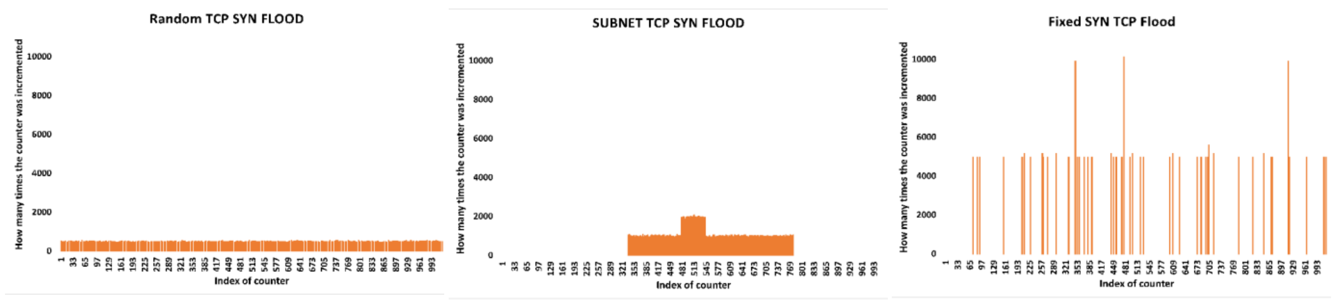


Fig. 1. TCP SYN attacks

Acknowledgements

This work has been supported by ngnlab.eu project, 7FP project HBB-Next, Slovak National research grant 1/0676/12. It is a partial result of the Research and Development Operational Program for the projects Support of Center of Excellence for Smart Technologies, Systems and Services, ITMS 26240120005 and Research and Development Operational Program for the projects Support of Center of Excellence for Smart Technologies, Systems and Services II, ITMS 26240120029, co-funded by ERDF. This work is also part of Katalyzer project www.katalyzer.sk and ngnlab.eu initiative.

References

[1] Chen Y. Y. W.: Throttling spoofed SYN flooding traffic at the source. Telecommunication Systems, vol. 33, no. 3, 2006, pp. 47-65.

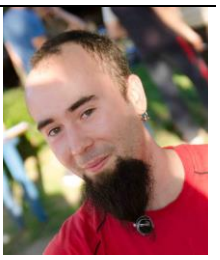
[2] Fan L. et al.: Sumary cache: A scalable wide-area web cache sharing protocol. IEEE/ACM Transactions on Networking, vol. 8, no. 3, 2000, p. 281-293.

[3] [CA-96.21] CERT. CERT Advisory CA-1996-21 TCP SYN Flooding and IP Spoofing Attacks: http://www.cert.org/advisories/CA-1996-21.html

[4] Network monitoring tool Catalyzer: http://www.katalyzer.sk

Ing. Tomáš Halagan
e-mail: tomas.halagan@stuba.sk

Tomáš Halagan was born in Snina in 1988; he is currently a Ph.D. student at the Slovak University of Technology in Bratislava. His general research area is Software Defined Networking (SDN) and Network Function Virtualisation (NFV) synergy. He is member of research team specializing in computer and wireless networks and that is partially supported by Slovak National research grant and by Tatra Banka foundation.



Ph.D. Ing. Tomáš Kováčik
e-mail: tomas.kovacik@stuba.sk

Tomáš Kováčik (Ph.D.) born in Topoľčany in 1984, graduated in telecommunications in 2008 and received his Ph.D., both from Slovak University of Technology in Bratislava, in 2012. His research and teaching activities include IP telephony, IP multimedia subsystem, hybrid television and mobile applications. He participated in several Slovak national research projects, in EU FP7 HBB Next project and is part of ngnlab.eu initiative.

otrzymano/received: 09.04.2014 *przyjęto do druku/accepted: 11.06.2014*