A Lightweight Countermeasure to Cope with Flooding Attacks Against Session Initiation Protocol

Intesab Hussain[∓], Soufiene Djahel[‡], Dimitris Geneiatakis[±], and Farid Naït-Abdesselam[∓]
[∓] LIPADE, University of Paris Descartes, France
[‡]Lero, UCD School of Computer Science and Informatics, Ireland
[±] Digital Systems, University of Piraeus, Greece
{intesab.hussain, naf}@parisdescartes.fr, soufiene.djahel@ucd.ie, dgen@unipi.gr

***Abstract*—Session Initiation Protocol (SIP) is a widely used protocol for voice and video communication in Internet architecture. Due to its open nature and the lack of robust security mechanisms, SIP is vulnerable to several attacks similar to those existing in Internet infrastructure, such as the flooding attack. An attacker can use any SIP request to launch a flooding attack, leading to severe consequences at either client or server side SIP elements or both of them. In this context, end user's devices are considered more vulnerable to flooding attacks due to their limited capabilities. In this paper, we focus on INVITE flooding attack for which we propose a simple and robust detection scheme. This scheme prevents an attacker from launching an INVITE flood through a transition state table used by the proxy to analyse the incoming INVITE requests and exclude the suspicious ones. Our scheme requires also that the end-user keeps track of the time and IP addresses of each incoming request. Furthermore, we modify the header of the REGISTER request by adding a new field named Critical number which holds the value of maximum number of users or callers that could easily be handled by the end user. Unlike the existing solutions, our scheme does not require any special detection device or firewall at the SIP server. The proposed mechanism has been implemented in SIP Express Router (SER) and the obtained results have confirmed its effectiveness.**

***Keywords* – Voice over Internet Protocol (VoIP), Session Initiation Protocol (SIP), Denial of Service, INVITE Flooding Attack, Security.**

I. Introduction

Session Initiation Protocol (SIP) is an application layer protocol which creates, modifies and terminates multimedia sessions [3]. SIP can be combined with other protocols such as RTP (Real Time Transport Protocol) [1] and SDP (Session Description Protocol) [2] to build complete multimedia architecture, for example.

Using the SIP address, the user finds the current location of the destination from a registrar which extracts it from the location server. The user needs to register with the combination of SIP addresses with its current IP location. Registration in SIP [3] is a way to associate the SIP URI (Uniform Resource Indicator) with the machine into which the user is currently logged on. It helps to find the current location of the callee through the proxy. The proxy server queries the registrar that contains the location server which is a database holding the users' records (i.e. their SIP URIs and their current IP addresses).

There are different types of attacks targeting VoIP protocols, such as DoS attacks, call hijacking, toll fraud, SPam over Internet Telephony (SPIT) and vishing [4]. DoS attack is the most devastating attack amongst all. A DoS attack includes signaling attacks, malformed packets and flooding attacks. Flooding is one of the SIP application attacks which exhausts the memory, CPU and bandwidth of the victim user. Due to its harmful impact on SIP performance, we will mainly focus, in this paper, on designing a robust solution to deal with this attack.

SIP security has attracted a lot of attention from the research community in order to cope with the aforementioned attacks. As a result, numerous solutions have been proposed to deal with flooding attack in SIP, such as [5], [6], [7], [8], [12] and [9]. To complement these efforts and overcome their limitations (e.g., cost, implementation complications and extensive changes in servers), we propose in this paper a lightweight countermeasure to secure SIP against this harmful attack.

In this paper, We design and implement in a test-bed a simple, practical and robust solution to secure the end user against SIP request flooding attack. This work is an extension and improvement of our previous work [13] based on the strategic model. In this work, we use the address of the incoming requests, destination address and arrival time of a particular request to monitor the number of transactions towards a particular end user. In SIP, each user device can handle a number of incoming requests according to its processing capacity, memory and bandwidth. According to SIP specifications [3], the SIP headers are extendable, so that we have added a new metric named "Critical Number" in REGISTER header. The end user uses this header to inform the server side about the maximum number of requests that it is able to handle. This information is defined in the "Critical Number" (CN) field which should be set during the device registration phase at the proxy server. To prevent overwhelming the end user, the proxy ensures that the number of requests forwarded to this end user doesn't exceed the critical number value.

The rest of the paper is organized as follows. In section II, we give a brief description of the flooding attack. The next section summaries the literature followed by a detailed description of the proposed solution, in section IV. The evaluation results of our solution are presented and discussed in section V. Finally, we conclude the paper in section VI.

II. Request flooding Attacks in SIP

INVITE Flooding attack is one of the most devastating attacks targeting SIP. In this attack, the attacker generates several INVITE requests to exhaust the server and callee resources. Both of SIP proxy and end user are vulnerable to flooding attack. The proxy must be connected with the callee for several minutes and due to that it is easy to keep it busy by sending a large number of INVITE requests, without waiting for the corresponding acknowledgment. Figure 1 shows the SIP call establishment. We can distinguish several scenarios of flooding attack in SIP, as described in the following.

The attacker could be a legal SIP user, so it has an account in SIP server and consequently it is treated as a legitimate SIP user. However, it puts the callee in busy situation by the flood of INVITE requests. On the other hand, the attacker could be an outsider if no authentication mechanism is in place. This type of attack is not easy to detect due to the insecure environment. Another possibility of INVITE flooding could be the Distributed Denial of Service (DDoS)

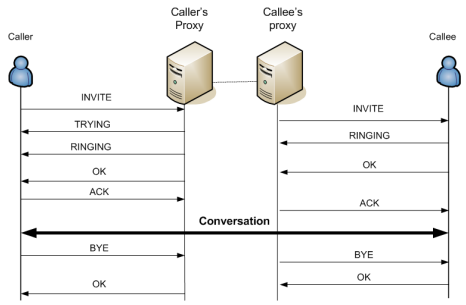

Figure 1: An example of SIP session with multiple proxies

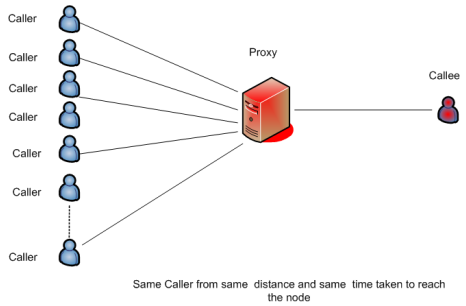

Figure 2: Illustration of Flooding attack on UAS

in which many attackers launch the attack from different locations in a distributed fashion. Every soft-phone has different number of parallel calls receiving capability. Therefore, coping with such attack is a real challenge for the research community.

Figure 2 depicts the flooding of INVITE request. Here, the caller sends many INVITE requests to flood the proxy or end user. Until and unless the proxy is connected to the desired callee, it must keep connection with the caller. On the other hand, despite that the caller receives TRYING message from the proxy it ignores it and sends a new INVITE request without waiting for further signals.

As defined in SIP draft, the proxy has to keep connection at least for some minutes when it receives an INVITE request from any caller. Therefore, the caller exploits this vulnerability and floods INVITE requests to exhaust both of the end user and proxy, which paralyzes the system and prevents the target end user from responding to other callers as well as to the proxy.

III. Related work

In this section, we present the most significant contributions that has been proposed to cope with the flooding attack in SIP.

In [5], the authors have proposed a special architecture to implement a scalable prevention scheme to defend against DoS attacks in SIP. Their scheme is based on an SIP-aware firewall design composed of two filters, a dynamic pinhole filter for media traffic and SIP-specific filter for signaling traffic. Moreover, this scheme implements different defense mechanisms, such as Return Routability Filter, Rate-limiting Filters and SIP transition state validation. Although this scheme can offer some levels of robustness against DoS attacks it still have some limitations, for example a firewall does not protect against authorized internal users who behave maliciously. It cannot also protect from the internal flaws of the network. Additionally, the rules defined by this scheme to detect DoS attacks do not cover all possible attack scenarios.

To avoid the aforementioned limitations of this scheme, [6] has proposed a secure architecture, while [10] has applied Hellinger Distance and Sketch technique together to avoid flooding attacks.

The authors of [7] have used an anomaly detection device which is placed before the proxy to detect the INVITE flooding attack. These anomaly detection devices are designed based on Finite State Machine (FSM) and each of them has the capability to detect flooding through different threshold parameters. This solution requires to put a detection device at every SIP server, which makes it a costly solution that doesn't suit our needs and our network constraints.

In [12], two types of DoS attacks have been dealt with. The first attack type is that launched by the external users which is usually detected through firewall like solutions. The second attack type is known as network reconfiguration attack. This work is focused on monitoring the users for generating false messages, broken sessions, and abrupt increase in the number of transactions on stateful proxies. It emphasizes on server design in case of stateless proxy, however there is no specific mechanism proposed for the prevention of INVITE flooding except monitoring the traffic.

The work described in [11] suggests to put a Snort at the entry point of SIP traffic. This solution is based on Snort IDS (Intrusion Detection System) which inspects the packet load to protect it from the multiple signaling requests through specific detection rules in IDS. This solution is useful for DNS (Domain Name Service) blocking, malicious messages and flooding in SIP, however there is no solution proposed to deal with flooding attack targeting the end user.

Compared to the solutions presented in this section, the solution that we will present throughout the rest of the paper is practical and cost effective. It does not need any add-on detection device or any other complex technique, but it just require a slight modification to SIP REGISTER header along with a transition state table to be added at the proxy side, as described in next section.

IV. The proposed scheme

Most of the existing solutions have focused on adding firewalls, anomaly detection devices and intrusion detection systems at the SIP proxy to defend against INVITE flooding attack. In contrast, our proposed scheme is cheaper in cost and practical. It consists in adding a table with three metrics at the proxy and a new field dubbed "Critical Number" in the registrar of SIP proxy during user's device registration phase. Our solution could generally applied on any SIP request method, however, in this paper, we focus on INVITE request method.

Our main concern, in this work, is to reduce as much as possible the callee's extra-load incurred by the unusual SIP flooding request attack. Usually, a caller may receive a large number of calls in some specific circumstances like emergency, anniversaries, happy and sad moments. Therefore, it is difficult to distinguish those special occasions from a flooding attack, but at least we will prevent a huge number of SIP request to be forwarded towards a particular callee, because this behavior is considered as an attack that leads to callee's resources depletion.

Table I: Different callers sending INVITE requests at different times

Method	Source IP	Destination IP	Arrival time
INVITE	206.219.77.11	195.228.240.177	16:21:45-50
INVITE	64.95.79.4	134.96.68.36	14:53:12-17
INVITE	216.34.121.18	129.15.12.256	5:13:52-57
INVITE	—	—	—
INVITE	202.212.5.47	129.105.12.256	9:26:12-17

Table II: The same caller is sending many INVITE requests at the same time

Method	Source IP	Destination IP	Arrival time
INVITE	202.212.5.47	129.105.12.256	10:42:42-47
INVITE	202.212.5.47	129.105.12.256	10:42:42-47
INVITE	202.212.5.47	129.105.12.256	10:42:42-47
INVITE	—	—	—
INVITE	202.212.5.47	129.105.12.256	10:42:42-47

A. Transition state table for the SIP proxy

We propose to add a transition state table in the SIP proxy to detect the malicious behavior of SIP users. As it is well known, there are two types of SIP proxies, stateful and stateless. Stateful proxy saves the history of all connections, transitions and traffic in the buffer of the server for future use, whereas the stateless proxy is used only as an intermediate element that forwards the request from the caller to the callee. In stateless proxy, the information is kept in the buffer at least during the SIP transaction. Unlike stateful proxy, it cannot save the transactions for future use. In our solution, we add a transition state table that contains three entries as follows; (i) command, (ii) IP address, and (iii) arrival time. The command shows the type of request sent by the caller. As we are trying to protect both of the proxy and the end user from the flood of INVITE requests, hence the proxy adds an entry in the table upon reception of INVITE request only. We then use the source and destination IP addresses entries to keep track of the caller and callee, respectively, as shown in Table III. The Arrival time entry records the time at which the proxy has received the INVITE request from the caller.

We extract the source and destination IP addresses as well as the arrival time as metrics of the transition state table. later, we use these metrics for detecting flooding attacks. The time unit used to detect the flooding attack is micro-seconds. Normal SIP call establishment is described in Table I where different callers are calling the same destination but at different times and origins.

Table III: Different callers sending INVITE requests at the same time (DDoS)

Method	Source IP	Destination IP	Arrival time
INVITE	208.219.77.11	129.105.12.256	09:26:22-27
INVITE	195.228.240.177	129.15.12.256	09:26:22-27
INVITE	64.95.79.40	129.105.12.256	09:26:22-27
INVITE	—	—	—.
INVITE	cyan 202.212.5.47	129.105.12.256	09:26:22-27

B. The Critical Number (CN)

We have introduced a new header dubbed "Critical Number", as it is allowed to extend the SIP request by adding new headers to support new services (see RFC 3261 [3]). This new header is used to advertise the end users' capabilities, in terms of requests processing per second, to the server side.

Table IV: Example of an extended SIP Registrar: it contains the new field named "Critical Number"

SIP URI	IP Address	Critical Number
sip:abc@misc.com	134.206.11.61	10
sip:jkl@lifl.com	191.13.121.11	15
sip:xyz@cyber.com	139.12.131.61	12

We know that proxy searches current location of the destination from the registrar, which in turn extracts this information from the location server. The user needs to register the combination of SIP addresses with its current IP location. In our detection and prevention mechanism, we should know the critical number of each end user. So, if the proxy is aware of all critical numbers values, it can easily block the flooding attack according to the capacity of the callee. To do so, we propose that the end user sends its critical number with its SIP URI and current IP Address during the registration phase. Table IV depicts the three metrics; SIP URI, IP Address and the critical number of the end user.

To avoid legitimate calls blocking according to the device capacity, we propose a dynamic threshold that differs according to the capability of different devices. For example the same user can use different devices, e.g., Mobile, Soft-phone, traditional phone and conferencing tool.

Figure 3 illustrates the registration process of a device according to our scheme, where the user sends its CN value with its registration details. Figure 4 presents the call establishment of SIP after applying our scheme. Firstly, the caller will send the INVITE request through the proxy. Then, this latter will examine the Arrival time, IP Address and the total number of requests (TN) for a each user, then it compares the TN value for a specific user with its Critical number. To this end, the proxy demands the CN from the Registrar and if the TN is less than the CN value then it transfers the INVITE request to the callee. Otherwise, the proxy blocks the other requests until any of the previous requests has been terminated. Notice that the proxy treats the awaited requests in a priority of first-come-first-serve basis.

Algorithm 1 illustrates the reaction of the proxy when the number of received requests towards a given callee has reached its CN, in case of receiving multiples requests from a single caller and when these requests are generated by different senders. Upon reception of a new call towards a given end callee, the proxy checks first whether the CN has been reached or not. If so, it then generates a waiting queue for this callee in which the new incoming calls will be stored. We assume that the proxy is able to receive many requests at a specific time frame and maintain the queue, which can produce better security. Whenever any of the ongoing calls terminated, the proxy release a new call from waiting queue. In the second and third cases, the IP addresses of the request's sources as well as their arrival times are the main metrics used by our scheme to distinguish the normal call from an attack.

C. Dealing with single source & DDoS

In single source flooding (as depicted in Table II), the table holds the same value of IP address for all requests as well as the same arrival time for every single request. In case of receiving multiple requests from the same IP address during a particular time window, we can hold the flood of INVITE through a special detection mechanism that senses the high number of INVITE floods based on the information hold by the transition state table. When the detection system detects a high number of incoming INVITE requests from the same origin at the same arrival time, it blocks this user. Note

Algorithm 1 Queue generation at proxy in case of single and DDoS flooding attack

Case 1: when the Critical number of an end user has been reached;
if (Received BYE == Id_Callee) **then**
  if (Request table) is not empty **then**
    Forward new request;
    N - -;
  end if
end if
Case 2: of multiple requests from a single end user;
Extract IP addresses of the callers from Transition state table;
if (IP Addresses are the same for all requests) **then**
  Check the Arrival Time for those IP Addresses;
  if (Arrival time) is the same for all IP Addresses **then**
    Block those requests;
  end if
end if
Case 3: of DDoS attack;
Extract IP addresses of the callers from Transition state table;
if (IP Addresses are different) **then**
  Check the Arrival Time for those IP Addresses;
  if (Arrival time) is same for all IP Addresses **then**
    Block those requests;
  end if
end if
N: is the total number of requests stored at the Transition state table;

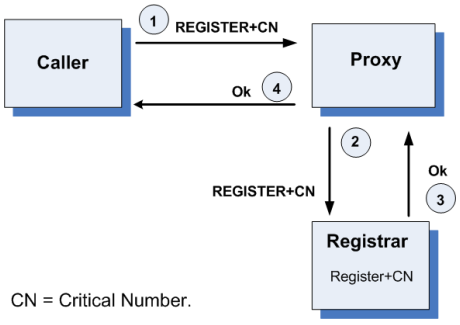

Figure 3: User Registration phase in our scheme

that the proxy does not take any actions against a suspected user until the INVITE flood level reaches the CN. Considering the user device, we can have the value of the CN as maximum number of requests which can easily be handled by the end user device. The proxy checks and works according to the CN value of the end user device which results in efficient prevention from flooding attacks. In the same manner, DDoS is prevented by analyzing the different callers calling to the same callee, at the same time slot, as shown in Table III.

V. Performance Evaluation

The proposed mechanism has been implemented in SIP Express Router (SER) which is one of the most known open source SIP

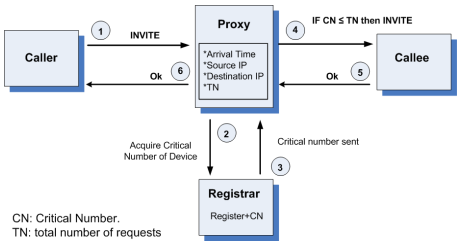

Figure 4: Block diagram of our scheme during call establishment phase

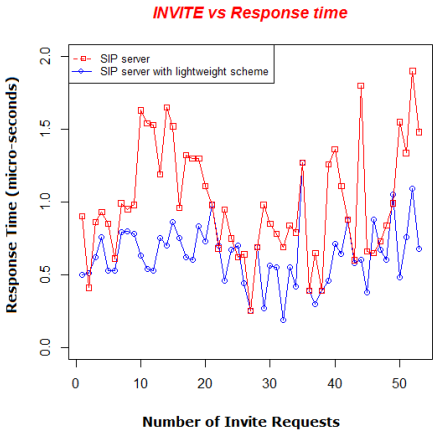

Figure 5: Number of INVITE requests vs. response time (under low flooding rate)

servers. The machine used for this platform is 1GHz processor with 1GB RAM. Figure 5 shows the number of INVITE requests versus the obtained response time, while Figure 6 depicts the number of REGISTER requests versus the obtained response time. As we can see from both figures, when our scheme is enabled (i.e. the blue curve) the obtained response time (in microseconds) is lower than that obtained in case of SIP standard (i.e. the red curve), and its average values range from 0.4 μs to 0.7 μs. Hence, this proves the effectiveness of our scheme.

It is worth mentioning that, in one part of our solution's evaluation scenarios, we have generated the attacks at a flooding rate that varies between 50 requests/second and 100 requests/second in both cases of INVITE and REGISTER. The obtained results show that the proxy server responds efficiently when our scheme is enabled.

As the proposed scheme introduces new header, we have extended the core of SER to support the CN header parsing similar to other existing mandatory headers. Consequently, as a SIP request message received by SER is able to be recognized and parsed automatically then the critical number header can provide this information to any application and service that needs it. Besides, the SER offers the ability to build new functionality and services through its modular architecture without affecting the existing services and applications. Thus, we have built our scheme in a module.

Figures 7 and 8 reveal that the response time of our scheme has increased compared to the results obtained in Figures 5 and 6 due to the high flooding rate that we have used in this scenario. The average values of the obtained response time range from 0.5 μs to 1.0 μs in both cases of INVITE and REGISTER requests. Despite this increase in the response time, our scheme still outperforming SIP standard as the response time of SIP server in this latter case is longer than that

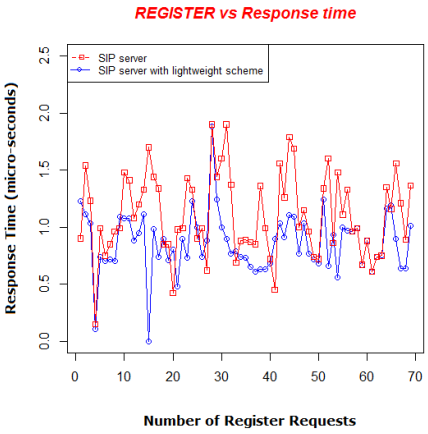

Figure 6: Number of REGISTER requests vs. response time (under low flooding rate)

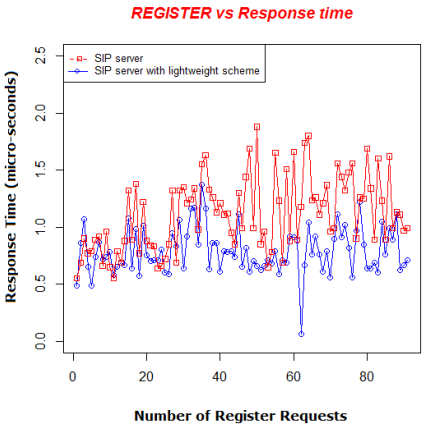

Figure 8: Number of REGISTER requests vs. response time (under high flooding rate)

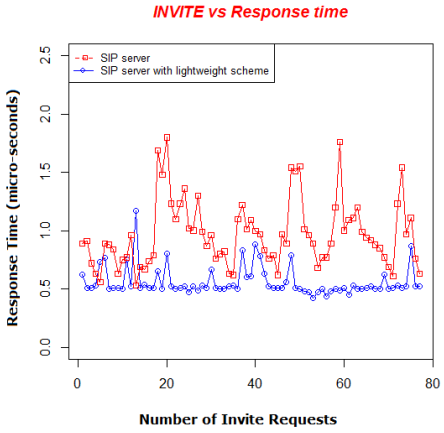

Figure 7: Number of INVITE requests vs. response time (under high flooding rate)

achieved under our scheme. Finally, we remark that the increase of the flooding rate leads to a little instability in terms of response time.

To summarize, we can say that our scheme performs very well under low rate DoS and DDoS attacks, however it experiences a slight instability under higher flooding rates. Nevertheless, the obtained response time under both of flooding rates is lower than that achieved by SIP standard (i.e., when our scheme is disabled). Therefore, our scheme is an effective countermeasure to the flooding attack in SIP.

VI. Conclusion

We have proposed a novel scheme to deal with INVITE flooding attack in SIP. In our scheme, we add a transition state table at the proxy side in order to ease the detection of the unusual reception of INVITE requests. Moreover, we slightly modify the REGISTER request header by adding the Critical Number field which is used by the end user to inform its corresponding proxy, during the registration phase, about the maximum number of calls that it can support. Thus, using this information the proxy is able to limit the number of INVITE requests forwarded to this callee and therefore prevents the congestion problem. We have mainly focused on two scenarios of flooding attack which are; the case of a single caller sending large number of INVITE requests towards the same callee within a small time interval and the case where multiple callers send INVITE requests towards the same callee simultaneously. According to the obtained results in our test-bed, our scheme significantly decreases the likelihood of congestion at both proxy and end user. This is justified by the short response time for both INVITE and REGISTER requests.

VII. Acknowledgement

This work was supported, in part, by Science Foundation Ireland grant 10/CE/I1855 to Lero - the Irish Software Engineering Research Centre (www.lero.ie).

References

[1] H. Schulzrinne et al., "RTP: A Transport Protocol for Real-Time Applications", *RFC 1889*, Jan. 1996.

[2] M. Handley et al.,"SDP: Session Description Protocol", *RFC 2327*, Apr. 1998.

[3] J. Rosenburg et al., "Sip: Session Initiation Protocol", *RFC 3261*, Jun. 2002.

[4] D. Kuhn, T. Walsh, S. Fries, "Security considerations of voice over IP Systems", *National Institute of Standards and Technology (NIST)*, Gaithersburs, MD, USA, Computer Security Division, Special Publication 800-58, Jan. 2005.

[5] G. Oramazabel, S Nagpal, E. Yardeni and H. Shulzrinne, "Secure SIP: A Scalable Prevention Mechanism for DoS Attacks on SIP Based VoIP Systems", *Springer-Valeg Berlin Heidelberg, IPTComm*, Jul. 2008.

[6] F. Huici, S. Niccolini and N. d'Heureuse, "Protecting SIP against Very Large Flooding DoS Attacks", *IEEE GLOBECOM, Hawaii, USA*, Dec. 2009.

[7] E. Y Chen, "Detecting DoS Attacks on SIP Systems", *IEEE VoIP Management and Security*, Vancouver, Canada, Apr. 2006.

[8] X. Deng, M. Shore, "Advanced Flooding Attack on a SIP Server", *ARES, pp.647-651, 2009 International Conference on Availability, Reliability and Security, Fukuoka, Japan*, 2009.

[9] W Conner, K Nahrstedt, "Protecting SIP Proxy Servers from Ringing-based Denial-of-Service Attacks", *In Proc. of the 2008 Tenth IEEE International Symposium on Multimedia (ISM), Berkeley, CA, USA*, Dec. 2008.

[10] J. Tang, Y. Cheng, and C. Zhou, "Sketch-Based SIP Flooding Detection Using Hellinger Distance", *IEEE GLOBECOM, Hawaii, USA*, Dec. 2009.

[11] G. Zhang, S. Elhert, T. Magendanz, D. Sisalem, "Denial of Service Attack and Prevention on SIP VoIP Infrastructure Using DNS Flooding", *In Proc. of the 1st international conference on Principles, systems and applications of IP telecommunications,* New York City, New York , Jul. 2007.

[12] Sislaem D Kuthan, J. Ehlert, S, "Denial of Service Attacks Targeting a SIP VoIP Iinfrastructure - Attack Scenarios and Prevention Mechanisms", IEEE Networks Magazine, Vol. 20, No. 5, Sep.-Oct. 2006.

[13] I. Husaain, F. Nait-Abdesselam, "Strategy based proxy to secure user agent from flooding attack in SIP" *In Proc. of the* 7^{th} *IWCMC 2011*, Istanbul, Turkey, 4-8 Jul., 2011.