Military Technical College
Kobry El-Kobbah,
Cairo, Egypt

7th International Conference
on Electrical Engineering
ICEENG 2010

Problems of SIP Flooding Attacks Anomaly Detection Algorithms

By

H. Al-Allouni* A. Rohiem** M. Hashem*** A. El-moghazy**

alohusam@yahoo.com alaa_rohiem@yahoo.co.uk mhashem100@yahoo.com moghazymtc@yahoo.com

Abstract:

Session Initiation Protocol (SIP) is vulnerable to a wide variety of Denial of Service (DoS) attacks, flooding is the most common, effective and the easiest to generate one. In this paper we present an evaluation study to four well-known anomaly detection algorithms, namely: Adaptive Threshold, Cumulative sum (CUSUM), Non Parametric Cumulative Sum (NP-CUSUM), and Hellinger Distance (HD). The evaluation is assisted using simulated traffic dataset. We show that these algorithms suffer from two main problems, the first is called attack masking and the second is adaptation with attack. In the attack masking, attacker sends preamble followed by the attack. The preamble changes the tuned parameters of the detection algorithm, these changes mask the attack and keep it undetected. Attacker in the second problem deviates the detection algorithm parameters gradually, in such a way the attack is considered as normal traffic. The paper also shows that NP-CUSUM and HD algorithms, which utilize the protocol behavior to detect intrusion, suffer from third problem, and they are very simple to con. Attacker simply follows the same protocol behavior, and its related traffic is considered as normal, and cannot be detected.

Keywords: *session initiation protocol; flooding attacks; denial of service; anomaly detection; Adaptive Threshold; cumulative sum; non parametric cumulative sum; Hellinger distance.*

* Syrian Armed Forces
** Egyptian Armed Forces
*** Ain Shams University, Cairo, Egypt

1. Introduction:

Transferring Voice over Internet Protocol (VoIP) is a new promising technology, due to its simplicity, flexibility and low cost compared to traditional public switched telephone network, however its security is still the greatest challenge, and being a real time service, Denial of Service (DoS) attacks are the more effective ones. The Session Initiation Protocol (SIP) has become the main signaling protocol for multimedia sessions in the Internet and IP telephony, and it has been very successful in recent years [1]. SIP servers are vulnerable to a wide variety of DoS attacks; authors in [2] classify them into three different classes, namely: Message Flows Attacks, Malformed Message Attacks, and the Flooding Attacks which is the common, affective and the easiest one [3]. In SIP flooding attacks, the attacker generates a large numbers of SIP requests, the SIP server receives the requests and maintains a transactional state for each one until the transaction completes or the transaction times out. The system is kept busy treating these requests, and the overall performance of the SIP server will decay. On the other hand, Intrusion Detection System (IDS) is a security system that monitors traffic and analyzes that traffic for possible hostile attacks. According to the analyzing method, IDS is classified into Misuse Detection Systems and Anomaly Detection Systems. Misuse detection approaches attempt to model attacks on a system as specific patterns, and then systematically scan the system for occurrences of these patterns. By contrast, anomaly detection approaches attempt to detect intrusions by noting significant departures from a normal behaviour [4]. Most of flooding attacks detection systems are anomaly based, their normal traffic models are mainly based on flow rates [3], Adaptive Threshold and Cumulative Sum (CUSUM) are the two common detection algorithms which belong to this category, they were used in [5] to detect SYN flooding, and in [6] to detect SIP flooding attacks. The other DoS detecting systems utilize the normal protocol behavior, Non-parametric Cumulative Sum (NP-CUSUM), and Hellinger Distance (HD) are two behavior-dependent detection algorithms, they were used in [3,7] and [4] respectively to detect SYN flooding, while they were used in [8] and [9] to detect SIP flooding attacks. Work which was done in [6] shows that HD is able to detected different types of SIP flooding attacks accurately, whereas the Adaptive Threshold and the Cumulative Sum cannot optimally detect different types of SIP flooding attacks using the same set of parameters value. Non-parametric Cumulative Sum also was investigated in [8] and considered as high detection accuracy and low complexity algorithm. We believe that the investigations which were done on these algorithms are not enough, especially when SIP flooding attacks spread over a wide range of request rates. In this paper we present a deeper analysis for them, identifying critical weak points for each one. In the following study, to make sure the wide range of SIP flooding attacks is considered, we classify the flooding attacks into three types: the first is the lower rate at which attack begins to make effect on server performance, it is called Low Rate Attack (LRA), while the second is the rate who guarantees maximum effect in shortest attack time and called High Rate Attack (HRA), and the third is the Medium Rate Attack (MRA) which mediates these two types. The rest of this paper is organized into 5 sections. Sections 2-5 present a detailed description for the four mentioned detection algorithms along with the problems associated with each one, while section 6 is the conclusion and the proposals future work.

2. Adaptive Threshold algorithm:

Adaptive Threshold Algorithm relies on testing whether the average of a given feature in a predefined time window exceeds a particular threshold. If X_n is the value of the feature in the n^{th} time interval, and μ_{n-1} is the average estimated feature from measurements prior to n, then the alarm condition is given by:

If $X_n>(\alpha +1)\ \mu_{n-1}$ then ALARM is signaled at time n. (1)

$\alpha > 0$ is the amplitude factor; that indicates the percentage above the mean value that we consider to be an indication of anomalous behavior. μ_n can be computed using an Exponentially Weighted Moving Average (EWMA) of previous measurements.

$\mu_n=\beta\ \mu_{n-1}+(1-\beta)\ X_n$ where β is the EWMA factor. (2)

Direct application of the above algorithm would yield a large number of false alarms. A simple modification that can improve its performance is to signal an alarm after a minimum number of consecutive violations of the threshold. Adaptive Threshold is applied to detect the SIP flooding attacks by monitoring the rate of SIP requests. Its performance varies significantly with the variation in attack metrics. Previous work on SIP flooding attacks [5,6] shows that Adaptive Threshold algorithm detects high rate attacks relatively accurately with fewer false alarms than that of low rate attacks. Also Adaptive Threshold algorithm performs better for short period attacks as compared to long period ones. The next two subsections show that Adaptive Threshold has two main problems, we call them attack masking and adaptation with attack problems.

2.1. Adaptive Threshold and attack masking problem:

The masking phenomena is related to the capability of attacker to block the server with a preamble of large rate of requests, these intrusive requests can be detected, but its main aim is to raise the detection threshold, creating the opportunity for attacker to inject another lower rate of requests that are not detected by IDS. The Adaptive Threshold alarm condition is given by the inequality (1), so if we raise the threshold where the following condition remains satisfied, the attack will be undetectable.

$X_{Attack}\leq(\alpha +1)\ \mu_{n-1}$ where X_{Attack} is the attack rate

From the equation (2), the difference between μ_{n-1} and μ_n is given as:

$$\mu_n- \mu_{n-1}= (1-\beta)(X_n- \mu_{n-1})$$

If $(X_n>\mu_{n-1})$ the threshold is increased. Attacker now begins to send mask requests (X_{mask}) where $(X_{mask}>\mu_{n-1})$, and continues sending the mask requests tell the following condition is satisfied:

$$\mu_{n-1} \geq \frac{X_{Attack}}{\alpha+1} \qquad (3)$$

It is save now for an attacker to send undetected attack X_{Attack}. Attacker continues attacking the server for any period of time, and the threshold remains high. Figure (1) demonstrates the Adaptive Threshold attack masking problem; it shows how MRA is used to mask LRA.

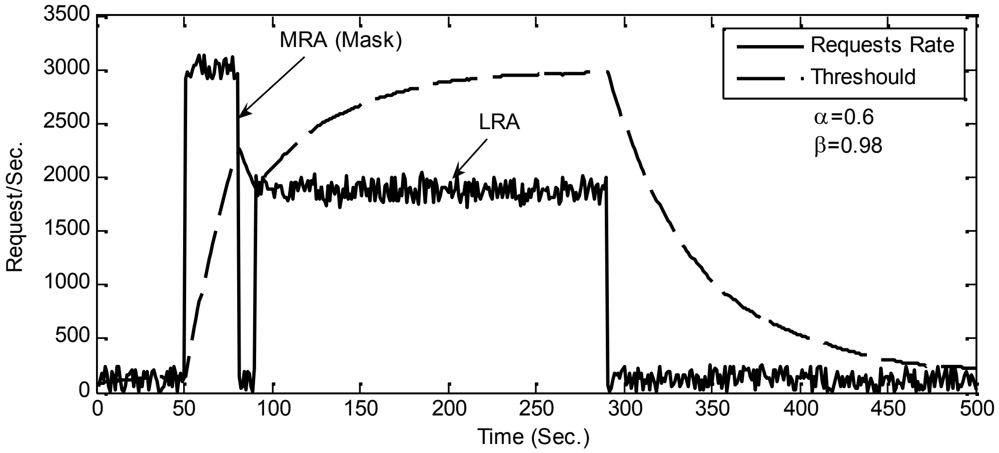


Figure (1): *Adaptive Threshold and attack masking problem*

Higher rate attacks raise the threshold faster than lower rate ones. Attackers try to use high rate attacks for short duration as masks. The EWMA (β) factor determines how much the current sample request rate affects the next estimated mean request rate. Large β means less current rate effect on the next estimated rate, (see equation (2)), and then the attacker needs larger mask volume to raise the threshold. On the other hand, the raised threshold takes longer time to adapt with the next normal request rate, thus attacker can leave more time between the mask and the attack. The amplitude factor α also has a significant effect on the masking phenomena, inequality (3) shows that as α increased as the masking condition satisfied earlier for a given “hidden” attack rate X_{Attack}.

2.2. Adaptive Threshold and adaptation with attack problem:

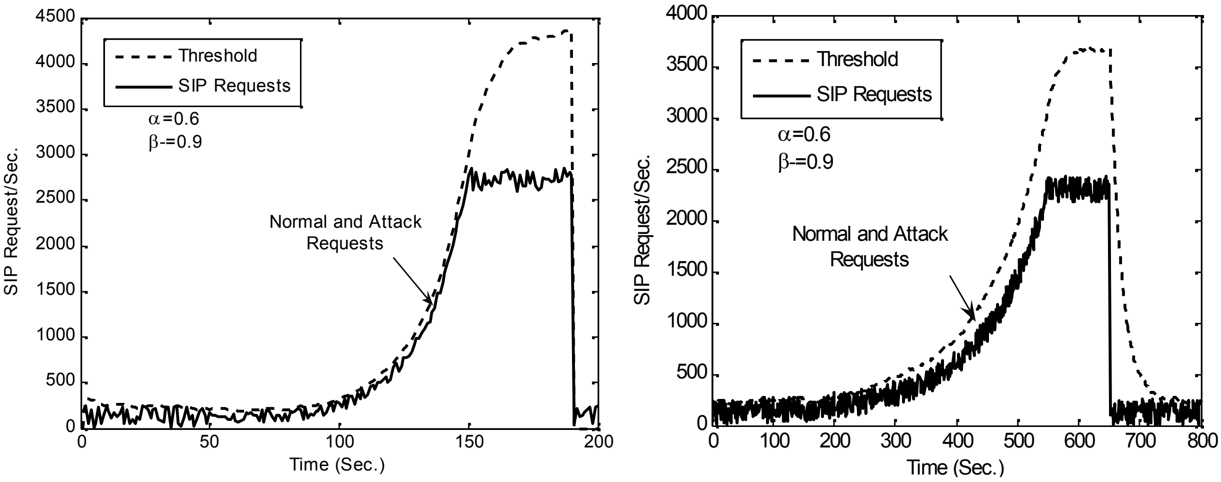
Attacker is not restricted to make a sudden change in the detection threshold. Let us suppose that attacker sends X_{Attack} requests rate which satisfy the following condition:

$$X_{Attack} = (\alpha+1)\,\mu_n$$

According to inequality (1), X_{Attack} is considered as normal request rate. The next estimated mean request rate is given as:

$$\mu_{n+1}=\beta\,\mu_n+(1-\beta)\,X_{Attack}$$
$$\mu_{n+1}=\beta\,\mu_n+(1-\beta)\,(\alpha+1)\,\mu_n$$
$$\mu_{n+1}=\mu_n+\alpha\,(1-\beta)\mu_n \qquad (4)$$

We have 0 < α, β <1, and then according to equation (4) we have $\mu_{n+1}>\mu_n$. The attacker succeeds to raise the mean requests and thereby the detection threshold is raised too. Repeating this scenario raises the detection threshold up to unlimited bound, causing the attack to pass without any noticeable trace. Since attacker raises the request rate up to (α+1) μ_n periodically, both α and β will have significant effect on the attacking process. It is clear that large α values permit large volume of X_{Attack}, and then the attack request rate grow rapidly. The EWMA (β) factor has the opposite effect; large β means less current rate effect on the next estimated rate, so X_{Attack} could be increased by small amount, and then the attack request rate grows slowly. Figure (2-A) shows how attacker raises the detection threshold in such a way that MRA becomes undetectable. Attacker who cannot estimate the μ_n and α may simplify the problem and begins his attack by very low rate, then increases it periodically also by very low request rate too, such a way μ_n increased and thereby threshold increased too. In Figure (2-B) attacker begins his attack only by ten requests per second, then increases it gradually one percent from previous request rate (or one request at least). Figure shows that attacker after a few minutes can pass MRA without any noticeable trace. Here α and β will not have any effect since attacker independently increase his request rate by very small amount of requests. Adaptation with attack will be satisfied, but after longer time other than previous one.



(A) (B)

***Figure (2):** Adaptive Threshold and adaptation with attack problem*

3. Cumulative Sum algorithm:

The CUSUM algorithm belongs to the family of change point detection algorithms that are based on hypothesis testing [10]. It detects the abnormality much faster than the Adaptive Threshold algorithm [11].The choice of CUSUM is based on its simplicity in computation as well as its generally excellent performance [7]. CUSUM was developed for independent and identically distributed random variables $\{x_i\}$. According to the approach, there are two hypothesis θ_0 and θ_1, 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 S_n

$$S_n=\sum_{i=0}^{n} s_i \qquad \text{where} \qquad s_i= \ln \frac{P_{\theta 1}(x_i)}{P_{\theta o}(x_i)}$$

The typical behavior of the log-likelihood ratio S_n includes a negative drift before a change and a positive drift after the change. 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 CUSUM algorithm takes the following form:

If $g_n \geq h$ then an alarm is signaled at time n,

Where: $g_n = S_n - m_n$ and $m_n = \min_{1\leq j\leq n} S_j$, and the parameter h is the threshold.

Applying CUSUM algorithm to detect SIP flooding attack was done in [6]. It was assumed that $\{x_n\}$ are independent Gaussian random variables which represent the SIP requests rates in consecutive time intervals. x_n has known variance σ^2, which is assumed to remain the same after the change, and μ_0 and μ_1 are the means before and after the change, respectively. According to [6] g_n can be formed as:

$$g_n= [g_{n-1}+ \frac{\alpha*\mu_{n-1}}{\sigma^2} (x_n- \mu_{n-1}- \frac{\alpha*\mu_{n-1}}{2})]^+ \qquad (5)$$

Where [x]+ is equal to x if x>0 and 0 otherwise, x_n is the SIP request rate in the nth time interval, α is the amplitude percentage parameter, and μ_n is an estimate mean for SIP request rate at time n, which is computed using the EWMA as follow:

$$\mu_n=\beta\ \mu_{n-1}+(1-\beta)\ x_n \qquad (6)$$

Experimental work which was done in [6] shows that CUSUM has better performance with respect to low rate attacks as compared to high rate attacks. In all cases, CUSUM algorithm, similar to the Adaptive Threshold algorithm, has attack masking and adaptation with attack problems, as will be shown shortly.

3.1. CUSUM and attack masking problem:

Suppose that previous SIP requests are normal, by referring to equation (5) we note that the current attack rate is considered as normal one if the following condition is satisfied:

$$\frac{\alpha*\mu_{n-1}}{\sigma^2}(x_n- \mu_{n-1}- \frac{\alpha*\mu_{n-1}}{2}) \leq 0$$

$$\text{But} \qquad \frac{\alpha*\mu_{n-1}}{\sigma^2} \geq 0, \quad \text{then} \quad (x_n- \mu_{n-1}- \frac{\alpha*\mu_{n-1}}{2}) \leq 0$$

$$x_n \leq \frac{2+\alpha}{2} * \mu_{n-1} \qquad (7)$$

Attacker who plans to mask attack with x_{Attack} request, must increases the mean request μ up to $\frac{2}{2+\alpha}*x_{Attack}$ then sends his flooding requests. As said befor, two attacks are used, the first is the preamble, which is used to incrase the mean request rate, and the second is the intended hidden attack. Figure (3) shows how HRA can be used to mask MRA.

The effect of β is similar to that ones in Adaptive Threshold masking problem. The amplitude factor α also has a significant effect on the masking phenomena, inequality (7) shows that as α increased as the masking condition satisfied earlier for a given “hidden” attack rate X_{Attack}.

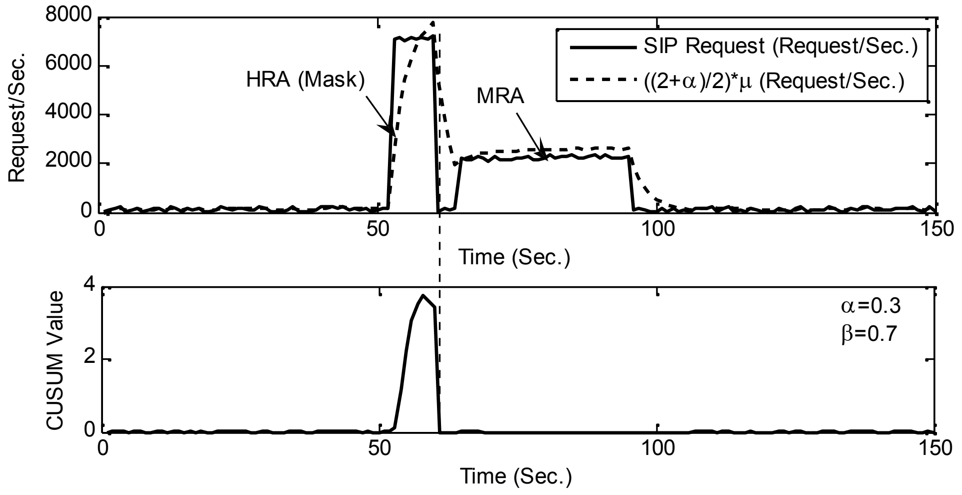
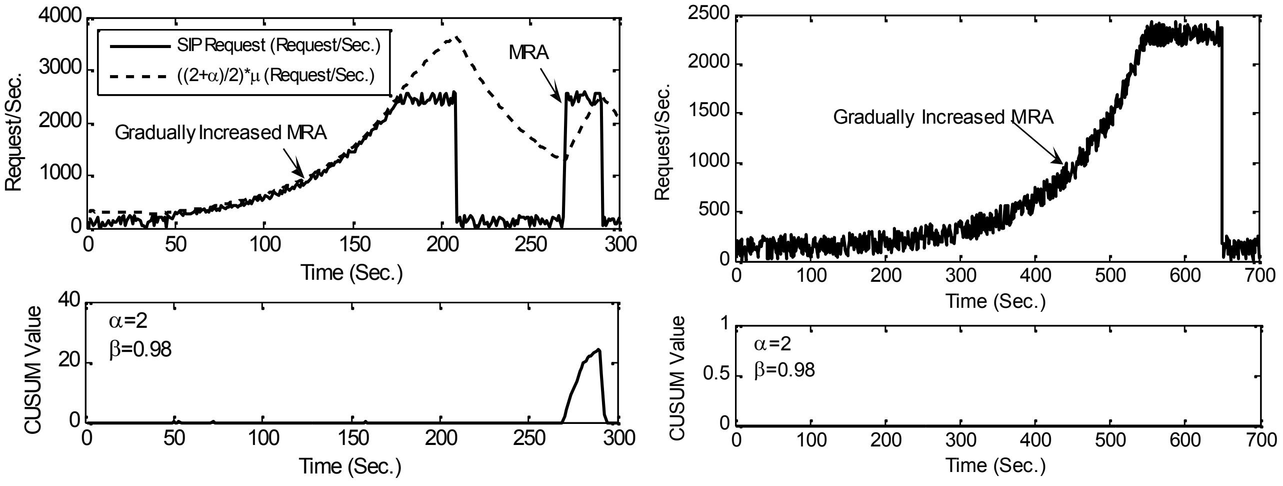


***Figure (3):** CUSUM and attack masking problem*

3.2. CUSUM and adaptation with attack problem:

Clever Attacker may keep himself completely undetectable. According to inequality (7) the requests x_n may be increased up to $\frac{2+\alpha}{2}$ * μ_n-1 and considered as normal. According to equation (6), increasing x_n increases the μ_n, but increasing μ_n in turn adds the opportunity to increases the normal x_{n+1}, and so on; the request rate is increased gradually and the CUSUM algorithm has not any indication about this new state. CUSUM algorithm adapts himself for each new increased request rate, and thus the attack remains undetectable. Figure (4-A) demonstrates how the CUSUM algorithm is not able to detect the gradually increased MRA, while it simply detects the fixed rate one. Again, as we had seen for the Adaptive Threshold, Attacker may begins his attack by very low request rate, then increases it periodically also by very low request rate. In Figure (4-B) attacker begins his attack request only by ten requests per second, then he increases it gradually one percent from previous request rate (or one request at least), Adaptation with attack satisfied but after longer lime.



(A) (B)

***Figure (4):** CUSUM and adaptation with attack problem*

4. Non Parametric Cumulative Sum algorithm:

NP-CUSUM is used to detect DoS attacks when it is not possible to model the total number of session request arrivals by a simple parametric description. NP-CUSUM detection mechanism belongs to change point detection algorithms and based on protocol behavior [3,12]. It was used in [8] to detect SIP flooding attack relaying on the SIP request/response pairs relations. Analysis which was done on SIP server traffics shows that difference in number between INVITE/BYE and 200OK/ACK pairs does not vary too much with the change of time, and have a strong correlation [4,8]. Any deviation from this kind of correlation is considered as an indication of DoS attacks; that is the key idea of NP-CUSUM.

Suppose X is the number of INVITE minus BYE requests. Value of μ is the average number of X. μ can be estimated in real time and updated periodically in the same way we had seen in Adaptive Threshold and CUSUM algorithms, according to the following equation:

$$\mu_n = \beta\, \mu_{n-1} + (1-\beta)\, X_n \quad (8),$$ where β is the EWMA factor.

Let $\{X_n, n = 0,1,...\}$ be the number of INVITE minus that of the corresponding BYE, X_n is dependent on the number of the calls, it may also vary with time. In order to alleviate these dependencies, X_n is normalized by the average number μ, during the sampling period. Define $Z_n = X_n / \mu_n$, value of Z_n is no longer dependent on the network size or time-of-day. Let $E(Z_n) = c$. We choose a parameter that is the upper bound of c, i.e. a >c and define $\tilde{Z}_n = Z_n - a$, so it has a negative mean during normal operation. When an attack takes place, $\tilde{Z}_n$ will suddenly become positive.

Let $$y_n = [y_{n-1} + \tilde{Z}_n]^+ \quad (9)$$
$$y_0 = 0;$$

Where $[x]^+$ is equal to x if x > 0, and 0 otherwise. Large y_n is a strong indication of an attack. IDS developer may apply the same detection process for 200OK/ACK pair, then it can be used as standalone flooding attack indication, or it may be combined with INVITE/BYE detection to make a robust decision.

4.1. NP-CUSUM and attack masking problem:

NP-CUSUM considers the current SIP request sample as normal one if (X_n / μ_n) is small enough, and does not exceed the upper bound of its expected value (a). In normal traffic this value is small since the difference between INVITE and BYE rate is small. Attacker, also, may preserve it small if he succeeds to enlarge the estimated difference (μ_n). Equation (8) indicates that one way to enlarge μ_n is to enlarge X_n, but this considered as attack, equation (8) also indicates that β amount of the current raised μ_n will be preserved in the next estimated one, and then μ is preserved large for the next few samples. Consequently, attacker sends some flooding requests as preamble to raise μ_n, this raised μ_n gives the attacker adequate time to send undetected flooding attack, as seen Figure (5).

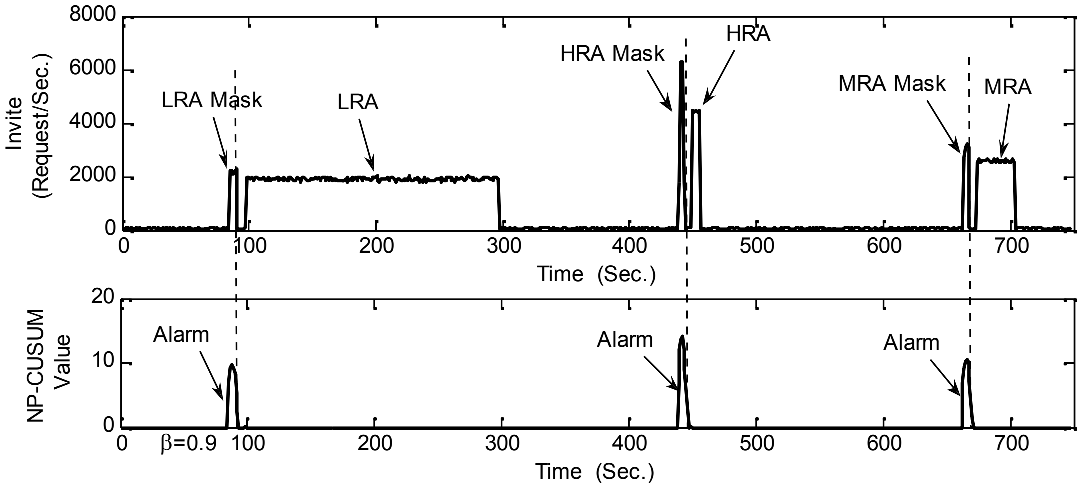


Figure (5): *NP-CUSUM and attack masking problem*

4.2. NP-CUSUM and adaptation with attack problem:

Let us suppose that Invite_n, Bye_n are the numbers of INVITE and BYE requests consequently at instant n, and the previous requests are normal, so according to equation (9), the current requests will be considered normal if:

$$\tilde{Z}_n \leq 0$$

$$(x_n / \mu_n) - a \leq 0$$

$$((\text{Invite}_n - \text{Bye}_n) / \mu_n) \leq a$$

$$\text{Invite}_n \leq \text{Bye}_n + \mu_n * a \qquad (10)$$

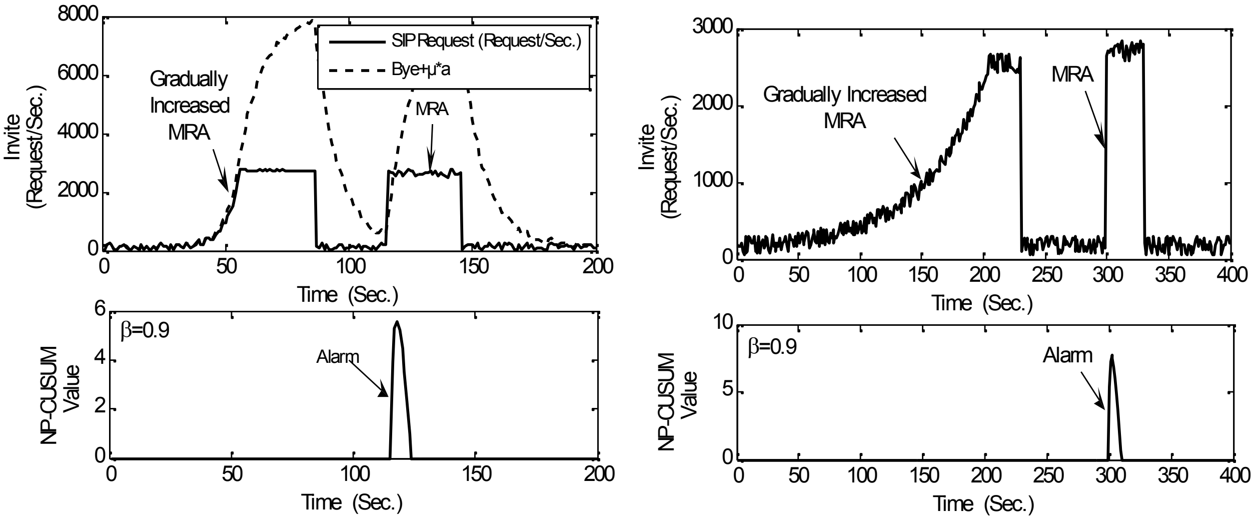
So, the attacker can increase the current Invite requests up to ($\text{Bye}_n + \mu_n * a$) and considered as normal request. From equation (8) we realize that μ_{n+1} will be increased too, and then the next INVITE request rate (Invite_{n+1}) can safely be increased up to $\text{Bye}_{n+1}+\mu_{n+1}*a$. Attacker repeats this scenario and gradually increases the INVITE request rate, whereas the NP-CUSUM algorithm modifies his estimated difference to be adapted with the new situation. Attacker may decide to increase the request rate up to upper limit then continue sending a fixed rate of flooding requests. Figure (6-A) illustrates this case. Figure (6-B) shows simple gradually increased attack, where attacker begins his attack request only by fifty INVITE requests per second, then he increases it gradually two percent each time.

4.3. NP-CUSUM and request balancing problem:

NP-CUSUM algorithm suffer another fatal bug, attacker can con the algorithm. According to the inequality (10), the sample n is considered as normal if the following condition is satisfied:

$$\text{Bye}_n \geq \text{Invite}_n - a * \mu_n$$

Thus attacker can send any arbitrary number of INVITE requests then balances them by sending BYE requests which satisfy the inequality, and the attack remains undetected. Simply, attacker may sends the same number of INVITE and BYE requests simultaneously to remain undetected.



(A) (B)

Figure (6): *NP-CUSUM and adaptation with attack problem*

5. <u>Hellinger Distance:</u>

HD measures the deviation between probability measures that does not make any assumptions about the distributions themselves. To explain, let P and Q be two probability distributions on a finite sample space Ω, where P and Q on Ω are N-tuples (p_1, p_2, ..., p_N) and (q_1, q_2, ..., q_N) respectively, satisfying inequalities p_i≥0, q_i≥0, $\sum_{i=1}^{N} p_i$=1 ,$\sum_{i=1}^{N} q_i$=1, Then, the HD between P and Q is defined as [9]:

$$d_H^2(\mathrm{P,Q}) = \frac{1}{2}\sum_{i=1}^{N}(\sqrt{p_i} - \sqrt{q_i})^2$$

Sometimes, the factor 1/2 is not used. The HD satisfies the inequality $0 \leq \mathrm{d}_\mathrm{H}^2 \leq 1$ and d_H^2 =0 when P=Q. Disjoint P and Q shows the maximum distance of 1if factor $\frac{1}{2}$ is omitted, if the observed HD is greater than a predefined threshold value then an alarm is raised. Authors of [9] had use HD to detect anomalies in SIP protocol. They took four attributes of SIP which are the number of INVITE, 200 OK, ACK and BYE messages arrived in a predefined time-window. The algorithm consists of training and testing phases. In the training phase, the normalized frequencies $\mathrm{p}_{\mathrm{INVITE}}$, $\mathrm{p}_{\mathrm{200OK}}$, $\mathrm{p}_{\mathrm{ACK}}$, $\mathrm{p}_{\mathrm{BYE}}$ for INVITE, 200OK, ACK and BYE respectively are calculated over the training dataset. Similarly, the normalized frequencies $\mathrm{q}_{\mathrm{INVITE}}$, $\mathrm{q}_{\mathrm{200OK}}$, $\mathrm{q}_{\mathrm{ACK}}$, $\mathrm{q}_{\mathrm{BYE}}$ are calculated in the testing phase for each time window n. The HD between these frequency distributions of two phases is:

$$\mathrm{HD} = (\sqrt{p_{INVITE}} - \sqrt{q_{INVITE}})^2 + (\sqrt{p_{2000K}} - \sqrt{q_{2000K}})^2$$
$$+ (\sqrt{p_{ACK}} - \sqrt{q_{ACK}})^2 + (\sqrt{p_{BYE}} - \sqrt{q_{BYE}})^2$$

To keep track of the normal attribute behaviors, authors in [9] and [6] 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 α and mean deviation ν, given measurement HD, are computed as:

$$\text{Let} \quad Err = HD_n - \alpha_{n-1} \qquad (11)$$

$$\alpha_n = \alpha_{n-1} + g * Err \qquad (12)$$

$$\nu_n = \nu_{n-1} + h*(|Err| - \nu_{n-1}) \qquad (13)$$

where HD_n is the HD for the current sample, $\alpha_{n-1}$and α_n are the previous and current estimated average HD, respectively, ν_{n-1} and ν_n represent the previous and current mean deviations, g and h are chosen to be negative exponents of 2.

The estimated average HD (α_n) is based on the observed HD, which is measured between the probability measures P and Q. During the testing periods, the Threshold (TH) is estimated using the estimated average HD (equation 12) and the mean deviation (equation 13).

$$TH_n = x_* \alpha_n + y_* \nu_n \qquad (14)$$

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. The test which was done in [6,9] shows that HD is more accurate and has fewer false alarms, in detecting high rate attacks more than low rate ones. Here, we limit our study to two common requests, which are INVITE and BYE. This restriction will not affect the essence of HD, since it is a simple polynomial of positive values. Suppose there are N_{INVITE}, N_{BYE} of INVITE and BYE requests respectively, during the training period then the normalized frequency of p_{INVITE} and p_{BYE} over the training data set are defined as follows:

$$p_{INVITE} = \frac{N_{INVITE}}{N_{INVITE}+N_{BYE}} \qquad p_{BYE} = \frac{N_{BYE}}{N_{INVITE}+N_{BYE}}$$

Also, suppose that there are M_{INVITE}, M_{BYE} of INVITE and BYE requests respectively, during the testing period, then the normalized frequency of q_{INVITE} and q_{BYE} over the testing data set are defined as follows:

$$q_{INVITE} = \frac{M_{INVITE}}{M_{INVITE}+M_{BYE}} \qquad q_{BYE} = \frac{M_{BYE}}{M_{INVITE}+M_{BYE}}$$

The HD between the normalized frequencies of the training and testing data set is computed as follows:

$$HD = (\sqrt{p_{INVITE}} - \sqrt{q_{INVITE}})^2 + (\sqrt{p_{BYE}} - \sqrt{q_{BYE}})^2$$

In the following we will show that HD suffers from attack masking, adaptation with attack, and request balancing problems.

5.1. HD and the attack masking problem:

HD has the attack masking problem; to explain it we rearrange the previously defined equations. By merging equation (11) and equation (12), α_n is expressed as:

$$\alpha_n = (1-g)\,\alpha_{n-1} + g * HD_n \quad (15)$$

Equation (15) shows that (1-g) of the estimated average HD is obtained from the previous one. Now, by substituting the estimated average HD in equation (14), by the obtained one in the equation (15), equation (14) can be rewritten as:

$$TH_n = x_* (1\text{-}g)\, \alpha_{n\text{-}1} + x_* g_* HD_n + y_* v_n$$

This equation shows that $x_*(1\text{-}g)$ of the previous estimated average HD, is preserved in the current threshold value. The first multiplication factor (x) is chosen large enough to reduce the false alarm rate [9], and g, as defined before, is chosen to be negative exponents of 2, thus the preserved value is significant. Consequently, if the attacker succeeds to raise the estimated average HD, the threshold will be raised too and kept high for some time, and then the attacker has the opportunity to send undetected attack. Figure (7) demonstrates how attacker can use this bug to mask different types of SIP flooding attacks.

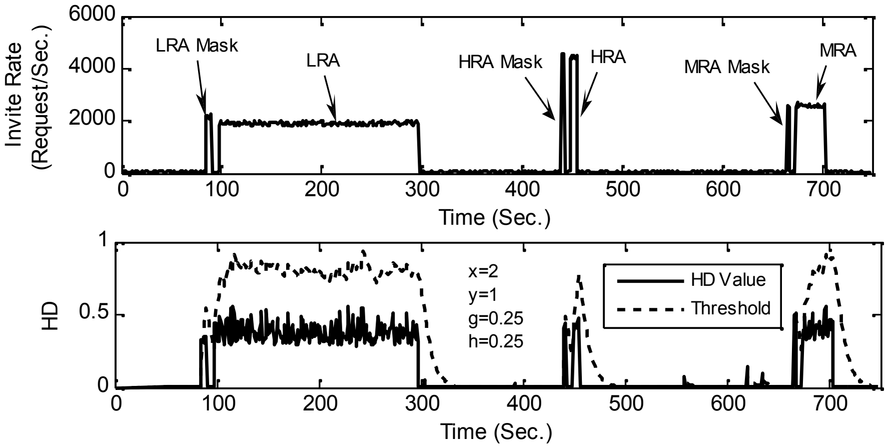
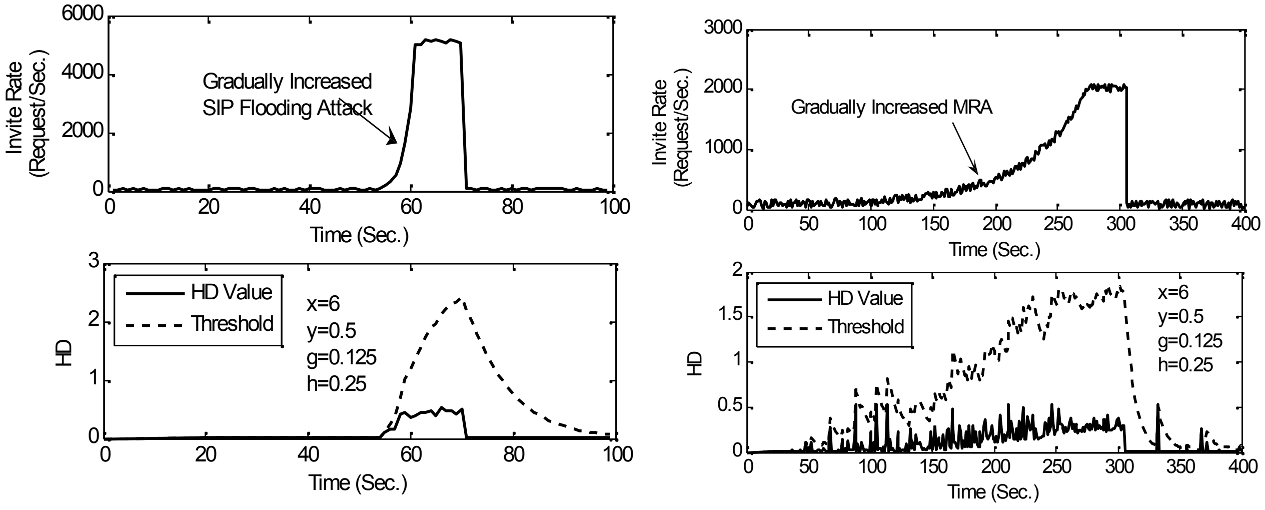


Figure (7): *HD and attack masking problem*

Attacker sends the mask as a preamble; its main aim is to raise the threshold. The detection algorithm will be able to detect the mask, but the intended attack remains undetectable.

5.2. HD and adaptation with attack problem:

The threshold value in equation (14) permits to the current INVITE requests to be increased in such a way that HD_n becomes close to $x_* \alpha_n$ and kept undetectable, we must note here that α_n is the estimated average HD for the current sample. This margin is big enough to gradually increase the detection threshold, attacker may get aware of this fact and begins to abuse it as follow: Increasing the current HD_n will increase the Err value; see equation (11), and then α_n will be increased too, (equation (12)). Consequently, the TH_n will get higher. Attacker repeats this scenario and the detection threshold increased gradually. When the threshold becomes high enough, attacker continues sending the flooding requests. Figure (8-A) shows how quickly attacker can increase the HD threshold to pass the attack. Figure (8-B) shows simple adaptation with attack where the attacker begins by ten requests per second then increases it gradually two percent each time.



(A) (B)

Figure (8): *HD and adaptation with attack problem*

5.3. <u>HD request balancing problem:</u>

Again, attacker easily defeats the HD algorithm. He can hide himself by sending equal amounts of INVITE and BYE requests. Let us assume the current system situation is normal, attacker sends a large and equal number of INVITE and BYE requests. This means M_{INVITE} is nearly equal to M_{BYE} and then q_{INVITE} and q_{BYE} will be adjacent to $\frac{1}{2}$, it looks like the situation is normal, then p_{INVITE} and p_{BYE} also are adjacent to $\frac{1}{2}$, thus the HD value will be close to zero, and no attack detection occur.

6. <u>Conclusion and future work:</u>

The analytical and experimental work which is performed in this paper demonstrates that the studied four flooding attacks detection algorithms have two common problems, the attack masking and adaptation with attack problems. These problems resulted from the memorized quantity which preserved by the algorithms to adapt with the new request rate. In the first problem, attacker makes a sudden change in the detection algorithm parameters by sending an attack mask, which consists of adequate quantity of requests. This sudden change is detected as attack, but before the algorithm parameters turn again to their average normal values, attacker sends the intended attack and it is kept undetected. In the adaptation with attack problem, attacker gradually increases the attack requests in such a way that the increased amount remains close to the upper normal limit. Consequently, the algorithm parameters are deviated gradually to adapt with the new normal increased requests. The paper also demonstrates that anomaly detection algorithms, which utilize the protocol behavior, are simple to be defeated, and the only thing attacker required to do is to consider the same protocol behavior during the attacking process.

The subject of future work is to build a new flooding attack detection algorithm that overcome the attack masking and adaptation with attack problems, and is also independent from protocol behavior.

Refrences:

[1] V. Hilt and I. Widjaja, Controlling Overload in Networks of SIP Servers, IEEE International Conference on Network Protocols, pp. 83-93, 2008.

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

[3] 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.

[4] H. Sengar, H. Wang, D. Wijesekera, and S. Jajodia, Fast Detection of Denial-of-Service Attacks on IP Telephony, fourteenth IEEE International Workshop on Quality of Service, New Haven, June 2006.

[5] V. Siris and F. Papagalou, Application of Anomaly Detection Algorithms for Detecting SYN Flooding Attacks, Computer Communications, vol. 29, no. 9, pp. 1433–1442, 2006.

[6] 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.

[7] 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.

[8] H. Zhang, Z. Gu1, C. Liu1, and T.Jie1, Detecting VoIP-specific Denial-of-Service Using Change-Point Method, 11th International Conference on Advanced Communication, February, 2009.

[9] H. Sengar, H. Wang, D. Wijesekera, and S. Jajodia, Detecting VoIP Floods using the Hellinger Distance, IEEE Transactions on Parallel, and Distributed Systems, vol. 19, no. 6, pp. 794-805, June 2008.

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

[11] B. Rozovskii, A. Tartakovsky, R. Blaˇzek, 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.

[12] A.G. Tartakovsky, K. Shah, and B.L. Rozovskii, A nonparametric multichart CUSUM test for rapid intrusion detection, in Proceeding of Joint Statistical Meetings, Minneapolis, August 2005.