Modeling SIP Normal Traffic to Detect and Prevent SIP-VoIP Flooding Attacks Using Fuzzy Logic

Mahsa Hosseinpour
Department of Computer Engineering
Ferdowsi University of Mashhad
Mashhad, Iran
mahsa.hosseinpour@stu.um.ac.ir

Mohammad Hossein Yaghmaee Moghaddam
Department of Computer Engineering
Ferdowsi University of Mashhad
Mashhad, Iran
hyaghmae@um.ac.ir

Seyed Amin Hosseini Seno
Department of Computer Engineering
Ferdowsi University of Mashhad
Mashhad, Iran
hosseini@um.ac.ir

Hossein Khosravi Roshkhari
Department of Computer Engineering
Ferdowsi University of Mashhad
Mashhad, Iran
hos.khosravi.r@gmail.com

***Abstract*—As Voice over Internet Protocol (VoIP) or internet telephony became so popular, it has faced more security threats in comparison with traditional Public Switched Telephone Network (PSTN). Using IP-based infrastructures like public internet and signaling protocols such as Session Initiation Protocol (SIP), have been subjected this technology to various kinds of attacks. Denial of Service (DoS) attack, due to the flooding different kinds of SIP messages, is one of the most well-known type of these attacks. In this paper a new anomaly-based method for detecting and preventing different kinds of flooding attacks using SIP normal traffic modeling, is proposed. To reach this goal, SIP specifications are modeled and required parameters are extracted by the help of a FSM in order to use in fuzzy systems. Fuzzy systems results, put the proposed method in a predefined state. For prevention purposes, a filtering-based method using whitelist, is provided. Implementation results represent the fact that, the proposed method detects mentioned attacks more accurately in comparison with similar methods.**

Keywords—VoIP; Session Initiation Protocol (SIP); flooding attack; Finite State Machine (FSM); fuzzy logic

I. INTRODUCTION

Voice over Internet Protocol (VoIP) or internet telephony technology, allows users to make their calls using existing IP-based packet switched networks, like public internet, instead of circuit switched networks, like traditional Public Switched Telephone Network (PSTN). VoIP has advantages like flexibility in usage and cost effectiveness for both individuals and businesses [1]. However, these advantages comes with security drawbacks, since VoIP inherits security threats from public internet and makes it attractive target for attackers [2]. VoIP operates on various protocols including signaling protocols, media and transport protocols and gateway protocols [3]. While several signaling protocols exist, often Session Initiation Protocol (SIP) [4] is used for signaling purposes in VoIP. SIP is an application layer protocol which is used to initiate, modify and terminate multimedia sessions among the users. Considering being text based and simplicity of the SIP protocol, attackers can easily perform different kinds of attacks against this protocol. One of the well-known attacks, is flooding attack which malicious user is trying to overwhelm system's resources like CPU, memory or bandwidth, by creating numerous of requests. Flooding attack, is the simplest way to exploit SIP infrastructures vulnerabilities. Some of these attacks exploits the existing vulnerabilities in RFC specification of the SIP protocol and the others exploits different vendors SIP implementations. Accordingly, different kinds of this attack, can be categorized in four main group [5]: SIP Registration Flooding Attack, Authentication Flooding Attack, INVITE Flooding Attack and PING Flooding Attack.

In this paper, detecting and preventing of flooding attacks against SIP server, is considered. To reach this goal, an anomaly based method is proposed. In this method, two different phase are considered for training and testing the system. In training phase, by taking into account different times of day, a Finite State Machine (FSM) is constructed from normal calls of a secure VoIP network and important parameters like number of messages in each state transition and time differences between states, are extracted. By the help of the obtained parameters in training phase, a fuzzy logic approach is used for detecting attack in testing phase. After each time window, the system will be placed in a predefined state and according to that, a countermeasure scheme is used. Different kinds of flooding attacks can be detected by this approach. Also, a whitelist of legitimate users is provided to prevent from suspicious users to continue their calls in attack intervals.

The rest of the paper is organized as follows: section II includes an overview of the SIP protocol, different kinds of flooding attacks and presented flooding detection and prevention methods. In section III the proposed method including training and testing phases and prevention mechanism, are presented. In section IV the implementation test bed and experimental results will be discussed and section V concludes the paper.

978-1-5090-3586-1/16/$31.00 ©2016 IEEE

II. BACKGROUD AND RELATED WORKS

A. SIP Overview

Session Initiation Protocol (SIP) [4], is a signaling protocol for initiating, managing and terminating multimedia sessions which is standardized by the Internet Engineering Task Force (IETF). Like the HTTP, SIP is text-based with request and response structure which uses HTTP Digest Authentication method for authenticating users. SIP structure is consist of number of entities, including endpoints, proxy server, registrar, redirect server and location server.

A typical message flow for establishing a two-party call, is shown in Fig.1. Alice, User Agent client (UAC), sends an INVITE message to proxy server to initiate a call with callee, Bob, User Agent Server (UAS). Since caller and callee are placed in two different domains (Domain A and B), the proxy server in Domain A, relays the message to the Domain B's proxy server and sends back an "100 TRYING" message to the Alice. By receiving the INVITE message, the proxy server in Domain B forwards it to the Bob and sends back an "100 TRYING" to Domain A's proxy server. When Bob's phone starts ringing, an "180 RINGING" response is sent back to proxy server in Domain B. When Bob accepts the call, a "200 OK" response is sent by his phone which contains preferred parameters encoded within SDP. By receiving "200 OK", Alice's phone sends an "ACK" message to acknowledge the reception of the "200 OK" response. After this message exchange, the two endpoints can begin multimedia session using a media transport protocol which are agreed upon it, typically RTP, independently. At the end of call, one of the caller or callee, here Alice, sends a "BYE" request for terminating the call and the other responds with a "200 OK" message and call will be terminated [6].

B. SIP Flooding Attack

In a flooding attack, the attacker targets the victim's system resources like CPU, memory and bandwidth in order to make a DoS (Denial of Service) attack and so makes the system unavailable. As previously mentioned, there are different ways to launch this kind of attacks. In this paper we focus on the INVITE flooding attack which in this case malicious user sends several INVITE messages to the victim. We have categorized INVITE flooding attack into two groups, including:

- INVITE single/multiple source flooding attack
- Session flooding attack

In the first case, the attacker creates numerous INVITE messages and sends them to a proxy server in order to consumes its resources and causes a DoS attack. By receiving each new INVITE message, the proxy server will allocate memory for that message until a final response is received or time-out is triggered. Therefore, lots of this messages consumes server's memory and leads to DoS. Obviously, by using INVITE messages from multiple sources, memory and CPU consuming speeds up [7]. The second case, means session flooding attack, is like TCP/SYN flooding attack which consumes server's memory. When the attacker sends an INVITE message to the server, the proxy server allocates memory to that message in order to manage a new session and sends it to the callee. After accepting the call by callee and receiving 200 OK, the server forwards 200 OK message to caller and waits for receiving acknowledgment. Since the INVITE message is spoofed, no one will acknowledge the request. The attacker can exhaust the server's memory and make DoS attack by sending a large number of this INVITE messages [7, 8].

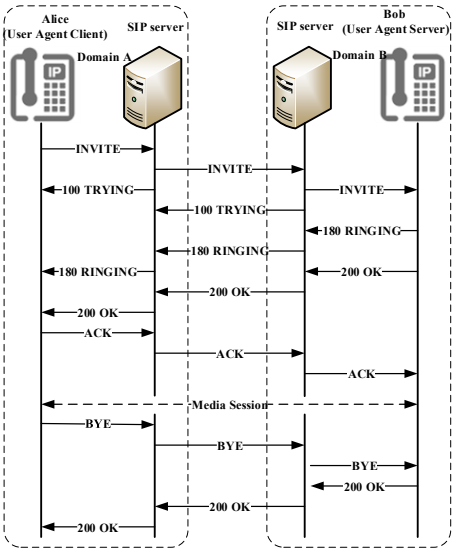

Fig. 1. Message flow for establishing connection in SIP

C. Related Works

Several security solutions have been proposed to detect and prevent SIP-VoIP related flooding attacks. Using threshold based methods is one of the most common strategies in order to detect flooding attacks. When the number of messages exceeds from a specified threshold, the attack could be detected. But accurate determination of this threshold is so important to minimize false positives and false negatives.

Considering various features of the different SIP messages and congestion of the network, in [9] upper bound of the possible number of messages for detecting attack, is presented. The authors of [10] are computed the threshold value using analyzing normal SIP traffic statistically when there is no congestion. In [11] an Adaptive Threshold (AT) is proposed for threshold value which is computed dynamically based on the history of previous traffic patterns. Detecting flooding attack using Hellinger Distance (HD) by measuring variability between two probability distributions is proposed in [12-14]. Computing entropy, is another method for detecting flooding attacks which is proposed in [15-18]. The authors of [19] introduced a new parameter named Critical Number (CN) to informed proxy server about allowed number of each kind of message. In [8], an anomaly based method by the help of SIP specification in RFC 3261, named SIPAD, is proposed.

For preventing SIP flooding attacks, the most common methods, are filtering based ones by the help of blacklists and whitelists. As an example, in [2] a blacklist is created using an Intrusion Detection System (IDS) which prevents attackers to continue their calls. Using whitelists is another filtering based method that only legitimate users can establish their calls. Considering users which registered successfully [20] or making a SIP session normally [21], can be used to create these lists. Since huge amounts of user's session information must be keep in whitelists, fast data structures with low

memory requirements, like bloom filters, are proposed in [22, 23].

III. The Proposed Method

In this paper, a new method for detecting and preventing SIP flooding attacks by the help of SIP normal traffic modeling, is proposed. To reach this goal, two phases are considered: 1) training phase for creating a FSM and extracting required parameters from normal SIP calls and 2) testing phase which uses fuzzy systems and also a new equation and whitelist method in order to detect and prevent flooding attack, respectively. in this section these two phases will be described in details.

A. Training Phase: Modeling Normal SIP Traffic

In this phase we have used normal and secure calls of a VoIP provider in order to create a Finite State Machine (FSM) from them. Each state in this FSM shows a kind of SIP messages and each edge of FSM, means inputs of it, represents transition between states and flow of call. For creating this FSM, a parser module is provided. With the arrival of each kind of SIP messages to the parser module, important parameters like FROM, TO, Call-ID and etc. fields are extracted. Since network traffic conditions are changed in different times of a day, extracting parameters is done in various time intervals. Each session in a SIP-VoIP call is specified using <From-tag, Call-ID> two tuples. Therefor different sessions could be distinguished by the parser module and required parameters, means number of messages in each state transition and time differences between them, could be extracted. In Fig.2, a sample of created FSM and call flow between two users is shown, which one call is considered with authentication and the other is without authentication.

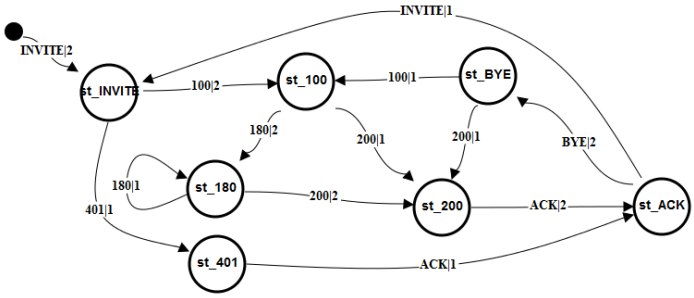

Fig. 2. Sample of call flow between two users

In this figure, the first number of the each edge represents the kind of message that leads to state transition and the second one shows number of that kind of message.

B. Testing Phase: Detecting And Preventing Flooding Attack

As mentioned before, flooding INVITE messages from single and multiple sources and session flooding attack, are considered in this paper. The entire proposed detection algorithm is shown in Fig.3. Like the training phase, in order to use obtained results in this phase, different time intervals in a day are considered. Also for detecting attacks more accurately, smaller time windows are used in this phase. Three different states, means NORMAL, ALARM and ATTACK states, are assumed which by placing in each one a different prevention mechanism will be used.

At the end of each time window, based on the observed messages and also extracted parameters, means number of messages in each transition and average of time differences between each two states, corresponding fuzzy systems starting to execution. Results of the fuzzy systems execution, place proposed system in one of the predefined states. In the presented method, sensitivity of the system to this kind of attack can be determined by the system administrator.

```
<variables>
Message_Cnt =Avg_Time = 0 at the start of the time window;
Defining Normal_Thr, Attack_Thr and Session_Thr;

<algorithm>
Extracting received packets parameters (kind and time of
message)
While (!end of time window){
    Message_cnt++;
    Computing Avg_Time;
    If session ended and call terminates successfully? {
         Updating whitelist
    }
}
Running fuzzy systems; //result
if ( result < Normal_Thr )
    State=NORMAL;
elseif (result >= Normal_Thr and result < Attack_Thr)
    State=ALERT;
elseif (result > Attack_Thr)
    State=ATTACK;
Session_Distance = #INVITE messages / #ACK messages;
If (Session_Distance> Session_Thr)
    State=ATTACK;
Prevention mechanism;
```

Fig.3 entire attack detection algorithm

In this paper, Mamdani approach is used as fuzzy derivation method. As previously mentioned, there are number of fuzzy systems in the proposed algorithm that each one has two inputs and one output variables. Inputs of each fuzzy system are number of messages in each transition and average of time differences between each two states and its output is attack severity. For example, in Fig.4, 5 and 6, membership functions for inputs and output of INVITE to 401 (Un-Authorized) messages are shown.

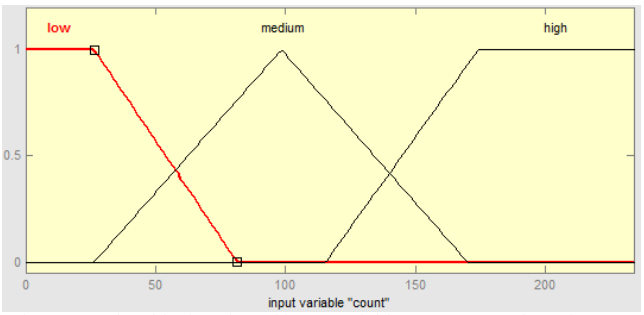

Fig. 4. Membership function of input variable "count" (number of message)

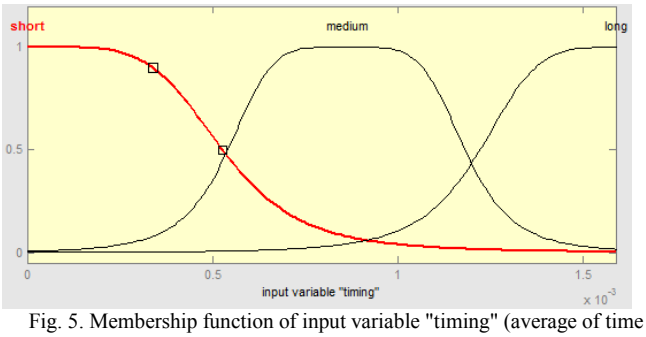

Fig. 5. Membership function of input variable "timing" (average of time difference between states)

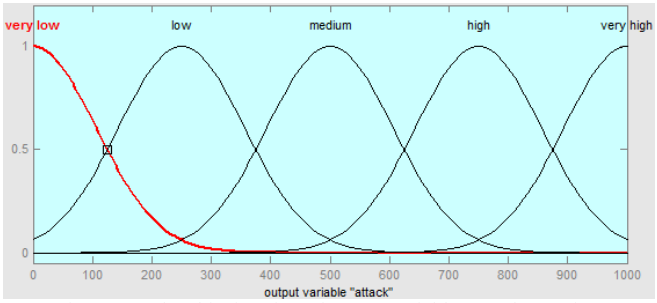

Fig. 6. Membership function of output variable "attack severity"

After specifying the membership functions, fuzzy rule base must be designed. The fuzzy rule base, is composed of some If-Then rules for decision making. In the following table, rules which are used in the proposed algorithm, are presented.

TABLE I. FUZZY RULE BASE

Count/timing	*short*	*medium*	*long*
low	Low (0.5)	very low (0.6)	Low (0.5)
medium	High (0.4)	Medium (0.1)	High (0.4)
high	very high (0.3)	High (0.4)	very high (0.3)

As mentioned before, degree of system's sensitivity can be specified by the system administrator. For this purpose, ten degrees of sensitivity is assumed. Accordingly, weighted fuzzy rules are considered in order to sensitivity degree be included. In Table I, numbers in parenthesis, show considered weights when degree of sensitivity is equal to 5.

When the attacker wants to flood the server with numerous open sessions and also low rate attacks, number of open sessions must be controlled. In normal conditions, number of INVITE messages and number of corresponding ACK messages at initiating sessions, are approximately equal. Considering this fact, in (1), an equation for controlling this proportion, named session distance, is proposed:

$$session\_distance = \frac{\#\,INVITE\ messages}{\#\,ACK\ messages} \qquad (1)$$

If the attacker uses spoofed INVITE messages, when the server sends back corresponding 200 OK response, since there is no real UAC in order to acknowledge the reception of 200 OK message, no ACK message will be sent. Also it is possible that in order to hold the number of INVITE messages lower than specified threshold in each time window, the attacker uses low rate flooding. In this kind of attack, since number of open session increases and server's memory uses, 200 OK message can't be created by the server and doesn't reach to UAC and therefore corresponding ACK message will not send to server from UAC. With regard to the mentioned details, in normal cases this proportion will be approximately equal to 1 or a little more than 1 (because of resending of INVITE messages in order to cope with packet loss in UDP) but in attack conditions, the value of this relation increases.

C. Attack Prevention Mechanism

When system placed in ATTACK state, in order to prevent from malicious users calls to reach the server, a whitelist mechanism is used. Users which terminates their calls successfully are added to this whitelist. In order to distinguish such users, Call Details Records (CDR) of calls are used.

In each time window, if the system is in attack state, only those users in whitelist can continue their calls and the other calls will be filtered and not permitted to reach to the server. In alarm state, the callers are asked to answer a captcha, it is clear that users with true answers can continue their calls and finally in the normal state, callers can continue their calls as usual.

IV. IMPLEMENTATION RESULTS

A. The Test Bed

In order to test the presented system, Spirent Abacus 5000 device is used to generate SIP normal traffic. This device could generate traffic with different transmission rates and distributions. With the help of Spirent, it is possible to define large number of user agents to call each other in a normal way. Also, Asterisk software is used to implement proxy server.

For generating attack traffic, SIPp tool [24] is used which is a free open source for traffic generating using SIP signaling protocol. With this tool it is possible to define different SIP traffic generation scenarios with different rates by using a certain type of SIP signaling packet or combination of them. When the network begins to work, the required parameters, as mentioned before, are collected using parser module in a database. These parameters are used for session distinction and attack detection. In order to control the database's tables volume in testing phase, stored data are periodically archived. The intended test bed for testing phase is shown in Fig.7.

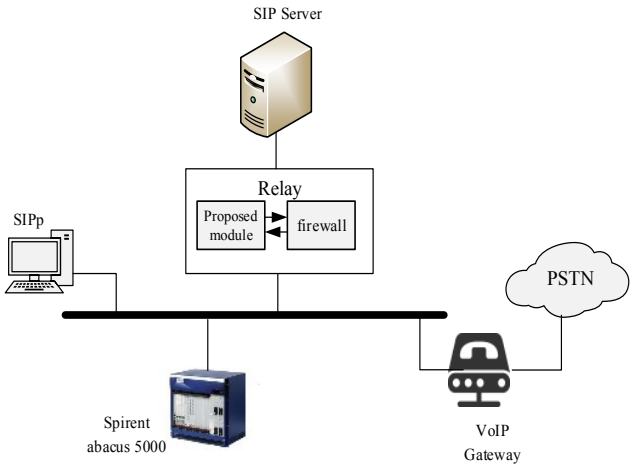

Fig. 7. Testing phase test bed

B. Experimental Results

We have done several experiments to evaluate and analyze the proposed system against flooding attacks. In all of them the generated Spirent calls as normal traffic, are mixed with SIPp attack ones with different rates. As mentioned before, in order to detecting attacks more accurately, time windows are considered. Using different rates of attacks, false positives and false negatives in detecting attacks are calculated. As can be seen in Fig.8, the best size for time window for detecting attacks is equal to 8 seconds since we have minimum false positives and false negatives.

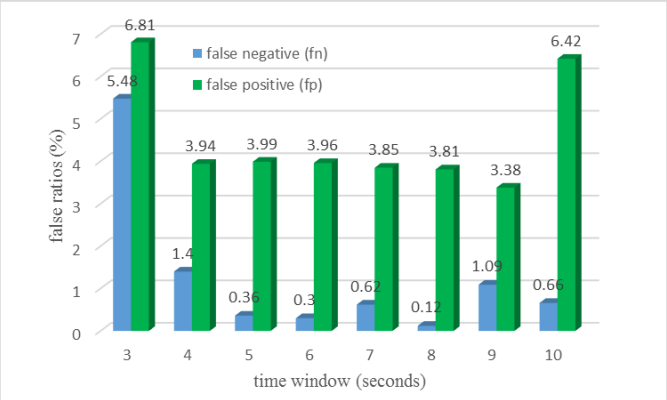

Fig.8. False ratios in different time window sizes

In this paper, we compare our proposed method with one in [8], which is named SIPAD. The authors of SIPAD, adopted modified state transition models from RFC 3261 in order to detect SIP anomalies. Accordingly, to detect flooding attack, different threshold values are defined for each state. For example, in INVITE server transition model, these states are Proceeding, Completed, Confirmed and Terminated which in each one only some kinds of messages and transition between messages, are allowed. Since in SIPAD, a number of messages are allowed in each state, in comparison with our method which there can be only one kind of SIP messages in each state transition, detection probability will be increased significantly. In Fig.9, the difference between number of messages in different states in each method is shown.

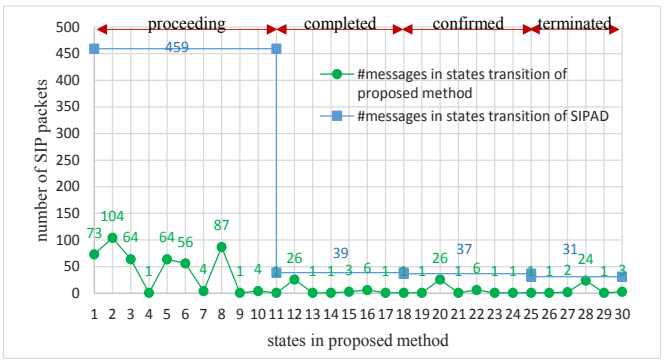

Fig.9 number of messages in different states in each method

The numbers on the horizontal axis, shows each state in our proposed method. For example, number 3 shows a 100 TRYING state when we reach to it from INVITE state.

Considering aforementioned notes, we computed detection probability for comparison purposes. We use this parameter as the percentage of the successful identified attack samples over the total launched attacks. In order to achieve these values, a number of SIPp scenarios with different calls rate is used. In Fig.10, detection probability in different rates of attack is depicted. As can be seen, detection percentage at the end of each time window in our method is significantly higher than SIPAD method.

In order to evaluate proposed method when there are numerous open sessions in system (session flooding) due to spoofed INVITE messages, we used another SIPp scenario which creates number of open sessions with different call rates. In this case we did several experiments for each rate to compute detection probability in the experiment period. The results of this experiment is shown in Fig.11. Since in SIPAD method only 200 OK messages than INVITE ones are considered to detects open sessions, attack situations which use spoofed INVITEs, can't be detected carefully as the attacker will not send ACK messages for acknowledgment purposes.

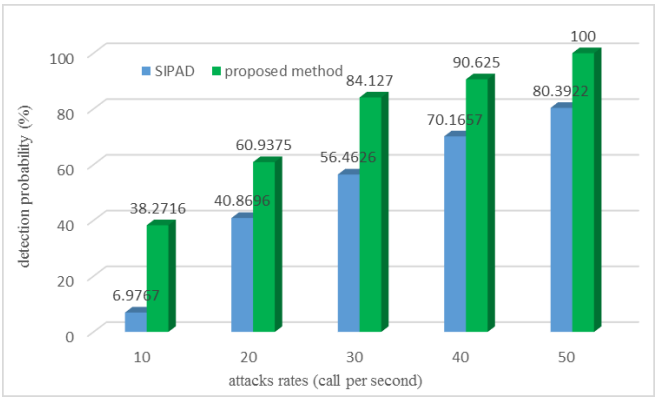

Fig.10 detection probability considering different rate of attacks

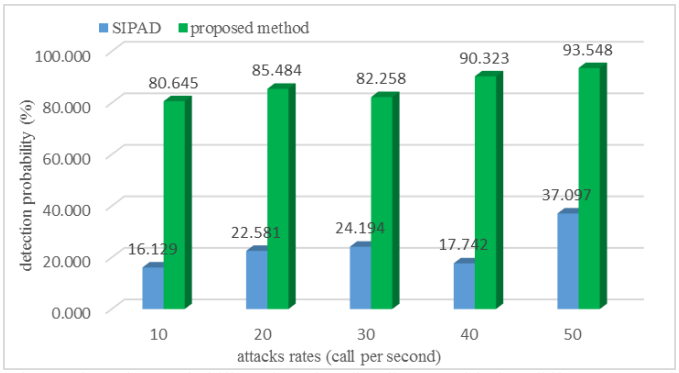

Fig. 11 detection probability of session flooding considering different rates of attacks

In Fig.12, we can see the number of different SIP packets when there is flooding attack in the VoIP network. In this figure, the rate of attack is considered equal to 50 cps (call per second). By using our proposed method to detect and prevent this kind of attack, number of messages which reach to server after using prevention mechanism, which allows only normal users to continue their calls in attack situations, is shown in Fig.13.

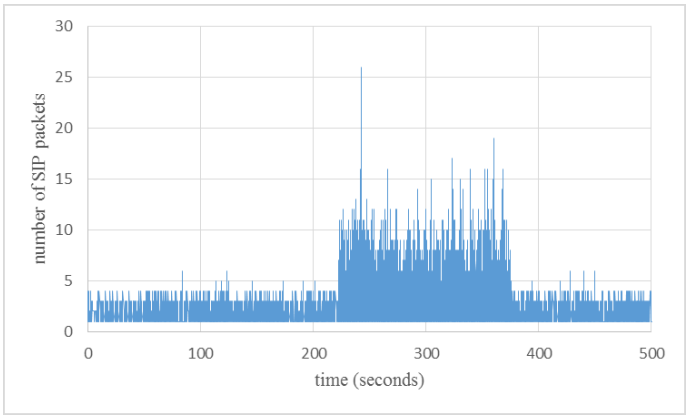

Fig. 12 number of different SIP packets in normal and attack interval

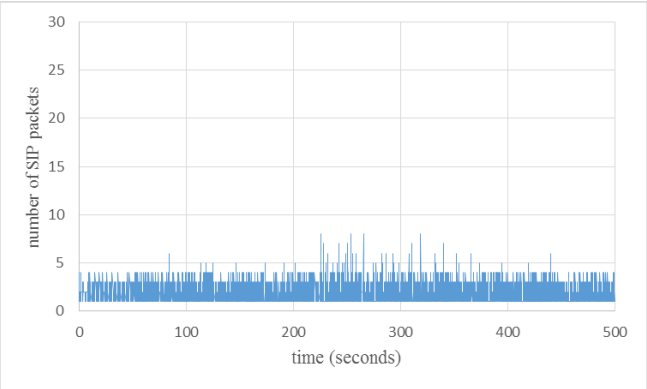

Fig. 13 number of different SIP packets which reach to server after prevention

V. CONCLUSION

SIP-based multimedia systems like VoIP services, are becoming so popular for communication purposes. But these services are exposed to some kinds of attacks such as DoS attacks due to flooding different kinds of SIP messages. In this paper, a new method is presented for detecting and preventing this kind of attacks. To reach this goal, a FSM (Finite State Machine) is used for modeling SIP normal traffic of a VoIP provider. Two type of flooding attacks, by sending numerous INVITE messages with different rates and creating numerous open sessions for consuming server resources, are considered. Using extracted parameters in this phase in different times of day, fuzzy systems are used in order to detects flooding attack due to numerous SIP messages. Furthermore, a new equation is proposed to detect open sessions in test period. At this point and after detecting attack, the system will be placed in a predefined state, means NORMAL, ALARM and ATTACK. In order to prevent from attacker's messages to reach to server in ATTACK state, a prevention mechanism using whitelist based methods, is used. Also, in ALARM state, users will be asked to answer a played captcha in order to continue their calls. This method is completely implemented and obtained results show that in comparison with similar methods, this method is more accurate and have less false positives and false negative alarms.

REFERENCES

[1] D. Butcher, X. Li, and J. Guo, "Security challenge and defense in VoIP infrastructures," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 37, pp. 1152-1162, 2007.

[2] F. Huici, S. Niccolini, and N. d'Heureuse, "Protecting SIP against very large flooding DoS attacks," in Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE, 2009, pp. 1-6.

[3] V. Srihari, P. Kalpana, and R. Anitha, "Security aspects of SIP based VoIP networks: A survey," in Current Trends in Engineering and Technology (ICCTET), 2014 2nd International Conference on, 2014, pp. 143-150.

[4] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, et al., "SIP: session initiation protocol," RFC 3261, Internet Engineering Task Force2002.

[5] A. Kumar and S. Tilagam, "A novel approach for evaluating and detecting low rate SIP flooding attack," International Journal of Computer Applications, vol. 26, pp. 31-36, 2011.

[6] A. D. Keromytis, "A comprehensive survey of voice over IP security research," Communications Surveys & Tutorials, IEEE, vol. 14, pp. 514-537, 2012.

[7] D. Geneiatakis, N. Vrakas, and C. Lambrinoudakis, "Utilizing bloom filters for detecting flooding attacks against SIP based services," computers & security, vol. 28, pp. 578-591, 2009.

[8] D. Seo, H. Lee, and E. Nuwere, "SIPAD: SIP–VoIP anomaly detection using a stateful rule tree," Computer Communications, vol. 36, pp. 562-574, 2013.

[9] J.-T. Ryu, B.-H. Roh, and K.-Y. Ryu, "Detection of SIP Flooding Attacks based on the Upper Bound of the Possible Number of SIP Messages," KSII Transactions on Internet & Information Systems, vol. 3, 2009.

[10] Y. Rebahi, M. Sher, and T. Magedanz, "Detecting flooding attacks against IP Multimedia Subsystem (IMS) networks," in Computer Systems and Applications, 2008. AICCSA 2008. IEEE/ACS International Conference on, 2008, pp. 848-851.

[11] M. A. Akbar, Z. Tariq, and M. Farooq, "A comparative study of anomaly detection algorithms for detection of SIP flooding in IMS," in Internet Multimedia Services Architecture and Applications, 2008. IMSAA 2008. 2nd International Conference on, 2008, pp. 1-6.

[12] J. Tang, Y. Cheng, Y. Hao, and W. Song, "Sip flooding attack detection with a multi-dimensional sketch design," Dependable and Secure Computing, IEEE Transactions on, vol. 11, pp. 582-595, 2014.

[13] N. Chaisamran, T. Okuda, and S. Yamaguchi, "Using a Trust Model to Reduce False Positives of SIP Flooding Attack Detection in IMS," in Computer Software and Applications Conference Workshops (COMPSACW), 2013 IEEE 37th Annual, 2013, pp. 254-259.

[14] H. Sengar, H. Wang, D. Wijesekera, and S. Jajodia, "Detecting VoIP floods using the Hellinger distance," Parallel and Distributed Systems, IEEE Transactions on, vol. 19, pp. 794-805, 2008.

[15] Z. Tsiatsikas, D. Geneiatakis, G. Kambourakis, and A. D. Keromytis, "A Privacy-Preserving Entropy-Driven Framework for Tracing DoS Attacks in VoIP," in Availability, Reliability and Security (ARES), 2013 Eighth International Conference on, 2013, pp. 224-229.

[16] Z. Tsiatsikas, D. Geneiatakis, G. Kambourakis, and A. D. Keromytis, "An efficient and easily deployable method for dealing with DoS in SIP services," Computer Communications, vol. 57, pp. 50-63, 2015.

[17] C. Hui and H. Chao, "Monitoring SIP Traffic Using Statistical Approaches," in 2nd International Conference on Electronic & Mechanical Engineering and Information Technology (EMEIT), 2012, pp. 1585-1588.

[18] Q. Qia and Z. Wang, "A new attack detection in large scale network based on entropy," Journal of networks, vol. 7, pp. 863-868, 2012.

[19] I. Hussain and F. Nait-Abdesselam, "Strategy based proxy to secure user agent from flooding attack in SIP," in Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International, 2011, pp. 430-435.

[20] E. Y. Chen and M. Itoh, "A whitelist approach to protect SIP servers from flooding attacks," in Communications Quality and Reliability (CQR), 2010 IEEE International Workshop Technical Committee on, 2010, pp. 1-6.

[21] C. V. Zhou, C. Leckie, and K. Ramamohanarao, "Protecting SIP server from CPU-based DoS attacks using history-based IP filtering," Communications Letters, IEEE, vol. 13, pp. 800-802, 2009.

[22] B.-h. Roh, J. W. Kim, K.-Y. Ryu, and J.-T. Ryu, "A whitelist-based countermeasure scheme using a Bloom filter against SIP flooding attacks," Computers & Security, vol. 37, pp. 46-61, 2013.

[23] K. Ryu, J. Kim, and B.-h. Roh, "Whitelist-based SIP flooding attack detection using a Bloom filter," Proc. ICTA, vol. 2011, 2011.

[24] http://sipp.sourceforge.net/.