ISSN 2047-3338

New Algorithm for SIP Flooding Attack Detection

Dahham Allawi, Alaa Aldin Rohiem, Ali El-moghazy and Ateff Ghalwash

***Abstract*—Voice over Internet Protocol (VoIP) services are based on standardized and open technologies (i.e., SIP, H.323) using servers reachable through the Internet, implemented in software and provided often over general purpose computing hardware. Therefore, such services can suffer from various security threats as denial of service attacks. In this paper we present a new hybrid (anomaly and misuse) SIP flooding attack detection algorithm, which overcomes the existing problems in many of other detection algorithms. The proposed algorithm is tested using simulated traffic datasets, and compared with three well known anomaly algorithms and one misuse detection algorithm. The test results show that the new algorithm has high detection accuracy and high completeness.**

***Index Terms*— VoIP Security, Session Initiation Protocol (SIP), Denial of Service (DoS) and Intrusion Detection System (IDS)**

I. Introduction

VOIP is the newest and the fastest growing technology which consists of routing voice conversations over IP-based network. The flexibility of the VoIP system and the convergence of voice and data networks brings with it additional security requirements. VoIP availability is the most important one. Denial of service (DoS) attacks are the main concerns causing loss of VoIP availability. Its impact ranges from decreasing of service level to complete loss of service.

SIP denial of service attack mechanisms differ according to attack type. Some attacks exploit vulnerabilities in SIP protocol implementation, another utilize drawbacks existing in RFC protocol specification, where others are resources consuming such as network bandwidth or agent processing capability.

Authors in [1] classify the SIP denial of service attacks into three different classes, namely: message flow Attacks, malformed message attacks, and the flooding attacks which is the most common, effective and the easiest to generate one.

Flooding attack involves transmitting a large quantity of forged SIP messages to a targeted VoIP system. SIP agents have to maintain transactional state until the transaction completes or the receiving agent times out, and thereby, slowing down when faced with a high request rate, resulting in a noticeable elongation in connection setup times.

The goal of intrusion detection systems is to accurately detect anomalous network behavior or misuse of resources (i.e. incidents), sort out true attacks from false alarms, and notify network administrators of the activity.

IDS is classified into two main categories: misuse detection systems and anomaly detection systems.

Misuse detection approaches attempt to model attacks on a system as specific patterns or signatures, and then systematically scan the system for occurrences of these patterns or signatures. While anomaly detection approaches attempt to detect intrusions by noting significant deviations from a normal behavior.

Flooding attack detection algorithms (anomaly and misuse) have several problems, these problems create an opportunity for an attacker to make undetectable harm. In this paper we will present these problems briefly, and then introduce a new hybrid SIP flooding attack detection algorithm. This new algorithm overcomes the mentioned problems. In addition, it has the ability to detect the wide range of flooding attack rates accurately. The rest of paper is organized into 4 parts. Part 2 describes the SIP flooding attacks parameters, evaluation parameters, and problems of major detection algorithms, part 3 presents the new detection algorithm. The evaluation of the new algorithm and the comparative study are done in section 4. Section 5 includes the conclusions.

II. Major SIP flooding attack evaluation

A. Test bed

For SIP protocol, attackers can use techniques to overload a SIP machine with uncompleted SIP transactions, such as INVITE or REGISTER requests. In the SIP INVITE flooding attack, the attacker generates a large numbers of INVITE requests, the SIP server receives the requests and maintains a transactional state for each one until the transaction completes

Dahham Allawi, Military Technical Collage, Cairo, Egypt, dahham78@hotmail.com.
Alaa Aldin Rohiem, Military Technical Collage, Cairo, Egypt, alaa_rohiem@yahoo.co.uk.
Ali El-moghazy, Military Technical Collage, Cairo, Egypt, moghazymtc@yahoo.com.
Ateff Ghalwash, Helwan University, Cairo, Egypt, Atef_ghalwash@yahoo.com.

or the transaction times out. Attacker in most cases does not authenticate his requests, causing the server to resend the authentication challenges many times latter [2]. As a result, the system is kept busy treating the bogus messages, valid ones will be treated at a much slower rate and the overall performance of the SIP server will decay.

To evaluate effect of SIP flooding attacks on SIP server, we use test bed which consists of: switch, attacker and monitoring station, 3CX SIP server, and two 3CX clients. In addition to Hardware components of test bed, we wrote two programs in C# programming language. The first is used for generating SIP normal traffic and measuring the response delay. While the second is used as attacking program, the two programs have the ability to capture the sent packets from a client to another, and generate a fixed predefined rate of standard SIP INVITE requests.

While attacker begins to direct his INVITE-flooding traffic to 3CX SIP server, clients begin to request their services (SIP normal traffic program starts working), at the same time, the monitoring station begins to capture, filter and log the traffic. The logged traffic is analyzed using special built Matlab software.

B. SIP flooding attack parameters

Generally, SIP requests are characterized using Poisson process [6]. In the normal state, the request rate is equal to or less than the SIP server service rate. During flooding attack, the average request rate is larger than the average service rate, and then the system is unstable, so the queuing model is not applicable and cannot provide any helpful information. The central element of the system is the server, which provides the service to the incoming requests and then issues the responses.

RFC 3261 specifies the maximum transaction timeout (tto) in SIP that equals to 64*T1 seconds, where T1 is an estimate of the round-trip time, and it defaults to 500 milliseconds [2]. Transactions that exceed this limit are discarded by the client. Here, we define two attack metrics: Safe Attack Effective Rate (SAER) and Attack Effective Time (AET). The first is defined as required attack rate causing a significant increase in the response time. The second is the required attack time causing delay in the response more than tto.

Figure 1 shows the test bed server response delay when we load the server with normal load (up to 90 request/second), where the high load for the used SIP server is approximately 1000 calls/min (about 90 request/sec) [4].

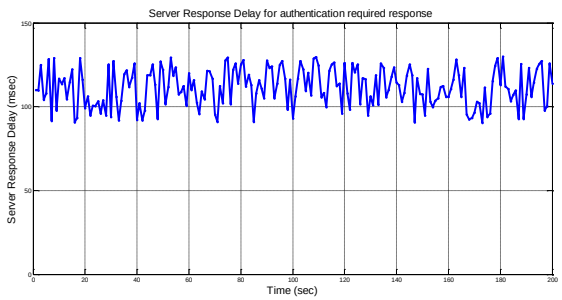


Fig. 1. SRD of REGISTER request for normal 3CX SIP traffic generator

By applying INVITE-flooding attack against 3CX SIP server, which is Windows-based software and fully supports RFC 3261, we note that the performance of the attacked SIP server is affected by two major factors, the attack time (needed attack duration to reach flooding state) and the attack rate. To identify this effect, different rates of SIP invitation flooding attacks were applied and the server response delay was measured. The Server Response Delay (SRD) is defined as the needed time for the server to receive the request, analyze it, and compose the response.

Now, we apply different attack rates on 3CX SIP server and then we analyze the results. Figure 2 shows the SIP server SRD when attacked by different rates of SIP INVITE-flooding attack.

Figure 2 allows us to define another new impact attack metric when the 3CX SIP server is attacked by different rates of flooding attack. This metric is the average server response delay in seconds which is given by the following equation:

$$T_{avrg} = \frac{\sum_{i=1}^{N} SRD_i}{N} \qquad (1)$$

Where:

T_{avrg} : is a mean value of response delay in second.

N : is total number of out coming responses from 3CX server every second.

SRD_i: is Server Response Delay for its request number i.

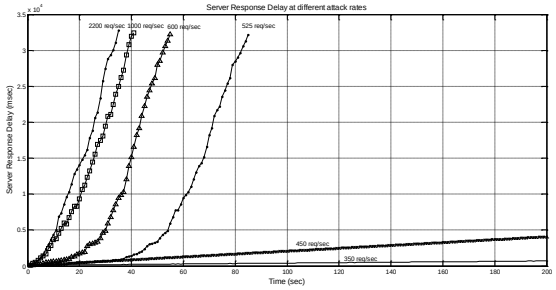


Fig. 2. 3CX SIP server SRD for different attack rates

In addition to the increase in the response time, attacks also cause requests losses (having no response). These losses may result from: Server buffer (queue) saturation and delayed responses (more than time out). Then the attacked SIP server responses are classified into the three following categories:

- Normal responses: which are generated in time less than tto (32 second according to RFC 3261).
- Time outed response: responses whose delay exceeds tto.
- Lost requests: requests which are lost in the network or discarded by the server due to congestion.

These categories percentages are affected by attack duration and attack rate. We can define two new attack impact metrics, called Percentage of Served Requests (P_{serv}) and Percentage of Failed Requests (P_{fld}), given by the following equations:

$$P_{serv} = \frac{N_{serv.req}}{TNR_{incom.\ req}} \qquad (2)$$

$$P_{fld} = \frac{NLR + NTOR}{TNR_{incom.\ req}} \quad (3)$$

Or

$$P_{fld} = 1 - P_{serv} \quad (4)$$

Where:

NLR: Number of Lost Requests. It is the number of requests that are lost in the network or discarded by the server.

NTOR: Number of Time Outed Response. It is the number of extra delayed requests (more than 32 sec).

$TNR_{incom.\ req}$: Total Number of incoming Requests.

$N_{serv.req}$: Number of served request with response delay less than 32 second.

Moreover, we can define relationship between percentage of served requests and attack rate. We note that percentage of served requests decreases as attack rate increases.

Figure 3 shows the relationship between attack rate and the attack effective time for 3CX SIP server (we obtained this relationship when we applied different attack rates to this server, then we measured flooding time). We can note that increasing attack rate, up to upper limit, Saturation Rate (SR), reduces the attack effective time, but rates that exceed SR nearly have the same effecting time. Here we define the SR as the upper limit of requests rate the server can receive, thus all attacks whose rates exceed SR will have the same effect.

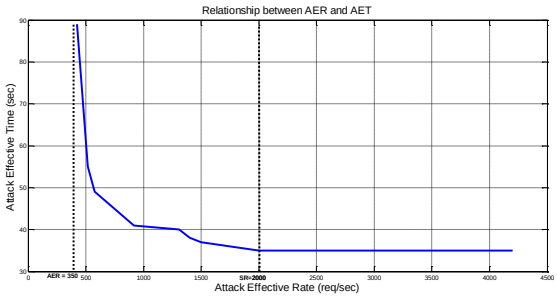


Fig. 3. Attack rate and attack effective time relationship for 3CX SIP server

Figure 2 and Figure 3 show that 3CX SIP server is able to process up to 350 requests/second with normal delay, but when requests exceed this rate the response delay begins to increase. So, according to our test bed, we can define the SAER as 350 request/second and 2000 request/second as saturation rate (SR). From Figure 3, we can conclude the followings:

- Attacks whose rate is in the vicinity of the SAER (350 request per second) cause a significant transaction failure after long periods of the attack time. We can define this type of attacks as Low Rate Attacks (LRA).

- Attacks whose rates exceed the SAER, but less than SR, push the server toward the failure state. The required time for the attack to cause server failure decreases as the attack rate increases. When the server reaches the failure state it does not produce any acceptable response. Medium Rate Attack (MRA), is a suitable name for this type of attack.

- Attacks whose rates exceed the SR, have the same effect, and push the server to the failure state after the same attack time. These attacks are very dangerous and have very short AET. High Rate Attack (HRA) is a good name for this attack type.

The performance results for 3CX SIP server show that this server has different behavior when attacked by different rates of flooding attacks. This behavior is figured out through the parameters: SR, SAER, AET, T_{avrg}, P_{fld}, and P_{serv} metrics.

C. Analyzing common detection algorithms for SIP flooding attack

In this section we first present four major intrusion detection systems, then we present an analysis showing the inherent problems in these systems.

1. Detection algorithms for SIP flooding attack:

- Adaptive Threshold algorithm:

Adaptive Threshold algorithm is a straight forward and simple algorithm, which relies on testing whether the average of a given feature in a predefined time window exceeds a particular threshold [6]. If X_n is the value of the feature in the nth time interval, and $\mu_n - 1$ is the estimated average of the feature from measurements prior to n, then the alarm condition is:

If $X_n > (\alpha + 1)\,\mu_n - 1$ then ALARM signaled at time n. (5)

$\alpha > 0$ is the amplitude factor, it indicates the percentage above the mean value that one considers to be an indication of anomalous behavior. The mean μn can be computed using an Exponentially Weighted Moving Average (EWMA) of previous measurements, as follows:

$$\mu_n = \beta\mu_n - 1 + (1-\beta)\,X_n \quad (6)$$

Where β: is the EWMA factor.

Adaptive Threshold algorithm is used to detect the SIP flooding attack by checking the rate of SIP requests. Its performance varies significantly with the variation in attack metrics.

- Cumulative Sum algorithm:

Cumulative Sum algorithm (CUSUM) belongs to the family of change point detection algorithms that are based on hypothesis testing to find time of switching from normal to abnormal request rate [4, 7]. It detects the abnormality much faster than the Adaptive Threshold algorithm [8]. The choice of Cumulative Sum algorithm is based on its simplicity in computation as well as its generally excellent performance [9]. Cumulative Sum algorithm was developed for independent distributed random variables {yi}. According to the approach, there are two hypothesis θ0 and θ1, where the first corresponds to the statistical distribution prior to a change and the second to the distribution after a change. The test for signaling a change is based on the log-likelihood ratio Sn.

$$Sn = \sum_{i=0}^{n} si \quad \text{where} \quad si = \ln \frac{P_{\theta 1}(yi)}{P_{\theta 0}(yi)} \quad (7)$$

Where:

n: is number of samples, yi : is requests rate at instant i, si: is

log-likelihood ratio at instant i.

The typical behavior of the log-likelihood ratio Sn includes a negative drift before a change and a positive drift after the change [6]. Therefore, the relevant information for detecting a change lies in the difference between the value of the log-likelihood ratio and its current minimum value. Hence the alarm condition for the Cumulative Sum algorithm takes the following form:

$$\text{If } g_n \geq h \text{ then an alarm is signaled at time n} \qquad (8)$$

Where:

$$g_n = S_n - m_n \qquad (9)$$

$$m_n = \min_{1 \leq j \leq n} S_j \qquad (10)$$

And: h is threshold parameter.

- *Hellinger Distance algorithm:*

Hellinger Distance algorithm (HD) measures the deviation between probability measures that does not make any assumptions about the distributions themselves [4].

HD is used to detect anomalies in SIP protocol. For example, we can use some of SIP features which are the number of INVITE, 200 OK, and REGISTER packets arrived in a predefined time-window. HD algorithm consists of training and testing phases. In the training phase, the normalized frequencies p_{INVITE}, p_{200OK}, $p_{REGISTER}$ for INVITE, 200OK, and REGISTER respectively are calculated over the training normal dataset. Similarly, the normalized frequencies q_{INVITE}, q_{200OK}, $q_{REGISTER}$ are calculated in the testing phase for each time-window n or interval. The HD between these frequency distributions of two phases is:

$$HD = \left(\sqrt{p_{INVITE}} - \sqrt{q_{INVITE}}\right)^2 + \left(\sqrt{p_{200OK}} - \sqrt{q_{200OK}}\right)^2 + \left(\sqrt{p_{REGISTER}} - \sqrt{q_{REGISTER}}\right)^2 \qquad (11)$$

To keep track of the normal attribute behaviors more accurately, authors in [4] use a dynamic threshold for detection. The threshold value is a function of the average of observed HDs and their mean deviation. Such a dynamic setting of threshold makes an attack harder to evade. They employ the stochastic gradient algorithm to compute the dynamic threshold based on the HD observed during the previous training period. Fast estimators for average ν and mean deviation ε given measurement HD, are computed as follow:

$$Err = HD_n - v_{n-1} \qquad (12)$$

$$v_n = v_{n-1} + g \times Err \qquad (13)$$

$$_n = _{n-1} + h \times (|Err| - _{n-1}) \qquad (14)$$

Where:

HD_n is the current sample of the HD, v_{n-1} and v_n are the previous and current means of HD, respectively, $_{n-1}$ and $_n$ represent the previous and current deviations.

During the testing periods, the Threshold (TH) is computed using the mean of HD and the mean deviation as following:

$$TH_n = x * v_n + y * _n \qquad (15)$$

The purpose of the multiplication factors x and y is to get a safe margin for the setting of the threshold value, so that HD avoids false alarms without degrading its detection sensitivity. These two factors are adjustable parameters, and can be properly tuned during the training period.

- *Weighted Sum algorithm:*

Weighted Sum (WSUM) is misuse detection algorithm, it depends on a prior knowledge about attacks signature, it seeks for attacks signature in the incoming samples, this algorithm makes using AET to detect the different types of SIP flooding attacks accurately [6]. The algorithm defines a new attack parameter called Attack Effective Factor (AEF), and it equals to the inverse of AET.

$$AEF = \frac{1}{AET} \qquad (16)$$

This parameter introduces a quantized evaluation for the harm done by flooding attack into the server each second, as the AEF increases the danger of attack increases. Since the AEF for the different flooding attacks is already known, the algorithm can calculate the attack effect during Δt seconds, it is Δt*AEF. In other meaning, during Δt seconds, the attacked server is pushed by Δt*AEF value toward compromised state. For example, if AET value equals to 100 second, then during Δt = 5 second, the server will be loss 5 * 0.01 = 0.05 of its resources, this percentage of resources will be unavailable.

To keep trace of the attack effect, the Weighted Sum algorithm samples the incoming requests each Δt seconds. For each sample (i) it calculates the average request rate (λi), and then allocates the corresponding AETi and AEFi, finally it computes the sample effect (Δt*AEFi). At the sample (n), the attack effect can be computed by cumulating the previous samples effects, calculating Cumulative Attack Effect (CAE), given by:

$$CAE_n = \sum_{i=1}^{n} \Delta t * AEF_i \qquad (17)$$

CAE_n reflects the server state at the time n Δt seconds, it expresses how much the server is pushed toward compromised state. When the server is in the normal state the CAE equals to zero. As the server is pushed towards the compromised state, the CAE increases, finally when the server is fully compromised the CAE will be equal to one.

2. Problems in major detection algorithms for SIP flooding attack:

All anomaly detection algorithms (Adaptive Threshold, Cumulative Sum, and Hellinger Distance) show variation in performance against variation in attack intensity, in the other meaning; these algorithms are unable to detect all types of SIP flooding attack simultaneously, where this weakness is related to tuning operation of parameters for each algorithm. As these algorithms detect the attack in its beginning, while it does not raise any alarm after the first beginning.

Also, anomaly detection algorithms make estimation about the next normal behavior depending on memorized quantity of previous samples. This memorized quantity brings up the attack masking and adaptation with attack problems.

The attack masking operation is related to the capability of attacker to attack the server with high rate of requests, these intrusive requests can be detected, but the main aim of attacker is to increase the detection threshold, creating the opportunity for attacker to inject another lower rate of flooding attack that is not detected by IDS. Attackers try to use high rate attacks for short duration as masks, they may be detected, but the following lower rate attacks remain hidden. Figure (4) demonstrates attack masking problem with Adaptive Threshold algorithm.

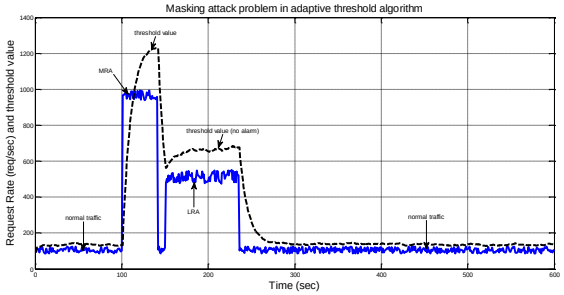


Fig. 4. Adaptive Threshold and attack masking problem

In adaption with attack problem, Attacker is not restricted to make a sudden change in the detection threshold, where attacker increases the attack rate gradually to raise the mean request and thereby the detection threshold is increased too. Repeating this scenario increases the detection threshold up to unlimited bound, causing the attack to pass without any noticeable trace. Figure (5) shows how attacker can increase the detection threshold, such that MRA becomes undetectable.

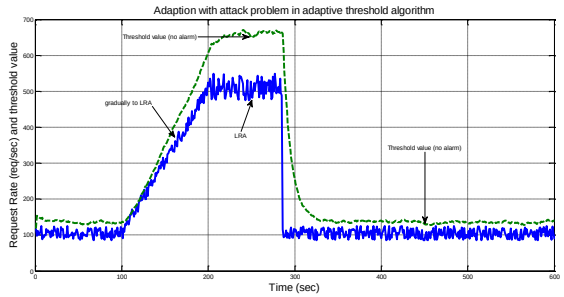
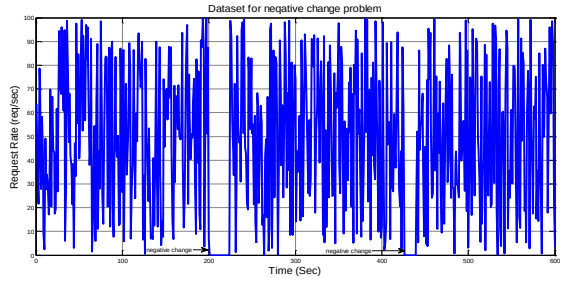
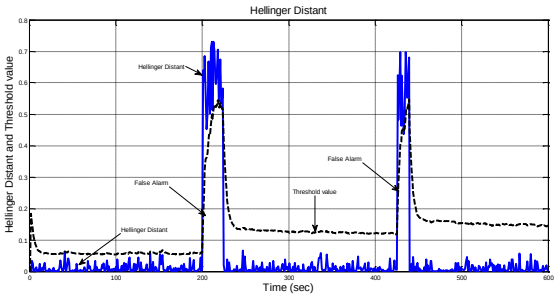


Fig. 5. Adaptive Threshold and adaption with attack problem

Hellinger Distance suffers from negative change problem. It detects negative changes in traffic rate as an intrusion [6]. HD algorithm is a sum of square values, so it does not distinguish between increasing requests rates or decreasing rates, it detects the decreasing rate as intrusion and issues a false alarm. Figure 6 demonstrates this case.



(a): Simulated dataset for Negative change problem



(b): Hellinger Distance value and its threshold value

Fig. 6. Negative change problem for Hellinger Distance algorithm

Weighted Sum algorithm has the ability to detect different attack types, so it has high detection accuracy, and minimum false alarms rate. But this algorithm suffer from important problem, this problem is adaption with threshold setting. Where the attacker can adapt with value of set threshold, then he can configure his SIP flooding attack rate depending on value of threshold. The attacker estimates time of attack, then he sends his SIP flooding attack with attack time less than estimated time, after that he can send several undetected successive attacks to push the server to the full compromised state. Figure 8 demonstrates adaption with threshold setting problem when using Weighted Sum algorithm in detection process.

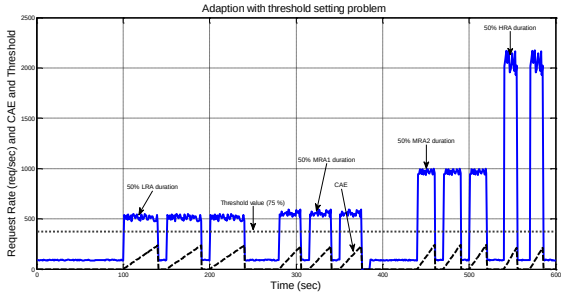


Fig. 7. Adaption with threshold setting in Weighted Sum algorithm

III. THE NEW PROPOSED DETECTION ALGORITHM:

A. Introduction

Rapid response, minimal false alarm rate, and the capability to detect a wide spectrum of attacks are the crucial features of any IDS [8].

In the previous study for the four algorithms we had seen that the previous three features are not satisfied completely in the studied detection algorithms, because of the following reasons:

- Most of these algorithms (Adaptive Threshold, CUSUM, and HD) are anomaly ones, they characterize the normal behavior and then seek for deviations. So these algorithms have no information about attacks types, they handle all attack types in the same way, making it impossible to have high detection accuracy for all attacks types.
- Most of these algorithms (Adaptive Threshold, CUSUM, and HD) make estimation about the next normal behavior depending on memorized quantity of previous samples. This

memorized quantity brings up the attack masking and adaptation with attack problems.
- Some of algorithms (HD) suffer from negative change problem, because this algorithm is a sum of square values, so it does not distinguish between increasing requests rates or decreasing rates.
- All the four algorithms do not detect the attack along duration of attack, it detects the attack only at the first seconds of attack, because it depends on the previous sample in calculating dynamic threshold value. The best case is that attack must be detected along attack duration.
- Some of algorithms (WSUM) suffer from adaption with threshold setting problem, because this algorithm uses static threshold value in its detection.

Table (1) summarizes all previous problems for the four algorithms. These previous reasons make us think about new detection one. We must seek for new robust alternative solution which satisfies the following requirements:
- It must have the ability to make distinction between different attacks types via the detection process. This distinction allows detection of all attacks types, with minimum false alarm rate.
- It must be able to overcome the attack masking, negative change, adaption with threshold setting, and adaption with attack problems.
- It must detect the SIP flooding attack along the whole duration of attack, not only at the beginning.
- It must respond rapidity to give the opportunity to administrator for prevention.
- Detection algorithm must ensure high probability of detection, and minimum false alarm rate.

TABLE (1): PROBLEMS FOR THE FOUR ALGORITHMS

Algorithm	Problems				
	Attack masking	Adaption with attack	Negative change	No high detection accuracy for all attacks types	Adaption with threshold setting
Adaptive Threshold	X	X		X	
CUSUM	X	X		X	
HD	X	X	X	X	
WSUM					X

B. Other related works

In [10], they propose an improved security-enhanced SIP System to reduce effect of SIP flooding attacks. This mechanism involves two components: an improved security-enhanced SIP server, and an improved security enhanced firewall. Requests from legitimate users are sent to a different queue at the firewall, and get passed to the SIP server directly, thus reducing the impact of the flooding traffic on legitimate users significantly. The experiment which done on special implemented SIP server shows that this mechanism reduces the call setup delay to a reasonable period, and improves the call setup delay during attack. In [11] a simple algorithm to detect SIP INVITE flooding attacks is introduced. It counts the received INVITE messages for the same destination within a certain amount of time. If there is a sudden surge of INVITE requests that exceed a predefined threshold, it is considered as a strong indication of flooding attack. Performance evaluation is done on simulated VoIP network topology and show that the online placement of the proposed algorithm has negligible impact and high detection accuracy. In [12] a finite-state machine for SIP transactions is defined, and the state is updated for each incoming request. Using this state machine four parameters are extracted, they are the number of transaction errors per second, the number of SIP application errors per second, the number of transactions per node, and the packet rate per transaction. One parameter threshold violation is considered as an attack indication.

C. The new proposed detection algorithm

As we know that in normal case, number of incoming requests is approximately equal to number of served requests (that are served by server during time less than 32 second), while when the SIP server is attacked the percentage of served requests (number of served requests divided by total number of incoming requests) will decreases, and the average response time will increases with increase of attack duration.

The main idea of the proposed detection algorithm is full monitoring for SIP server behavior during operation. The monitoring is based on simultaneous observation of three parameters (attack rate, percentage of served requests, and average response time). When SIP server is attacked, the algorithm will detects the different attack types of SIP flooding accurately, and it will overcomes all pervious problems.

We can summarize the steps of new method as following:
- Calculate R_{incom} by counting the requests that arrive to the server, where R_{incom} is number of incoming requests (normal traffic is merged with attack traffic) to SIP server per second.
- Distinguish source of that incoming requests depending on inspection operation in every request that arrives to server and determine source IP address of request sender.
- Identify threshold for R_{incom} called TH_R depending on relationship between the AET and AER, where AET is the attack effective time, AER is the attack effective rate.
- Calculate P_{serv}, that indicates percentage of served requests per second, and it is given by equation (2).
- Identify threshold for P_{serv} (as we see later) called TH_p depending on behavior of SIP server when it is attacked by different types of flooding attacks.
- Calculate T_{avrg}, that indicates mean value of server request/response delays in seconds, and it is given by equation (1).
- Identify threshold for T_{avrg} (as we see later) called TH_T depending on behavior of SIP server when it is attacked by different types of flooding attacks.
- The system raises an alarm when all of the followings are true:

$$R_{incom} > TH_R \,.$$
$$P_{serv} < TH_p \,. \qquad (18)$$

$$T_{avrg} > TH_T .$$

We show later that the new algorithm can detect all different SIP flooding attack types, as it solves all problems that are found in other algorithms, analyzed in section 2.

D. Inferring thresholds values of attack parameters

To estimate thresholds values, a statistical program is written with C# language. This program is installed on server to perform:

- Counting the incoming requests to 3CX SIP server, and identifying their sources, then sorting them depending on SIP method name.
- Counting the out coming responses from 3CX SIP server and calculating response delay time for each response, and calculating average of response delay time by equation (1)
- Identifying the served requests, failed requests, and lost requests every second.
- Calculating percentage of served requests P_{serv} using equation (2), percentage of failed requests P_{fld} every second using equation (4).
- launching alarm when SIP flooding attack is detected (if inequalities (18) are satisfied).

Evaluating thresholds values:

Our new algorithm calculates the incoming requests rate every second. According to behavior of 3CX Server shown in section 2, we can set value of incoming requests rate threshold equal to Safe Attack Effective Rate (SAER) for all different SIP flooding attack types, in other meaning we can say: $TH_R = 350 \ \ req/sec$

Now we make interpolation and curve fitting for the relation between incoming requests rate and percentage of served requests within predefined space depending on experimental results that we obtained it by using C# previous program, as shown in Figure 8.

Then, we define dynamic threshold for percentage of served requests TH_P, this dynamic threshold is related to incoming requests rate every second given by the following equation:

$$TH_P = a * R_{incom}^3 + b * R_{incom}^2 + c * R_{incom} + d \quad (19)$$

Where:

$a, b, c,$ and d are constant values, as shown Figure 8.

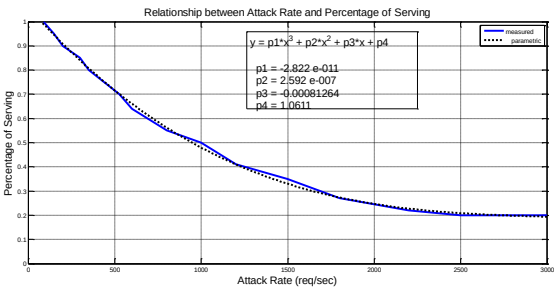


Fig. 8. Relationship between percentage of served requests and attack rate

In the same way, we can formulate approximately relationship between incoming requests rate and average of response delay within predefined rang of attack rates depending on experimental results that we obtained it by using C# previous program, as shown in figure (9).

Then, we define dynamic threshold for average of response delay. This is related to incoming requests rate by the following equation:

$$TH_T = e * R_{incom}^3 + f * R_{incom}^2 + g * R_{incom} + h \quad (20)$$

Where:

$e, f, g,$ and h are constant values, as shown in figure (9).

To avoid false alarm problem, we define safety factor (SF) as multiplication factor for this threshold. Effectiveness of safety factor (SF) appears in maximum load cases on the 3CX server where the average response time is relatively high (more than the average response time in normal cases). Thus safety factor increases the threshold value to reasonable value (sometimes up to double), this value increases response rapidity but it increases detection accuracy, and then the previous equation becomes:

$$TH_T = SF * \left(e * R_{incom}^3 + f * R_{incom}^2 + g * R_{incom} + h\right) \quad (21)$$

Where:

SF: safety factor, $1 < SF < 2$.

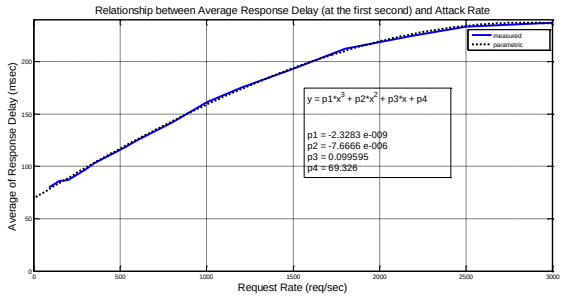


Fig. 9. Relationship between attack rate and average of response delay

E. Applying the new proposed algorithm to detect SIP flooding attacks

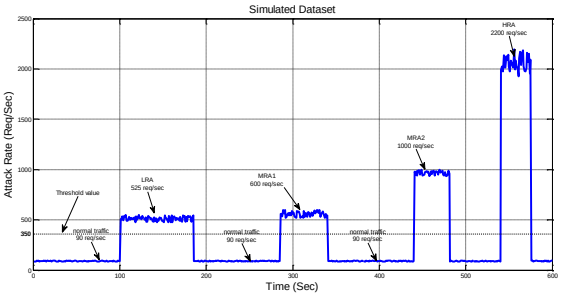
Now, using our test bed, we wish to apply our new proposed detection algorithm against different types of SIP flooding attacks. These attacks have different rates.

The algorithm calculates incoming requests rate, average of response delay, and percentage of served requests, after that, it calculates the three thresholds as described in the two previous subsections respectively. If the values of the compared three features with the three thresholds values are satisfied simultaneously, the algorithm will launch an alarm as indication of flooding attack.

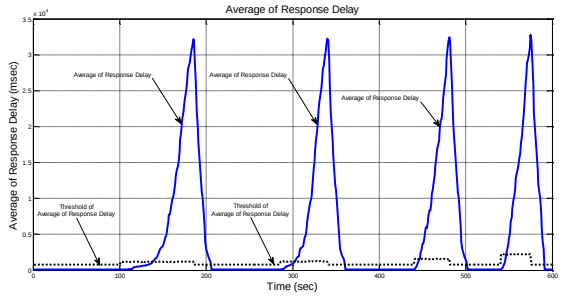
For graph presentation for attack detection, we define new factor, it is called detection level. This factor takes one of four values (0, 100, 200, 300) according to the satisfied thresholds in inequalities (18). For example, if one of thresholds is satisfied, detection level will be equal to 100, if two 200 and

thus. Our new proposed algorithm detects flooding attack when the three thresholds are satisfied, in other meaning, the detection level is equal to 300.

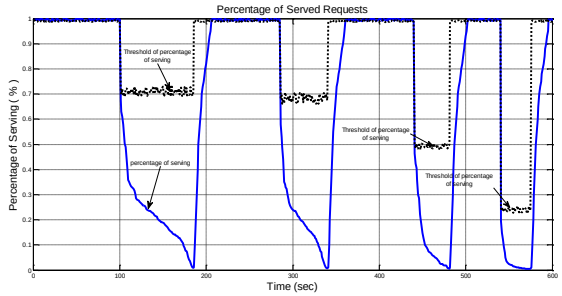
We apply our new proposed algorithm to the same dataset which was used previously with the other four detection algorithms. Figure 10 shows attack detection by the proposed algorithm, where (a) shows the simulated dataset (the continual line) and threshold values of attack rate (the pointed line), (b) shows the average of response delay along the time (the continual line) and its threshold values (the pointed line), (c) shows percentage of served requests along the time (the continual line) and its threshold values (the pointed line), while (d) shows detection level (the continual line) and alarms of attack detection by our new algorithm.



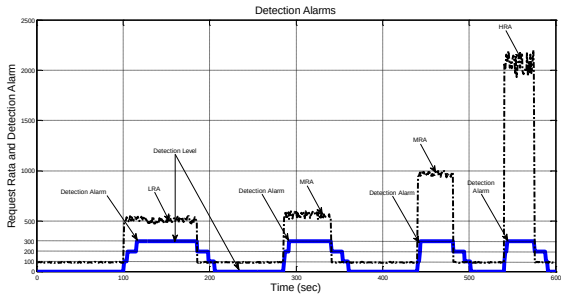
(a): Attack rates and its threshold value



(b): Average of response delay and its threshold value



(c): Percentage of served requests and its threshold value



(d): Detection alarms

Fig. 10. SIP flooding attack detection using our proposed algorithm

F. How new proposed algorithm overcomes the other algorithms problems

1. Attack masking problem:

Our proposed algorithm is considered hybrid (misuse and anomaly detection algorithm), depending on a previous knowledge about attacks signature, it seeks for attacks signature in the incoming requests, average of response delays, and percentage of served requests every second, and then it defines three dynamic thresholds to detect intrusions. No prediction about normal behavior is done, and inspection which is done on the current requests is not related to the previous ones. These features eliminate the chance for attack masking or adaptation with attack problems. Figure 11 and Figure 12 ensure that our proposed algorithm is not affected by these two problems. Figure 11 shows how our proposed algorithm solved attack masking problem, where it detects both attack mask and next attack because our proposed algorithm monitors the full behavior of SIP server (it monitors three main parameters). Where the continuous line demonstrates simulated dataset of attack masking problem, and the pointed line demonstrates the final threshold (all the three thresholds are true), while the dashed line demonstrates when the system will launch alarm as signaling existing intrusion.

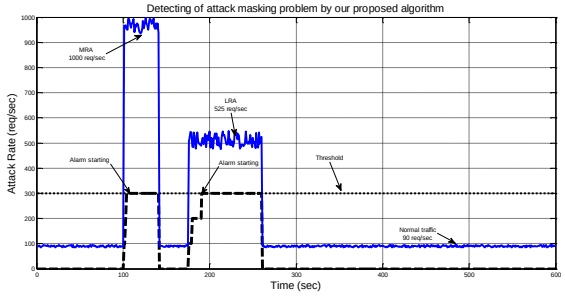


Fig. 11. Attack masking problem and our proposed algorithm

2. Adaption with attack problem:

Figure 12 shows how our proposed algorithm also solved adaption with attack problem, where it detects the attack with good accuracy within short time, the reason is that our proposed algorithm monitors the full behavior of SIP server. Where the continuous line shows simulated dataset of

adaption with attack problem, and the pointed line demonstrates the detection level (all three thresholds are true), while the dashed line demonstrates when the system will launch alarm signaling intrusion.

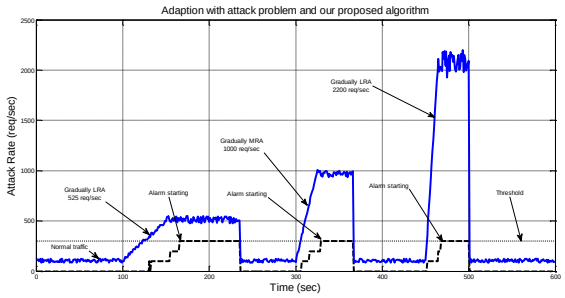


Fig. 12. Adaption with attack problem and our proposed algorithm

3. Negative change problem:

Our proposed algorithm processes the current requests only every second, for this reason, it does not suffer negative change problem. Figure 13 shows how our proposed algorithm does not suffer from negative change problem and it does not detect any intrusion. Where the continuous line demonstrates simulated dataset of negative change problem, and the pointed line demonstrates the final threshold (all the three thresholds are true), while the dashed line demonstrates when the system will launch alarm as signaling existing intrusion.

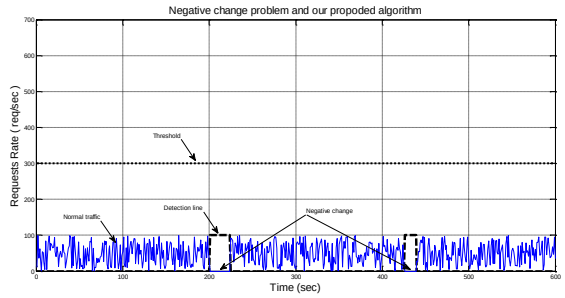


Fig. 13. Negative change problem and our proposed algorithm

4. Adaption with threshold setting problem:

Our proposed algorithm processes three parameters simultaneously, and it sets two dynamic thresholds every second. These two reasons make our proposed algorithm does not suffer adaption with threshold setting problem, as shown in Figure 14.

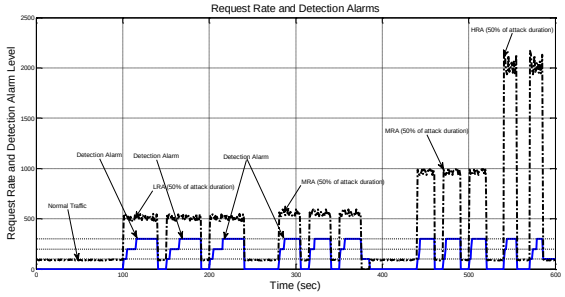


Fig. 14. Adaption with threshold setting problem and our proposed algorithm

Our proposed algorithm processes the samples every second, and values of dynamic thresholds depend only on requests rate. For these two reasons, our proposed algorithm can detect the SIP flooding attack along attack duration, and it has no false alarm.

IV. IDSs PERFORMANCE CRITERIA AND THE NEW ALGORITHM EVALUATION

Before comparing our proposed algorithm with other detection algorithms, we recall the IDS performance criteria. The active IDS must accurately detect all attacks types with the lowest false alarms rate and rapid response. IDSs may fail to detect intrusions, or sound the alarm when no intrusion has occurred. The following parameters are usually used as industry standards to measure how good the developed IDS is. These parameters are [13]:

- *Correct alarm*: an intrusion has occurred and the IDS has generated an alarm.
- *Correct rejection*: no intrusion has occurred and the IDS has not generated an alarm.
- *False alarm*: no intrusion has occurred and the IDS has generated an alarm.
- *False rejection*: an intrusion has occurred and the IDS has not generated an alarm.

Related to these four cases, the detection completeness, detection accuracy and response rapidity are introduced as three main IDS parameters, these parameters are used to evaluate the IDS performance [13]:

The accuracy: this represents the number of correct alarms divided by the number of correct alarms plus false alarms.

$$Accuracy = \frac{CA}{CA + FA} \qquad (22)$$

Where:

CA, FA: Number of correct alarm and false alarm respectively.

The completeness: this represents the number of correct alarms divided by the number of correct alarms plus false rejections. The more complete IDS is the fewer the intrusions that remain undetected.

$$Completeness = \frac{CA}{CA + FR} \qquad (23)$$

Where:

FR: Number of false rejection.

In the ideal case, an IDS would be 100% complete (it detects all intrusions) and 100% accurate (it produces no false alarms).

Response rapidity: is the needed time to detect flooding attack. When this time is small, IDS has enough opportunity to prevent this attack, where most of resources for SIP server are still available. But when detection time is large, the SIP server will have smaller time to take action, and prevention countermeasures in this case will be useless.

Using the IDSs evaluation parameters introduced in previous subsection, we will evaluate our proposed algorithm performance. The evaluation process is done using several simulated datasets. Each dataset represents the SIP requests

rates for 6 hours, Poisson arrival with average request rate (90 requests/second) is used to simulate normal traffic. Datasets are injected by different types and different numbers of SIP flooding attacks according to the Table (2).

TABLE (2): THE DATASETS SIP FLOODING ATTACKS NUMBERS

Dataset	Number of LRA	Number of MRA	Number of HRA
1	20	30	30
2	50	75	75
3	50	100	100
4	75	100	150

For each dataset, the detection completeness, false alarms rates, and the corresponding detection accuracy, and response rapidity are measured for our proposed algorithm and the four other algorithms. The results are shown in Table (3).

We note that our proposed algorithm has very high detection accuracy, and very high completeness, so it has minimum false alarms rate. Moreover our proposed algorithm provides a good estimation for attacks types due to it counts the incoming request every second, this indicator helps in estimating the needed prevention time.

TABLE (3): EVALUATION PARAMETERS FOR DETECTION ALGORITHMS

Parameter		Proposed algorithm	WSUM	HD	CSUM	Adaptive Threshold
Accuracy	dataset1	1	1	0.76	0.86	0.73
	dataset2	1	0.98	0.83	0.87	0.81
	dataset3	1	0.99	0.91	0.95	0.84
	dataset4	1	0.97	0.93	0.98	0.85
	Average	1	0.99	0.86	0.92	0.81
Completeness	dataset1	1	1	0.99	0.97	0.99
	dataset2	1	0.99	0.99	0.97	1
	dataset3	1	1	0.99	0.99	0.99
	dataset4	1	0.98	1	0.99	0.98
	Average	1	0.99	0.99	0.98	0.99
Response Rapidity (Percentage of flooding time)	dataset1	0.16	0.50	0.04	0.09	0.09
	dataset2	0.10	0.50	0.05	0.09	0.08
	dataset3	0.11	0.50	0.05	0.10	0.07
	dataset4	0.07	0.50	0.09	0.08	0.07
	Average	0.11	0.50	0.06	0.09	0.08
Response Rapidity (Sec)	Average	6.05	27.5	3.3	4.95	4.4

The reason of superiority of our algorithm is due to that the algorithm monitors three main parameters in server behavior, thus it got on best accuracy (no false alarm) and best completeness (no false rejection) while other algorithms monitor only one parameter. The response rapidity for some other algorithms compared to our proposed algorithm is also related to the larger processing in our algorithm.

Generally, we can say that our proposed algorithm is active, trusted SIP flooding attack detection algorithm.

V. CONCLUSIONS

The detection algorithm is a new proposed SIP flooding attack detection algorithm that has the ability to detect different types of SIP flooding attacks with lower false alarms rate. We can say that it is a hybrid (misuse and anomaly) detection algorithm which utilizes several features to detect SIP flooding attack. These features reflect effectiveness of the flooding attacks on the server performance as a signature which is used in the detecting process. It does not suffer from the attack masking, adaptation with attack, negative change and adaption with threshold setting problems. Moreover, it estimates the attack type that could help in prevention process. The minor increase in response delay can be tolerated by using modern powerful tools for detection.

REFERENCES

[1] H. Al-Allouni, A. Rohiem, M. H. Abd El-Aziz, A. El-moghazy, "VoIP Denial of Service Attacks Classification and Implementation", in the proceeding of 26th National Radio Science Conference, Cairo, Egypt, March 2009.

[2] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP:Session Initiation Protocol", RFC 3261, IETF Network Working Group, 2002.

[3] H. Schulzrinne, S. Narayanan, J. Lennox, and M. Doyle, "SIPstone- benchmarking SIP server performance", Technical Report, Department of Computer Science, Columbia University, New York, 2002.

[4] M. Akbar, Z. Tariq and M. Farooq," A Comparative Study of Anomaly Detection Algorithms for Detection of SIP Flooding in IMS", In 2nd International Conference on Internet Multimedia Services Architecture and Applications, India, 2008.

[5] H. Wang, D. Zhang, and K. Shin, "Change-Point Monitoring for the Detection of DoS Attacks", IEEE Transactions on Dependable and Secure Computing, Vol. 1, No. 4, Oct.-Dec., 2004.

[6] Husam Al-Alouni, "security of voice over internet protocol", PhD of science thesis, military technical college, Cairo, 2010.

[7] M. Basseville and I. V. Nikiforov," handbook of Detection of Abrupt Changes: Theory and Applications", Prentice-Hall, 1993.

[8] B. Rozovskii, A. Tartakovsky, R. Bla˘zek, and H. Kim, "A novel approach to detection of intrusions in computer networks via adaptive sequential and batch-sequential change-point detection methods", IEEE Transactions on Signal Processing, 2006.

[9] H. Wang, D. Zhang, and K. Shin,"Detecting SYN flooding attacks", in Proceedings of Annual Joint Conference of the IEEE Computer and communications Societies, February, 2002.

[10] Xianglin Deng and Malcolm Shore, "Advanced Flooding Attack on a SIP Server", In Proceedings of the The Forth International Conference on Availability, Reliability and Security, Fukuoka, Japan, March 2009.

[11] H. Sengar, D. Wijesekera, H. Wang, and S. Jajodia, "VoIP Intrusion Detection Through Interacting Protocol State Machines, In Proceedgins of the 2006 International Conference on Dependable Systems and Networks (DSN 2006), June 2006.

[12] E. Chen, "Detecting DoS attacks on SIP systems," in 1st IEEE Workshop on VoIP Management and Security, P 53–58, 2006.

[13] Husam Al-Alouni, "An Intrusion Detection Approach to Computer Networks", master of science thesis, military technical college, Cairo, 2003.