International Journal of Computer Science, Communication & Information Technology (CSCIT)
Vol.7 pp. 7-12

An Efficient Approach to Secure Cloud Computing

Haddadi Mohamed[1,2,**]
[1] Département des sciences commerciales, Faculté des sciences économiques, commerciales, et sciences de gestion, Université de M'hamed Bougara de Boumerdes, Avenue de l'Indépendence, Boumerdes 35000, Algérie

Beghdad Rachid[2,*]
[2] Département d'informatique, Faculté des sciences exactes, Université de Bejaia, Bejaia 06000, Algérie

** haddadimohamed2013@gmail.com, m.haddadi@univ-boumerdes.dz *rachid.beghdad@gmail.com

***Abstract*— Cloud Computing is more and more used in the organizations because it can reduce the cost and complexity of applications. On the other hand, such complex and distributed architectures become an attractive target for attacks, such as flooding based Distributed Denial of Service (DDoS) attack, which represents a serious danger that can deny the legitimate users to access the service delivered by cloud. This paper proposed an Improved version of Hop Count Filtering (IHCF) technique to detect DDoS attacks, especially attackers that generally do not bother to spoof IP addresses. The proposed algorithm is implemented in cloud lab by using VMware, such as Virtual Machine Manager (VMM) and JAVA application. Compared to the original HCF technique and its variants, IHCF decreases the false negative and positive rates and consequently increases the detection rate of flooding attacks to 94% with low computation time.**

Keywords—cloud computing; cloud computing security; VMware architecture; DDoS attack; IP spoofing; IHCF.

I. INTRODUCTION

Nowadays, Cloud computing is a long-held imagination of computing as a utility. Because of this, it becomes more and more used for its advantages, such as access on-demand resources which means that you can consume resources as a service anytime and from anywhere and pay only for resources that you use by only your personal computer and internet connection.

Currently, Cloud computing is a model to provision on-demand network access to a shared pool of computing resources that can accommodate varying end user demands with minimal service provider intervention [1]. The services provided by the cloud are categorized as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) [2]. Generally, these services become unavailable at significant time because of security issues.

In practice, there are many types of DDoS attacks such as SYN flood, UDP flood and ICMP flood. Indeed, these attacks can be a serious threat to the resources canters [3]. Mostly, they generate a huge amount of attack packets towards the target server by using generally IP spoofing technique. The main purpose of the attacks is to block the legitimate access during a long period of time.

In the last two decades, serveral researches on DDoS attacks have been worked and lots of new techniques have been put forward, but their detection methods are able to recognize packets that match a known signature database. So, these techniques are vulnerable to flooding attacks which needs new defense mechanisms that should be able to differentiate attack packets from legitimate ones.

Practically, it is crucial to detect accurately DDoS attacks. This is due to various issues in getting significant performance metrics. So, some techniques have been designed to limit the efficiency and the effectiveness of DDoS attacks. But, they have not been generally implemented. On the other hand, there are various performance issues, trade-offs in placing these defense techniques and so on.

In this paper, our proposed algorithm (IHCF) works in the IaaS layer of the cloud stack. So, It changes the alert state of HCF to include all the possible Hop Count (HC) values. On the other hand, instead of using <*IP, HC*> we use <*IP,* HC_{list}*, count*>. This is because of using multiple alternative paths to the same destination, which can yield a variety of allowed HC values. Moreover, we use a discarding threshold (*thd*) that depends on server's workload when the communications are correctly established. This discarding threshold is computed at the end of each slot time by using statistical properties, such as variance and standard deviation. The source IP addresses are extracted from IP and TCP headres. In addition, we utilize a blacklist which aims to reduce the computation time of each attack packet. This backlist includes two fields, namely, *srcIp and xtime*. The *srcIp* is the attack source IP. The *xtime* is an amount of time. We make the source IP in the blacklist when its number is greater than the *thd* during a slot-time.

The rest of this paper is organized as follows: section II introduces some related works cited recentely whereas Section III provides a design of the proposed Algorithm (IHCF). Experiments and evaluations are given Section IV. Finally, section V covers a brief conclusion of our work.

II. LITERATURE RIVIEW

This section provides several techniques available in recent literature that detect flooding based DDoS attacks. Each of which comes with its benefits and limitations. We begin with the proposal of Wang et al. [4] who proposed the HCF algorithm to detect IP spoofing. The algorithm is based on the idea that although an attacker can spoof the source IP, the attacker cannot spoof the number of hops a packet traverses as it moves from sender to receiver. HC is not sent in the IP

ISSN 1737-930X

packet but is rather inferred from the TTL (Time To Live) field. The receiver can estimate the HC by subtracting the received TTL value from the closest initial TTL value bigger than the received packet's TTL. Usually, these initial TTL values are operating system dependent and are limited to few possibilities, which include: 30, 32, 60, 64, 128, and 255 [5]. Therefore, the algorithm first learns the IP to HC mapping and stores the mapping in an IP2HC table. Once a packet arrives, it is compared to the HC stored for this IP. If the HC values match, then the packet is legitimate. Otherwise, the packet is discarded. In general, the algorithm captures roughly 90% of spoofed packets. I believe improvements are needed to enhance the approach. One shortcoming is that this approach only works with attacks that involve IP spoofing. Another potential gap is that the author did not study the case where the packet uses multiple alternative paths to the same destination, which can yield a variety of allowed HC values. In that case, the detection rate (90%) may be systematically lower. Moreover, the high number of false positives identified through dropping of a large number of IP addresses belonging to legitimate users.

Another approach developed is by Karnwal et al. [6], who proposed a comber approach called filtering tree, to secure cloud against application layer attacks, such as HTTP-XML-DDoS attack. This proposed scheme includes five steps as follows: sensor filter, HC filter, IP frequency divergence filter, confirm legitimate user IP filter, and double signature filter. The proposed approach has a few shortcomings. The first being its vulnerability to attackers that generally do not bother to spoof IP addresses. A second shortcoming is that there is a negative impact, due to the additional processing to the network traffic. A third shortcoming is that the author did not study the case where the packet uses multiple alternative paths to the same destination, which can yield a variety of allowed HC values.

Vikas et al. proposed in [7] a filtering scheme which shares similar filtering ability as that mentioned in [4], but in addition to that, it extracts synflag to include four cases for each captured packet. The results show that the proposed scheme can save computation time, but it has a few limitations as [4]. Detection rate, false positive and negative rates were unreported in this paper.

Chapade et al. [8] proposed a classification scheme of network traffic into legitimate and malicous by using Mean Absolute Deviation (MAD) of TTL values. Their simulaton results show that the proposed scheme has a high detection rate with low false positive rate. MAD has some limitations. The first, it cannot recognize forged packets whose source IP addresses have the same HC value as that of a zombie. Moreover, an attacker may circumvent it entirely by not using spoofed traffic or partially by bombarding a victim with much more attacking traffic.

A framework called SBTA (SOA-Based Traceback Approach) [9] was proposed by Yang et al. It is a combination of SBTA and CF (Cloud Filter). The use of SOAP is to traceback the source of DDoS attacks and CF for filtering the attack traffic. The proposed scheme has a high efficiency and effectiveness for detecting and identifying the attack traffic, and high detection rate with low false positives. It has one shortcoming being its vulnerability to IP spoofing, i.e., an attacker can use a spoofed IP belonging to unreachable user that renders the path reconstruction undetermined during the attack.

Joshi et al. [10] proposed a method to traceback the source of attack traffic by using Cloud TraceBack (CTB) and employs neural network (Cloud Protector) to detect and filter the attack's bases on DDoS. The proposed scheme has one shortcoming being its vulnerability to IP spoofing, i.e., an attacker can use a spoofed IP address belonging to unreachable user which renders the path reconstruction undetermined during the attack.

Karnwal et al. [11] who proposed a filter Tree approach to protect cloud against HTTP-XML DDoS attacks. They present a Cloud Defender which uses IP addresses to recognize and traceback the illegitimate virtual machines (VMs). The Cloud Defender includes five steps as follows : sensor filtering, HC filtering, IP frequency divergence filter, confirm legitimate user IP filter, and double signature filter. The proposed approach has a few shortcomings. The first being its vulnerability to attackers that generally do not bother to spoof IP addresses. A second shortcoming is that the author did not study the case where the packet uses multiple alternative paths to the same destination, which can yield a variety of allowed HC values.

Doua et al. [12] proposed a Confidence Based Filtering method (CBF) to prevent DDoS attacks in cloud environment. This method monitors the transport and network layers and it based on few correlation characteristics of attributes in the IP and TCP headers during two periods, i.e., non-attack period and attack period. The simulation results showed that the model has a high efficiency and low storage requirement when working with high workload networks.I believe improvements are needed to enhance the approach. One shortcoming is that there is a negative impact, due to the high processing.

A framework called HCF-CBF (Hop-Count Filtering and Confidence Based Filtering) [13] was proposed by Mamtesh et al. First, HCF is used to detect spoofed packets. After, CBF is used to detect attackers that do not bother to use IP spoofing and that they were not detected by HCF. The simulation results show that the scheme is efficient and effective against DDoS attacks. I believe improvements are needed to enhance the framework. One shortcoming is that there is a static discarding threshold issue.

A novel probabilistic packet marking scheme [14] was proposed by Abdullah et al. It infers forward paths from attacker sites to a victim site. The simulation results show that the technique can construct the forward path from an attacker site after receiving 20.23 packets on the average for DoS attacks. I believe improvements are needed to enhance the technique. One shortcoming is that the technique can consume more bandwidth because it utilizes the 40 bytes record route options field of the IP header.

The above DDoS detection techniques, which are recently discussed in this section, are summarized in TABLE I that provides a few limitations of them.

ISSN 1737-930X

TABLE I
DDOS DETECTION TECHNIQUES SUMMARY

Approach	Prevention Technique	Limitations
HCF [4]	Packet Filtering	-Varied allowed HC values problem -False positive problem -Problem of attackers that do not bother to spoof IP
Filtering Tree [6]	Packet Filtering and variation	- Attackers that do not bother to spoofIP -Varied allowed HC values problem
HCF [7]	Packet Filtering	-Detection rate, FN and FP were unreported -Attackers that do not bother to spoof IP -Varied allowed HC values problem
MAD [8]	Distance based	- Attackers that do not bother to spoofIP
SBTA & CF [9]	Traceback method	-IP spoofing problem
CTB&CP [10]	Traceback Method and Neural Network	-IP spoofing problem
Filtering Tree [11]	Packet Filtering and variation	-Varied allowed HC values problem -Problem of attackers that do not bother to spoof IP
CBF [12]	Packet Filtering	-Negative impact, due to the processing
HCF-CBF[13]	Packet filtering & variation	-Static discarding threshold issue.
Probabilistic model [14]	Traceback method	-Consumes more bandwidth.

III. DESIGN OF THE PROPOSED ALGORITHM

A flow chart diagram of the proposed algorithm is shown in the Fig. 1. It consists of the following four levels as follows:

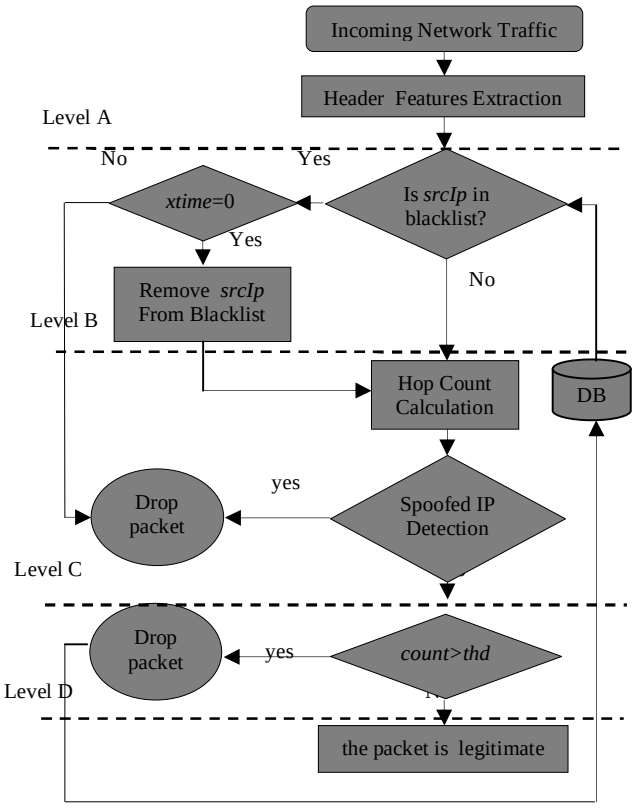

Fig. 1 Flow chart diagram of the proposed algorithm (IHCF).

A. Level A

In his level, *srcIp*, TTL_f value and *synflag* are extracted from each packet of incoming network traffic.

B. Level B

This level can reduce the computation time of each attack packet by using the blacklist. the packet is forwarded to the level C, if *srcIp* is not in the blacklist or *xtime* = 0. Otherwise, the packet is droped, If *xtime*≠0. The *srcIp* is removed from the blacklist when *xtime=0.*

C. Level C

In this level, TTL_f value is used to compute the number of hops that the packet has travelled (*HC*). An attacker can spoof the packet header, but not able to manipulate the *HC* value. Then, itBy compares the *HC* value with the stored Hop Count (*HCs*) into the $IP2HC_{list}$ table that have the same *srcIp*. If no accurate matches are found, then the packet is spoofed so discard it directly, else the packet is forwarded to the level D.

D. Level D

This level is used to verify the number of packets of the same *srcIP* during the same slot time into the the $IP2HC_{list}$ table, if the *count* > *thd*, the packet is spoofed, so delete *srcIp* from $IP2HC_{list}$ table and add it in the blacklist, else the packet is legitimate.

To help illustrate our proposed algorithm. TABLE II shows few parameters that have been used in the experiments.

TABLE II
IHCF ALGORITHM PARAMETERS

IHCFitems	Utilization
Nominal Profile	- To include all the possible available Hop Count values <*srcIp, HC$_{list}$, count*>. -To compute the first *thd*.
Dynamic Threshold (*thd*)	-To calculate before starting attack detection and after. -To compute the number of packets during a slot time. -Depends on server's workload when the connections are correctly established.
Counter (*count*)	-To compute the number of packet IP during a slot time in the $IP2HC_{list}$ table for detecting attackers that do not bother to use IP spoofing. - We initialize all their counters (*count*)in the $IP2HC_{list}$ table by zero (0) in the end of the slot time for restarting the computation.
Blacklist	-Contains attackers that do not bother to use IP spoofing when its count greater than a threshold during a slot time. -Minimizes the computation time and updates in $IP2HC_{list}$ table -We remove from it every IP address that is not used by an attacker during X amount of time.
***HC*$_{list}$**	-To contain a variety of allowed HC values

IV. PERFORMANCE EVALUATION

In this section, we will test the proposed algorithm (IHCF) in a cloud lab as shown in Fig. 2 in order to study its performance against DDoS attacks. We show and analyze the results obtained from experiments by taking into account the comparison with others like HCF [4], HCF [7], and CBF [12].

ISSN 1737-930X

A. Experimental Environment

In this subsection, we build a cloud lab by deploying one server, namely, a target server, which is a container of cloud services hosted in the form of two virtual machines (Client VM1 and Client VM2) that have windows 7 and windows xp like OSs (Operating Systems) respectively. These VMs were created by VMware ESXI 5.0.0 Hypervisor which is a Virtual Machine Manager (VMM).

Data Base (DB): The DB is implemented using MySQL and has two tables, namely, $IP2HC_{list}$ table and blacklist table. The $IP2HC_{list}$ table includes three fields, namely, *srcIip*, HC_{list} and the *count*. The blacklist table includes two fields, namely, IP address and a default value (*xtime)* which decreases automatically.

Three legitimate users, namely, User1, User2 and User3 which are generating normal traffic, such as UDP traffic.

Two attackers: The attackers are using attack generation algorithm as shown in the Fig. 3**,** which is built by using a combination of java libraries like Jpcap and WinPcap. There are a few routers between attackers (users) and that specific router (R).

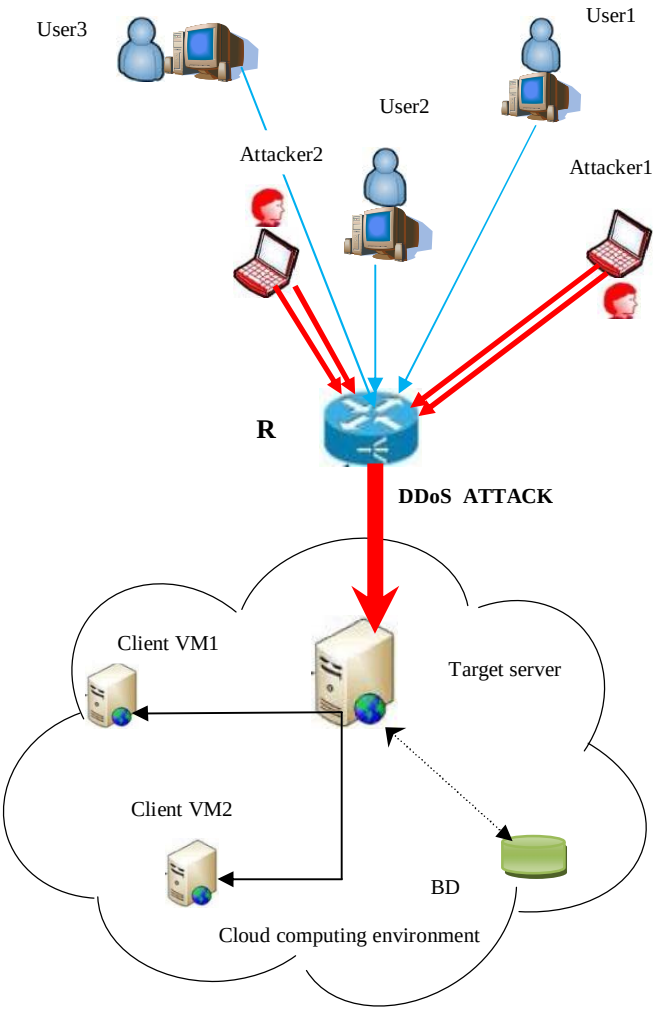

Fig. 2 Architecture of our experimentation with VMware ESXI 5.0.0

B. Traffic generation

In this subsection, we use DDoS attack generation algorithm which is a combination of three types of flooding based DDoS attacks used in the experiments. Each one sends

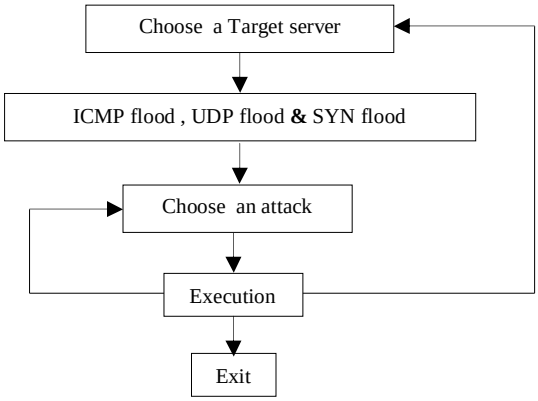

Fig. 3 DDoS attack generation algorithm

a huge number of packets to the target server. Haddadi at al. [15] summarized the following DDoS attacks.

1) TCP syn flood attack : It is an important form of DDoS attacks. It generates a huge number of SYN packets (connection requests) without ACK packet to the target server to consume all available TCP connection queues.

2) UDP flood attack: It continuously sends a large number of UDP packets to random ports on the target.

3) ICMP flood attack : It simply sends a large number of ICMP packets by broadcasting to the target.

To help generate these attacks, a combination java library like Jpcap based on WinPcap that interacts with the OS and NIC (Network Interface Card) to capture packets.

C. Software configuration

The experiments were processed within a NetBeans IDE 8.0.1 environment where a combination of Jpcap and WinPcap java libraries were carried out. Jpcap (Java packet capturing) is an open source java library for transferring and capturing network packets to establish applications for capturing packets from a network interface and explore them in java. Over them, the proposed algorithm (IHCF) was installed for packet monitoring in the cloud environment by extracting only TTL_f value, *synflag* and *srcIp* of each captured packet. At the beginning, all non spoofed packets are allowed in the learning period and all their headers (*srcIp*, TTL_f value and *synflag*) are extracted to include all the possible available *HC* values for each *srcIp* and compute the *thd* to verify the frequency counter (*count*) of each captured packet, if it exceeds it or not during a slot time. The fixed discarding threshold is feasible if there is no variation in the server's workload level during a slot time, but the dynamic discarding threshold can be adopted because of a variation in the each server's workload level. Selecting an inaccurate value of *thd* may raise false alarms. If the value is too low or if it is too high, it can cause the legitimate traffic being considered as malicious traffic. All packets that do not respect the precedent *thd* are directly

ISSN 1737-930X

discarded in order to stop similar packets to bypass the cloud lab. This might decrease the amount of false positives and false negatives to a small extent and increase the detection rate (> 90%) which means that the ability of the algorithm to detect attacks over the total amount of attacks.

D. Performance Metrics

Performance indicators [16] for the flooding based DDoS attacks are: False Negative (FN), False Positive (FP), Detection rate (Dr), True Positive (TP), and Computation Time (CT).

- **CT:** Is factor for performance measurement of cloud network and it improves processing relevant power of cloud server and minimizes loss of available resources.
- **FN:** Is an anomalous behavior we failed to detect.
- **FP:** Is that the legitimate client IP address which is incorrectly identified as spoofed.
- **TN:** Represents that the normal behavior which is correctly predicted as normal.
- **Dr**=TP/(FN+TP).

E. Simulation Results and Analysis

In the following, we simulate results between the proposed algorithm (IHCF) and a few previous algorithms that have been discussed in literature, such as HCF [4], HCF [7], and CBF [12] in term of the following performance metrics.

1) Computation time : The sample inputs are taken as an arrival rate in mseconds, various results have been analyzed and presented in Fig. 4.

For HCF [7], it has a very good performance, especially in samples 3, 4 and 5. This means that the sample 3 needs more time than sample 4 and 5 because it depends on receiving field of packets compared to the proposed algorithm (IHCF) which is continuously increasing. For CBF [12], it has a negative impact, due to the processing, i.e., examining the Packets for extracting header fields, such as TTL_f value, computing the confidence values, and updating the nominal confidence, if the new confidences are used in each time interval which can add additional processing to the network traffic compared to the proposed algorithm which does only some verifications in the first slot time during the attack. In HCF [4], the computation time rests in continuous increasing because the author did not study the issue of updates of IP to HC by using packets from established TCP connections (*synflag*) which ensures that an attacker cannot slowly pollute a $IP2HC_{list}$ table by spoofing source IP which can increase the processing time of each IP packet in comparison to the proposed algorithm (IHCF) which take into account this issue.

According to the Fig. 4 which shows that the proposed algorithm has better computation time in the high rate compared to other ones. In the other hand, It also obtained a desirable results that are surly adopted in the cloud computing environment. The usefulness of the proposed algorithm (IHCF) is that there is a bad performance in the beginning in comparison to other ones. This is because of several verifications, such as blacklist especially in the first slot time during the flooding attack. So, the proposed algorithm seems clearly to have a very good performance when there are several attackers that do not use IP spoofing technique.

2) Detection rate : Fig. 5 shows a few comparisons that have been done between the proposed algorithim (IHCF) and other ones in term of detection arte. So, they are mentioned as follows:

In the proposed algorithm (IHCF), it was approximately 94%: It means a very good detection rate.

In the HCF [7], it was approximately 86%: It means a bad detection rate. In the HCF [4], it was approximately 85%: It means a worse detection rate. In the CBF [12] algorithm, it was approximately 89%: it means a good detection rate.

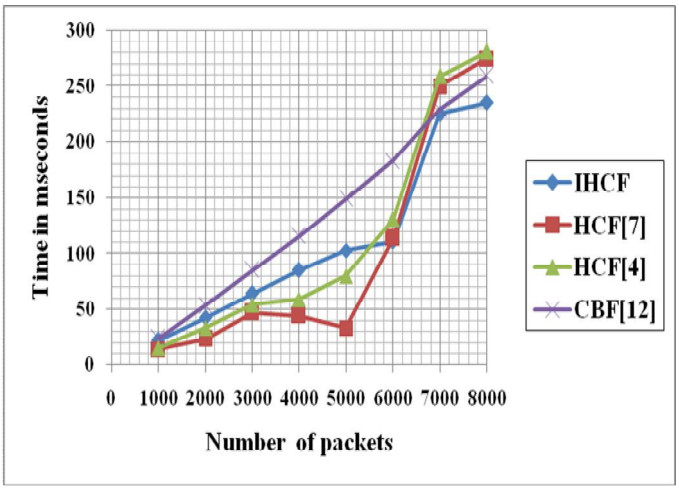

Fig. 4 Computation time comparison between the proposed algorithm (IHCF) and other ones

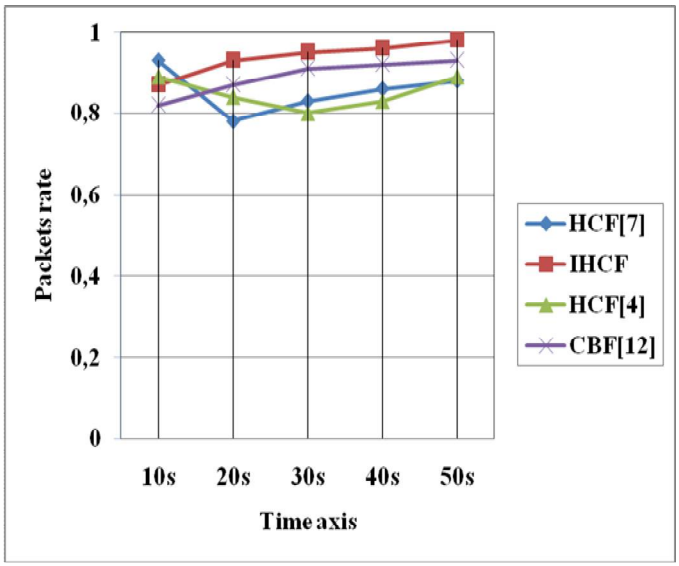

Fig. 5 The effect of TCP SYN flood attack in the proposed algorithm (IHCF) and other ones

ISSN 1737-930X

3) False positive rate:

In the proposed algorithm (IHCF), all packets are nearly identified correctly, so the false positive rate is approximately equal to zero (≈**0**). But in HCF [4] and HCF [7], they are approximately equal to 10%, so that they are very bad. In the CBF [12], it is equal to 7.7% so that it is bad.

In the beginning of the action state, HCF[4], HCF[7], and CBF[12] have better performance than our proposed algorithm (IHCF) in terms of detection rate which means that 0,93[7], 0,89 [4] and 0,93 [12] against 0,87(IHCF).

*4) False negative rate :*The proposed algorithm has only 6%. But in HCF[4] and HCF [7], they are equal to 5% and in CBF [12], it is equal to 7.7%.

According to this comparison, we clearly observed that the proposed algorithm (IHCF), in the detection rate and false positive rate, has better performance in comparison to the other ones (HCF [4], HCF [7], and CBF [12]). But in false negative rate, HCF [4] and HCF [7] have better performance in comparison to the proposed algorithm (IHCF). The results show that the proposed algorithm (IHCF) has one problem in terms of false negative rate.

The performance of flooding based DDoS attack detection schemes may be measured in terms of false positive and negative rates and detection rate as shown in Fig. 6.

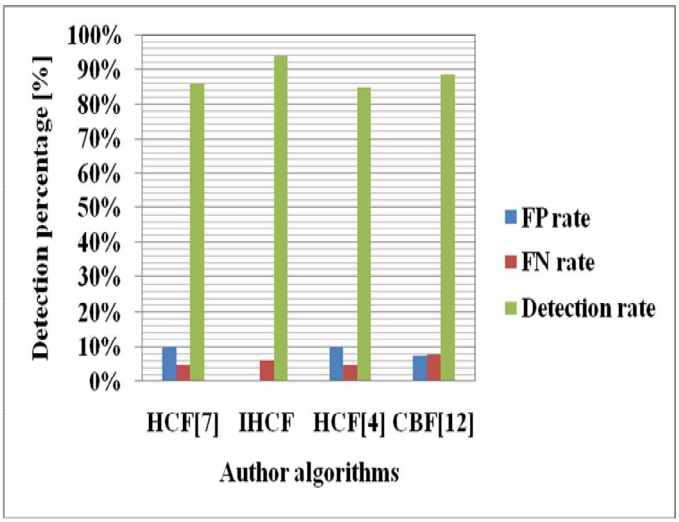

Fig. 6 Comparison between the proposed algorithm (IHCF) and other ones in terms of detection rate, false negative and false positive rates

V. CONCLUSION

In cloud infrastructure service where several users can share the same infrastructure which can cause DDoS attacks. Because of this, IHCF was proposed in the purpose to detect DDoS attacks, especially attackers that generally do not bother to spoof IP addresses. Firstly the proposed algorithm changes the alert state of HCF technique to include all the possible available HC values. Another parameter is used, instead of using <*IP*, *HC*> we use <*IP*, HC_{list}, *count*> because of using multiple alternative paths to the same destination, which can yield a variety of allowed HC values. Moreover, we added a counter column (*count*) into the IP2HC table for computing the number of the each packet IP during a slot time. In addition to that, we use a blacklist where we make *srcIp* that is detected like legitimate and forwarded to target server. But its frequency counter (*count*) is greater than the *thd* during a slot time for facilitating the research of *srcIp* and *xtime* during the execution of the proposed algorithm (IHCF) levels. The *srcIp* has to be deleted from the $IP2HC_{list}$ table. Finally, we must initialize all their counters to zero (*count*=0) for restarting the research of attacker IP.

REFRENCES

[1] P. Mell, and T. Grance, " The NIST definition of cloud computing, " Natl. Inst.Stand. Technol. Special Publication SP 800-145 , 2011.

[2] F. Sabahi, "Cloud computing security management," 2nd International Conference on Engineering Systems Management and Its Applications (ICESMA), pp. 1-7, 2010.

[3] Kumar, P. A. R., & Selvakumar, S, "M_2KMIX: Identifying the type of high rate flooding attacks using a mixture of expert systems," International journal of Computer Network and Information Security(IJCNIS), vol. 4, no. 1, pp. 1-16, 2012.

[4] H. Wang H, C. Jin , and K. G. Shang, " Defense against spoofed IP traffic using hop-count filtering, " IEEE/ACM Transactions on Networking (ToN), vol. 15, no. 1, pp. 40-53, 2007.

[5] Kravets D. (2011,March) "U.N. Report declares internet access a human right," (accessed April 28, 2012) Available: http://www.wired.com/threatlevel/2011/06/internet-a-human-right.

[6] T. Karnwal, T. Sivakumar, and G.Aghila, "A comber approach to protect cloud computing against XML DDoS and HTTP DDoS attack," in Proceedings of the IEEE students' Conference on Electrical, Electronics and Computer Science (SCEECS), Bhopal, pp. 1-5, 2012.

[7] C. Vikas and S. K. Peddoju, "Packet monitoring approach to prevent DDoS attack in cloud computing,"International Journal of Computer Science and Electrical Engineering (IJCSEE), vol. 1, no. 2315-4209, pp. 38-42, 2012.

[8] S. S. Chapade, K. U. Pandey and D. S. Bhade, "Securing Cloud Servers Against Flooding Based DDOS Attacks," *2013 International Conference on Communication Systems and Network Technologies*, Gwalior, pp. 524-528, 2013.

[9] L. Yang, T. Zhang, J. Song, J. S. Wang and P. Chen, "Defense of DDoS attack for cloud computing," *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, Zhangjiajie, China, pp. 626-629, 2012.

[10] B. Joshi, A. S. Vijayan and B. K. Joshi, "Securing cloud computing environment against DDoS attacks," *2012 International Conference on Computer Communication and Informatics*, Coimbatore, pp. 1-5, 2012.

[11] T. Karnwal , S. Thandapanii, and A. Gnanasekaran , "A filter tree approach to protect cloud computing against XML DDoS and HTTP DDoS attack, " Intelligent Informatics, Springer Berlin Heidelberg, pp. 459-469, 2013.

[12] Doua, W., Chen, Q., Chen, J, " A confidence-based filtering method for DDoS attack defence in cloud environment," Future Generation Computer Systems(FGCS), vol. 29, no. 7, pp. 1838–1850, 2013.

[13] [1] R. Nath and M. T. Scholar, "An Improved Defense Mechanism Based on Packet Filtering to Mitigate DDOS Attack in Cloud Computing Environment," vol. 5, no. 4, pp. 239–243.

[14] A. Y. Nur and M. E. Tozal, "Record route IP traceback: Combating DoS attacks and the variants," *Comput. Secur.*, vol. 72, pp. 13–25, 2018.

[15] M. Haddadi and R. Beghdad, "DoS-DDoS: TAXONOMIES OF ATTACKS, COUNTERMEASURES, AND WELL-KNOWN DEFENSE MECHANISMS IN CLOUD ENVIRONMENT," *Edpacs*, vol. 57, no. 5, pp. 1–26, 2018.

[16] Pajouh, H. H., Dastghaibyfard, G., & Hashemi, S, "Two-tier network anomaly detection model: a machine learning approach," Journal of Intelligent Information Systems, Springer, pp. 1-14, 2015.

ISSN 1737-930X