# Performance Evaluation of a Flooding Detection Mechanism for VoIP Networks

Dimitris Geneiatakis
Dept. of Telecommunications Science and Technology,
University of Peloponnese
End of Karaiskaki St., GR-22100, Tripolis, Greece
email: dgen@uop.gr

Nikos Vrakas[1], Costas Lamrinoudakis[2]
Laboratory of Information and Comm. Systems Security
Dept. of Information and Comm. Systems Engineering
University of the Aegean
[1]icsdm07036@icsd.aegean.gr
[2]clam@aegean.gr

*Abstract*—**The internet based telephony services (IPTel) are mainly exposed to set of vulnerabilities that inherited from the employed protocols such as TCP/IP and proprietary VoIP protocols. One of the most critical threats in this sensitive environments is considered the denial of service (DoS) attacks. The main concern of a mechanism that focuses on detecting such attacks is the potential end-to-end delay between communicating parties. In this paper is described a hash based flooding detection mechanism and evaluated in an experimental test bed architecture. The outcomes demonstrate the potentiality of the mechanism as the end-to-end delay is negligible.**

***Keywords- Voice Over IP; Intrusion Detection Systems; Bloom Filter;Session Initiation Protocol***

## I. INTRODUCTION

As happens in most cases, the malicious users are trying to cause malfunctions to a system through the employment of various methods and techniques. Besides it is well known Internet services suffers from a numerous of vulnerabilities and attacks [1]. On top of that, the rapid growth of the Internet based telephony has aroused the market and consequently the malicious user interest. In these types of applications, attack consequences can be very harmful as a malicious user might achieve an unauthorized access or even cause a denial of service (DoS). As regard the latter it should be stated that in IP telephony (IPTel) the loss of availability is considered "devastating" for provider revenues. Moreover, telephony user requires at least similar availability levels to those provided by Public Switch Telephone Network (PSTN) services. As a result, the guarantee of service availability must be among the first priorities of any IPTel provider.

More specifically, IPTel services, as described in [2], not only inherit underlying protocols' vulnerabilities and attacks (e.g TCP flooding attacks) but also VoIP specific ones. In [2],[3] are presented various methods to overwhelm system's resources like memory, CPU or bandwidth, by creating numerous of useless well formed requests and consequently causing DoS. The objective of any flooding attack is to consume target's system resources constitutes it unavailable. In IPTel any network component should be considered vulnerable against flooding attacks. However, current security research works [2],[3] demonstrate that attackers will focus mainly on signaling servers, since the domination of signaling text based protocols like Session Initiation Protocol (SIP) [4] offers new flooding attack opportunities with low cost. Note that these types of attacks could be accomplished through architectures similar to those utilized for flooding attacks against web servers [5]. In this paper we extend the evaluation of a hashed based filter relied on Bloom filter, presented in [5]. Particularly, the focus is on the end-to-end delay introduced by such a filter, considering the case in which the proposed mechanism could be implemented as a built in service in SIP components. The experimental results demonstrate that the end-to-end delay is negligible.

The rest of the paper is structured as follows. Section II presents various flooding scenarios against SIP components, while Section III describes briefly an efficient mechanism for detecting such attack scenarios. Section IV evaluates the mechanisms in terms of end-to-end delay. Finally, Section V concludes the paper.

## II. FLOODING ATTACKS AGAINST SIGNALLING SERVERS

As already mentioned, SIP is the dominant signaling protocol for Next Generation Network. The text-based form constitutes it adaptive to various service requirements with low cost comparing to other alternative signaling protocols like H.323 [6]. However, malicious users exploit this advantage also in order to create SIP signaling messages to flooding the corresponding server or launching other type of attacks. In SIP there are two main network components required for service operation:

- Registrar: Responsible for administer users' registration procedure, by processing SIP REGISTER messages.
- Proxy: Responsible for call administration (forwarding messages, discovering network components location, etc), by processing SIP INVITE messages.

Utilizing different kind of techniques in SIP environments, an attacker can flood not only the core network entities but also the end user equipment (UE). In the following subsections are briefly described the various techniques for launching flooding attacks against those services.

### *A. Registrar Flood*

In this type of flooding a malicious user tries to cause a DoS in the registration service by creating a numerous of SIP

REGISTER messages against it. Taking into the account that in usual case SIP REGISTER messages requires authentication, the Registrar should execute continuously expensive cryptographic operations (under a flooding attack) consuming at a rapid pace the available resources. This attack can be launched by more than one malicious user (multi-source) at the same time, draining server's resource faster.

### B. *Proxy Server Flood*

These types of flooding attacks try to constitute the "core" of the service unavailable, by generating a numerous of uncompleted sessions to the proxy servers overwhelming their memory similar to TCP flooding attacks [1]. Different message constructions could be utilized to launch either a single or multi source flooding attack. Particularly, the attacker generates syntactically correct SIP INVITE requests with different request URI or call identification number (which characterize uniquely every call) in order to open different sessions in the side of the proxy server. A very large amount of such (semi-open) session can cause DoS, due to incredible waste of resources. On top of that, a malicious user may develop a distributed INVITE flood attack architecture by exploiting "innocent" SIP devices, commanding them to launch a SIP INVITE flooding attack. An example of such architecture is depicted in Fig. 1.

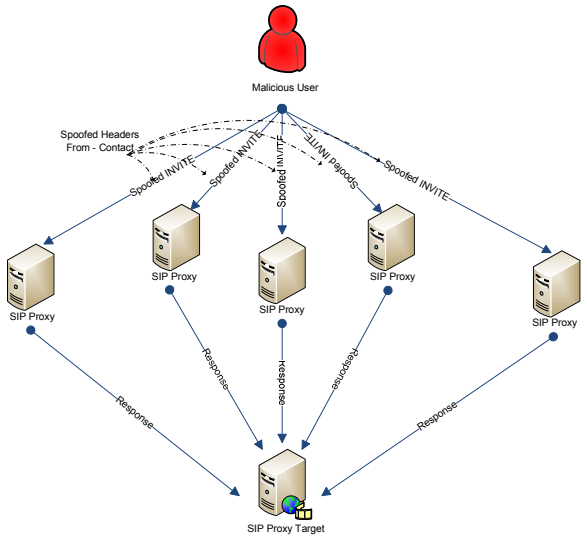


Figure 1: Distributed DoS in SIP

Alternatively, the attacker might craft an SIP INVITE message changing the sender's URI with the victim's one, forwarding it to "innocent" server. Afterwards, the latter instead of responding to attacker's address, reply directly to the victim's URI. This type of attack is achieved because in SIP there is no a validation procedure for the source URI of a received SIP INVITE message. Consequently, a distributed type of this attack could overwhelm victim's network and computational resources. Last by no means least, an attacker might use a non-existing URI in the FROM header in order to cause the proxy to allocate memory until an expiration alarm is triggered. This case of attack is depicted in Fig. 2.

### C. *End User Flood*

The attacks targeting to end user can easily accomplish their objectives due to the limited resources of the IPTel end devices. It is well known that the most devices of this type can handle a very small number of incoming calls. Inevitably, in case that receive tens of SIP INVITE message would cause a DoS to them. These attacks are very similar to those that could be launched against proxy servers. They main difference is founded on the fact that the destination in the SIP message should remain unchanged.

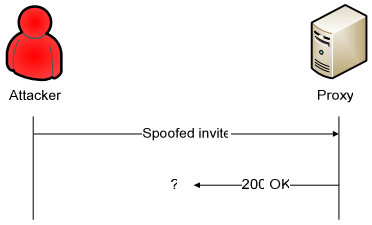


Figure 2: TCP-SYN attack

## III. MECHANISM DESCRIPTION

In order to detect the flooding attacks that could be launched against IPTel services a specific mechanism is suggested which is relied on the idea of Bloom filters [7] and a specific metric namely session distance. The proposed mechanism is consisted of a Session Establishment Monitoring System (SEMS) and a detection model utilizing the information collected by SEMS and the session distance.

### A. *Session Establishment Monitoring System*

This monitor consists of three distinct tables, one for every SIP message that required for the session establishment: INVITE, 200OK, ACK (Fig. 3). Note that session establishment in SIP is a 3-way handshake procedure. The caller generates a SIP INVITE message and forwards it to the appropriate SIP server, which locates the receiver (callee) and sends the call towards his device. The callee whenever receives such a request responds with a 200 OK message, considering that accepts the message. Afterwards, the session is established since the callee receives the final SIP ACK generated by the caller. Consequently to monitor these types of different SIP messages, we utilized three distinct "tables". The tables hold counters in their entries to enumerate the incoming messages. Every new message is hashed by two ore more functions in order to recorded it in the entries of the appropriate table. As input to hash functions are used the headers the headers "Call-Id" and "From" (identifies uniquely a session [4]). The hash function output points to a table position, in which the corresponding counter is increased by 1 (see Fig. 4). In Fig. 3 is summarized the monitoring algorithm this subsystem.

For each incoming message check the type
If type is request
Check the method
If method is INVITE
Update the invite Bloom filter (increase by one the appropriate entries of the filter)
Else if the method is ACK
Update the ack Bloom filter (increase by one the appropriate entries of the filter)
If type is final response
Update the response Bloom filter (increase by one the appropriate entries of the filter)

Figure 3: Monitoring algorithm

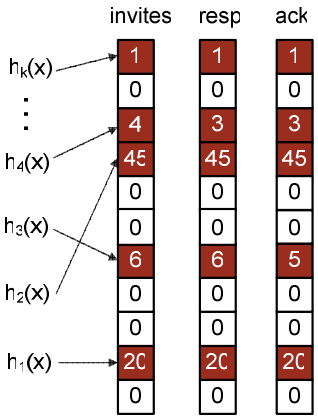


Figure 4: Snapshot of the session establishment monitoring system.

### B. *End-user Traffic Monitoring System*

The monitor traffic described in the previous sub-section could not record each and every of the incoming sessions are sent to a specific user, because this monitor logs only different sessions. Thus, we need an additional sub-system monitor to record sessions sent to a specific user. To accomplish this, the End-user Traffic Monitoring System consisted of one table; with its entries correspond to calls send to a specific user. As input to hash functions used the destination of the SIP incoming message ("To" header).

### C. *Detection Method*

The detection method is based on the observation that a flooding attack is related with a number of uncompleted sessions. This means that there is not an analogy between request, responses and final acks, while for each and every successfully established session exists an one to one relation between them. Thus, is introduced a metric that is responsible for modeling a completed session, called *session distance,* and defined as follows for the case of the SIP:

*Dist = Num of INVITEs - 0,5*(Num of OK + Num of ACK)*

Since the Dist value remains zero, a successful session has been established. Any other positive value represents uncompleted or dropped sessions. To detect a possible attack, the proposed system should be trained under normal traffic for some period of time in order to define the appropriate threshold value for Dist, which the provided service should not exceed during its "normal" operation. Fig. 5 presents the threshold assignment algorithm.

For all elements in the monitor do
$Session\_distance_i = \#invite_i - 0.5*(\#resp_i + \#ack_i)$
$Threshold\_value = +session\_distance_i$

Figure 5: Threshold value assignment algorithm

Afterwards, the final value for threshold, corresponds to different sessions, is computed by taking into account also some other factors such as the average network delay (Nd), and user responses delay (URT). The formula to compute the threshold value is the following one:

$$T_{alarm} = T_{sd1} + Nd_1 + URT_1 + \delta \ (1),$$

where $T_{sd1}$ is the mean value of the session distance after the training period and $\delta$ is a value that represents the capabilities of the host system. A DDoS alarm is triggered if the observed traffic exceeds $T_{alarm}$.

This threshold is utilized for detecting uncompleted sessions remains during a period into the server, however, as mentioned previously, could not detect flooding attacks against specific end user. For these reason, additional thresholds are required to detect attacks utilizing either the same message or recipient ("To header"). Particularly, to detect an attack that utilizes the same message, a $T_{single1}$ threshold is defined. This means that a specific entry in the filter could not exceed the $T_{single1}$. Furthermore, an additional threshold $T_{single2}$ is required to protect end users against attacks that correspond to different session but directed to the same end user. This specific threshold is applied on the End-user Traffic Monitoring System.

### D. *Detection Example*

The Fig. 6 presents a snapshot of mechanism and how thresholds been checked during sessions.

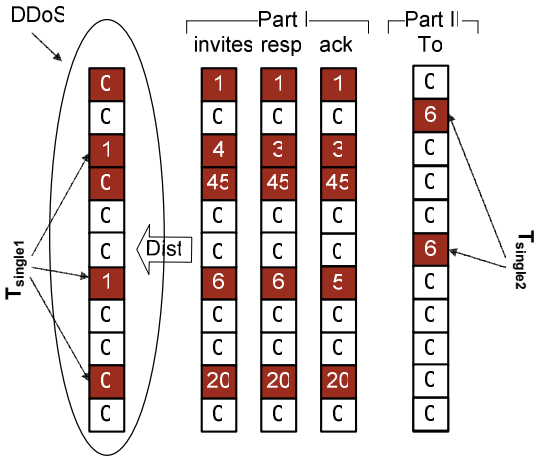


Figure 6: Threshold value representation

Specifically, the Dist function is applied on the table constitute the SEMS inferring the existence of DDoS attacks by comparing the DDoS table sum with $T_{alarm}$. Since no alarm is triggered in the first comparison every record of DDoS table is

checked if exceeds the $T_{single1}$ detecting the existence of an flooding attack utilizing the same message. Simultaneously, the records of End-user Traffic Monitoring System is searched to identify attacks targeting a specific user in case that exceeds the threshold $T_{single2}$ .

## IV. Evaluating the Proposed Mechanism

For the evaluation of the proposed mechanism, has been developed an experimental architecture that consists of the following components (Fig. 7):

- The SIP server: The well-known SIP Express Router (SER) (http://www.iptel.org/ser) has been utilized. Note that SER incorporates all the necessary network components (registrar, proxy, and redirects server)
- The legal call generators Alice and Bob: The SIP call generator namely as SIPp (http://sipp.sourceforge.net/) has been utilized.
- The malicious request generator Eve: A Proprietary SIP flooding attack generator has been utilized.

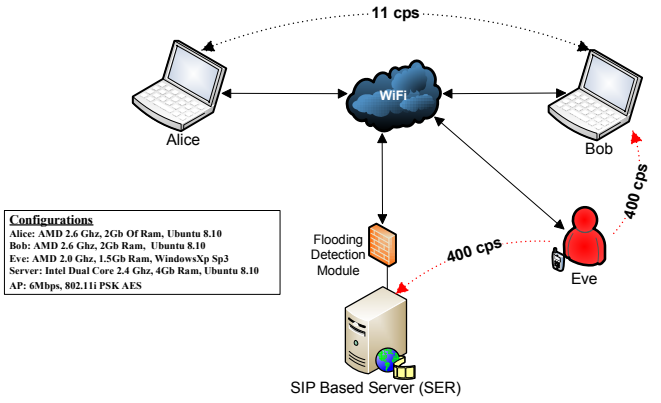


Figure 7: Test-bed architecture

For the scope of evaluation we employ five indicating scenarios described in table 1.

TABLE I. Test Scenarios

| Scenario Name | Scenario Description |
|---|---|
| ***Scenario 1 (S1)*** | In this scenario there is only legal traffic. Specifically the *"legal request generator"* generates requests (at a pace of 11 req / second), while the *"legal response generator"* generates the corresponding responses. |
| ***Scenario 2 (S2)*** | In this scenario the malicious user generates requests (at a pace of 150 req / second) that are addressed to a(n) (specific) innocent user who tries to respond to all of them with the existence of background traffic of 11 / call per second (cps). |
| ***Scenario 3 (S3)*** | In this scenario the malicious user generates requests (at a pace of 1 req / 10 microseconds) that are addressed through the proxy to various clients belonging to non existing domains with the existence of background traffic of 11 / cps. |
| ***Scenario4 (S4)*** | In this scenario the malicious user applies more effort to his attack increasing the generated requests at a level of 400 req / second. Apart from that, the other properties of the attack remain the same as in the S2. |
| ***Scenario 5 (S5)*** | In this scenario the malicious user applies more effort to his attack increasing the generated requests at a level of 400 req / second. Apart from that, the other properties of the attack remain the same as in S3. |

At this point it should be stated that the scenario 1, was utilized to train the detection mechanism for one hour period of time. After this period by utilizing the Formula 1 we compile the following threshold values: $T_{alarm}$ = 180, where Tsd1=150, Nd1+ URT1 = 30 and δ=0, while the values of $T_{Single1}$ and $T_{Single2}$ were set to 6 and 8 respectively. During the evaluation note that we did not block the attack traffic in order to demonstrate the robustness of the proposed mechanism, while we measure the end-to-end delay between the end terminals of Alice (caller) and Bob (calee). Note that we assume that the callee responds immediately since an incoming call has been received.

The attacker has bombarded the server in S4, S5 with 1.5 million messages/scenario and in S2 and S3 has forwarded 550 thousand messages/scenario. Figure 8 and 9 and Table II illustrate the end-to-end delay introduced by the proposed mechanism. Moreover, Fig. 9 depicts the necessity of such detection mechanism as in flooding attacks with high SIP message generation rate the attacker "introduce" a significant delay to the end user communication.

TABLE II. Results (Average End-to-End Delay)

| | S1 | S2 | S3 | S4 | S5 |
|---|---|---|---|---|---|
| ***Filter Off*** | 15.408[a] | 19.991 | 17.521 | 378.000 | 440.560 |
| ***Filter On*** | 14.975 | 19.708 | 18.727 | 417.000 | 446.010 |

a. Average delay time induced from end-to-end. Values are in mille-seconds.

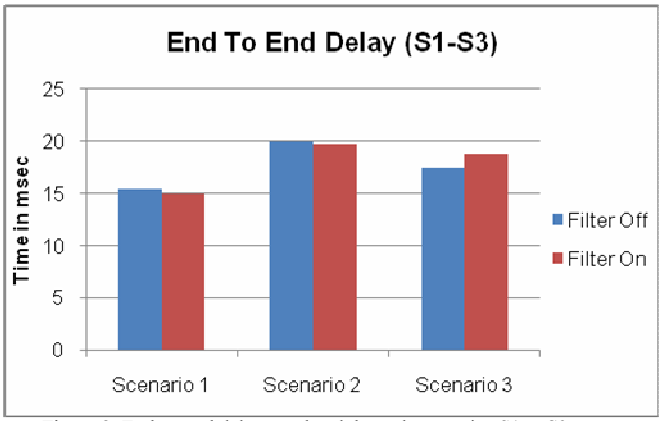


Figure 8: End-to-end delay overhead through scenarios S1 to S3.

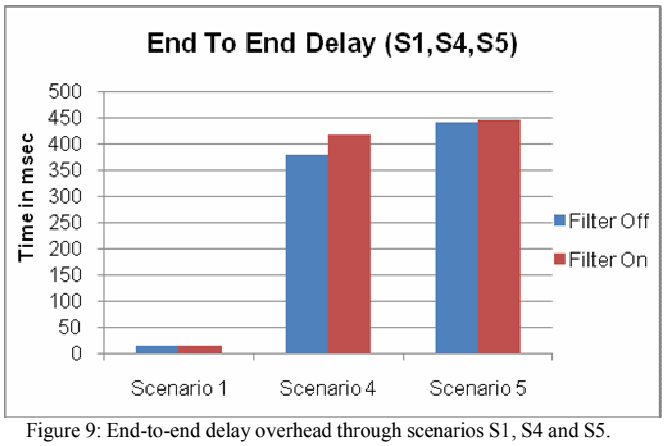


Figure 9: End-to-end delay overhead through scenarios S1, S4 and S5.

## V. CONCLUSIONS

Flooding attacks that could be launched against IPTel services could take various forms, affecting directly their availability, which is considered among the basic requirements for both end-user and service providers. Consequently, the employment of the appropriate mechanism to identify such attacks must be founded on the first defense line of a service provider. Under this context, in this paper we briefly describe flooding attacks in IPTel services relied on SIP signaling protocol, providing an efficient identification mechanism. The evaluation demonstrates that the introduced end-to-end delay is negligible, settles it among potential mechanism for identification in IPTel services.

## REFERENCES

[1] Peng, C. Leckie and R. Kotagiri. "Survey of Network-based Defense Mechanisms Countering the DoS and DDoS Problems" Accepted by ACM Computing Surveys.

[2] D. Geneiatakis, A. Dagiouklas, G. Kambourakis, C. Lambrinoudakis, S. Gritzalis, S. Ehlert, D. Sisalem, "Survey of Security Vulnerabilities in Session Initiation Protocol", IEEE Communications Surveys and Tutorials, Vo. 8, No. 3, pp. 68-81, 2006, IEEE Press.

[3] D. Sisalem, J. Kuthan and S. Ehlert, “Denial of service attacks targeting a SIP VoIP infrastructure: atttack scenarios and prevention mechanisms. IEEE Network, 20 (5):26-31,2006.

[4] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: session initiation protocol," RFC 3261, Internet Engineering Task Force, 2002.

[5] D. Geneiatakis, N. Vrakas and C. Lamrinoudakis, "Utilizing Bloom Filters for Detecting Flooding Attacks against SIP Based Services", Accepted for publication in Computer and Security, Elsevier.

[6] International Telecommunications Union, “Recommendation H.323”, available on-line http://www.itu.int/rec/T-REC-H.323/e

[7] Bloom B. Space/time trade-offs in hash coding with allowable errors. Communications of the ACM 1970;13:422–6.