ON THE DESIGN OF A LOW-RATE DOS ATTACK AGAINST ITERATIVE SERVERS

Gabriel Maciá-Fernández, Jesús E. Díaz-Verdejo, Pedro García-Teodoro
Dep. of Signal Theory, Telematics and Communications - University of Granada
c/ Daniel Saucedo Aranda, s/n - 18071 - Granada (Spain)
{*gmacia,jedv,pgteodor*}*@ugr.es*

Keywords: network security, denial of service, attack modelling, low-rate attack.

Abstract: Recent research exposes the vulnerability of current networked applications to a family of low-rate DoS attacks based on timing mechanisms. A kind of those attacks is targeted against iterative servers and employs an ON/OFF scheme to send attack packets during the chosen critical periods. The overall behaviour of the attack is well known and its effectiveness has been demonstrated in previous works. Nevertheless, it is possible to achieve a trade off between the performance of the attack and its detectability. This can be done by tuning some parameters of the attack waveform according to the needs of the attacker and the deployed detection mechanisms. In this paper, a mathematical model for the relationship among those parameters and their impact in the performance of the attack is evaluated. The main goal of the model is to provide a better understanding of the dynamics of the attack, which is explored through simulation. The results obtained point out the model as accurate, thus providing a framework feasible to be used to tune the attack.

1 INTRODUCTION

Nowadays, networked systems represent a vital infrastructure involved in a lot of everyday activities. As the dependence on those systems increases, the risks of compromising the network itself or the services provided become more threatening. In this context, one of the most devastating menaces are the so called Denial of Service (DoS) attacks (CERT Coordination Center, 2003). Recent incidents involving companies with big presence in Internet (Williams, 2000) guarantee the existence of this threat while, at the same time, demonstrate its potential impact.

The vulnerabilities and mechanisms enabling DoS attacks are of diverse nature (Axelsson, 2000), although two main families of attacks can be established. The first one is composed by software or hardware vulnerabilities that can be exploited by a suitable piece of software causing the stopage of the targeted system, service or network facility. This kind of DoS attacks is very similar to many other attacks (e.g. virus-based attacks) and so are the defence mechanisms. The second kind is based on saturating some necessary resource at the target by means of flooding. Due to the high number of requests usually needed to flood the resource, these kind of attacks are commonly called brute-force attacks (Douligeris, 2004). Anyway, recent research (Kuzmanovic, 2003) (Maciá-Fernández et al., 2006b) points out the existence of a more subtle kind of DoS attacks characterized by the use of a relatively low-rate of request packets to achieve the flood of the resource. The present paper is focused on exploring one of these attacks, specifically (Maciá-Fernández et al., 2006b), which will be called MF06 from now on.

On the other hand, the development and deployment of countermeasures does not completely eliminate the threat. The main defence mechanisms are solely based on preventive measures such as ingress filtering (Ferguson, 2001), egress filtering (SANS Institute, 2000), or disabling unused services (Geng, 2000), just to mention some of them. Some works propose to use honeypots (Weiler, 2002) and intrusion detection systems (IDS) (Axelsson, 2000) to handle an in-progress attack. In this sense, IDS' have proven to be a useful tool to detect brute-force attacks through the assessment of the volume of traffic. Anyway, this is an area of active research. Furthermore, the defence mechanisms against the above mentioned low-rate attacks are also under research. Thus, some solutions have been proposed for the TCP attack (Sun et al., 2004), (Shevtekar et al., 2005), although they

are specific for this kind of attack. More research should be made in order to capture the behavior of the attack in MF06 to develop protection mechanisms.

Before a deeper insight into MF06 is made in the following sections, some comments concerning the two above mentioned low-rate attacks should be pointed out. First, both attacks are founded on a predictable temporary behavior enabling an intelligent selection for the timings of the attack packets. Therefore, an ON/OFF attack waveform is used in both cases, which reduces the rate of packets needed to carry out the attack. But there ends the similarities. The temporary mechanisms exploited are clearly different in both cases: while (Kuzmanovic, 2003) exploits the TCP congestion control mechanism, MF06 exploits some knowledge concerning the inter-output time between responses in a service. This way, even the attacked resources are different: the links' capacity in the network and the incoming requests queue, respectively. Moreover, the levels at which the attack is carried out are different: transport/application layer, and so are their impacts: global to the targeted network or local to the targeted service.

One of the most interesting characteristics of the MF06 attack is related to its versatility. By adjusting some attack parameters that will be described later, it is possible to achieve a compromise between the effectiveness of the attack, in the sense of the level of denial of service achieved, and the rate of traffic generated by the attack. This property would allow a potential intruder to select the optimum parameters in order not to be detected by an IDS system based on some rate threshold mechanism. The purpose of this work is to study how these adjustments affect the behavior of the attack and, therefore, how to optimize the design the attack. This knowledge can lead to the development of new defence mechanisms. For this purpose, the authors propose to use and evaluate a recently developed mathematical model for the cited attack.

The rest of the article is structured as follows. A brief description of the low-rate DoS attack is reviewed in Section 2. Section 3 presents an overview of the mathematical model that supports the design of the attack. Some conclusions extracted from the model and concerning the behavior of the attack are compiled in Section 4, while Section 5 describes the simulations made to validate the model. Finally, Section 6 presents the conclusions of this work.

2 LOW-RATE DOS ATTACK

The low-rate DoS attack under consideration is targeted against an iterative server and uses certain *a priori* knowledge concerning the time between responses from the server. From the statistics of the so called inter-output time and the observation of the responses from the server it is possible to infer when the next output is likely to be generated. Therefore, the aim of the attack, when in a stationary stage with the server at plenty of its capacity, will be to replace the request being served, either legitimate or malicious, with a new malicious request from the intruder by timing it appropriately. Some details about the operation of the attack and the targeted scenario will be described next.

Let us consider an standard iterative server in which, as usual, requests are queued up in a finite length queue while awaiting for its processing in a FIFO discipline. Similarly, let us suppose a request arriving at the server at a given time, t. The behaviour of the service, related to that petition, can be described as follows. First, if there exists at least one free position in the input queue, the request is queued. Otherwise, the request is not accepted. Whether a reject message is sent back to the requester or not is irrelevant to our study. Next, after some queuing time, t_q, the request will be the first in the queue and, therefore, will be processed by the server during a service time, t_s. Finally, at $t + t_q + t_s$, a response is provided and sent back to the requester.

Up to now, just the standard behavior of an iterative server has been depicted. But, although both the service time and the queue time are random variables, some predictable timing can be expected under controlled circumstances. Thus, if an intruder manages to always request the same resource at the server, it it expectable for the service time, t_s, to be always identical. On the other hand, the time between two consecutive outputs, under the single condition of the existence of at least one pending request in the input queue, is directly the service time. Therefore, the inter-output time, τ_{int}, for the server is predictable and always has the same value. Nevertheless, even in this scenario, some variability in the inter-output time appears due to several reasons related to the functioning of the server and the machine it is running on (e.g. multithread operation, random access times and so on). To account for this variability, the intruder should determine an statistical characterization for the inter-output by sending some requests and observing the behavior of the system. This can be easily done in a non-intrusive way.

In this environment, the attack consists of the iteration of a basic period composed by a period of inactivity, called *offtime* followed by a period of activity, called *ontime* (Fig. 1). During ontime, the attacker iteratively sends the same request in the hope that at least one of them acquire a free position in the input queue. The task of the intruder is to forecast the instant at which a free position is going to be generated, which is, when an output is to be emitted, and to syn-

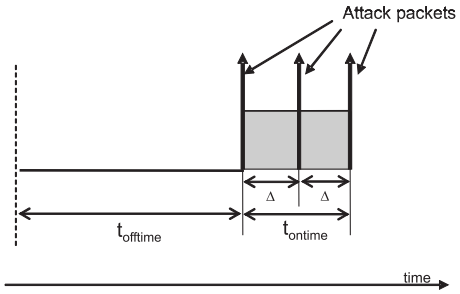

Figure 1: Attack waveform as generated by the intruder.

chronize the attack in such a way that an attack packet reaches the server in the minimum possible time after the output occurs. In an optimal situation, just a single attack packet per ontime period is necessary. Nevertheless, the above mentioned variability in the inter-output time makes advisable to use more attack packets per period. The low-rate nature of the attack relies in the fact that only a small number of attack packets are sent during each ontime period. Although not mentioned, the round trip time between the server and the intruder plays and additional role in this attack. The details are shown in MF06 and exceeds the scope of this paper.

2.1 Attack Parameters

The attack can be characterized by the following parameters (Fig. 1:

- *Ontime interval* (t_{ontime}): activity interval during which an attempt to seizure a free position in the service queue is made by emitting request packets.
- *Offtime interval* ($t_{offtime}$): inactivity interval previous to *ontime* in the period of attack, during which no attack packets are transmitted.
- *Interval* (Δ): the time elapsed between the sending of two consecutive packets during *ontime*.

2.2 Performance Indicators

The performance of the attack can be measured in relation to the level of denial of service achieved, which is the direct objective of the attack, and the number of attack packets that have to be send in order to get the free positions in the queue as they become available. Both aspects can be established through the following indicators:

- *Effort* (E): ratio between the traffic rate generated by the intruder and the maximum traffic rate accepted by the server (server capacity).
- *User perceived performance* (UPP): ratio between the number of legitimate users requests processed by the server, and the total number of requests sent by them.

Although UPP can be measured in a controlled environment, it is difficult to estimate it in a real one. Nevertheless, it is possible to derive UPP from a parameter that can be defined in an ideal situation and that gives an estimate of the time during which the service is vulnerable. This parameter, called *mean idle time* can be defined as follows:

- *Mean idle time* ($\overline{T}_{idle}$): the percentage of time during which the system has free positions in the service queue, related to the total duration of the attack when legitimate users send no traffic.

From the very definition, it is almost obvious that the higher the value of the *mean idle time*, the higher should be the UPP, as the service is available during more time for the use by a legitimate user. Therefore, the objective of the attack should be to reduce as much as possible the value of the *mean idle time* but with as lowest *effort* as possible.

3 ATTACK MODELLING

As indicated above, there is a need to find a relation between the specific settings of the parameters that define the attack (see Section 2), and the values for the indicators that these configurations produce. To address this problem, a mathematical framework proposed by the authors (Maciá-Fernández et al., 2006a) is briefly described. It allows to estimate, for a given server and attack characteristics, the effort and the efficiency of the attack.

The model is described as a set of functions that provide the values for the *mean idle time*, the *user perceived performance*, and the *effort* of the attack from a set of input values. These inputs come from two sources: the statistics for the behaviour of the server/network, and the parameters of the attack. The first category accounts for the probability function of the occurrence of an output in the server and the mean round trip time. The second consider the *ontime period*, t_{ontime}, the *offtime period*, $t_{offtime}$, and the *interval*, Δ.

The first step in developing the mathematical model is the evaluation of the *mean idle time*. Once a function for this parameter is deducted, it is possible to derive some expressions for the *user perceived performance* and the *effort* of the attack by making some assumptions about the behaviour of the server and the attack.

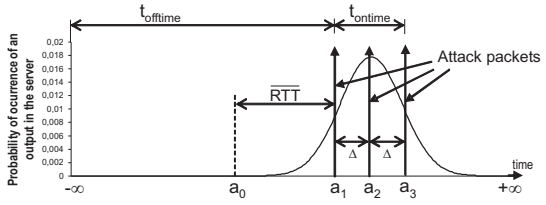

Figure 2: Diagram of occurrence for an output: probability function and associated calculation points. In this example, only three attack packets are considered and a normal distribution is assumed.

3.1 Model for the Idle Time

As indicated in Section 2, the strategy of the intruder is based on synchronizing the arrival of the *ontime interval* at the server with the occurrence of an output, event that generates a free position in the service queue. Doing this, the time during which this position is available for legitimate users ($\overline{T}_{idle}$) is reduced and, therefore, the denial of the service (measured by UPP) would be more annoying.

The intruder carries out a period of attack every estimated inter-output time, τ_{int}. The time between two consecutive outputs is considered as an stochastic process that can be modelled by a distribution. As the *mean idle time* is defined as a percentage of time during which at least a free position is available, for the purpose of developing a mathematical model that evaluates this indicator, an observation period should be defined. A period of an attack, that is, an *offtime* interval followed by an activity interval (*ontime*) will be the observation period. Moreover, it is considered that the service queue is full of requests at the beginning of this period.

Let us consider one attack period and relate it with the probability of generation of an output at the server. From the attack design, it is expected that the maximum probability of generation will occur at the same time that the central attack packet arrives at the server (Fig. 2). In the figure, the instants for the arrival of attack packets (during the *ontime interval*) are represented by vertical arrows. These arrivals occur at the instants labeled $a_i (i \geq 1)$.

On the other hand, and according to the definition, the *idle time* for this period will be the time elapsed between the actual generation of the output and the reception of the next incoming attack packet. To evaluate the *mean idle time* it is necessary to average this value over the whole period, taking into account both the probability of generation of an output and the period till the next incoming packet. For this purpose, a set of so called *calculation points*, $\mathcal{A}$, is defined. The *calculation points* delimit a set of intervals that will be used to calculate the instantaneous values of the idle time, T_i. This set is composed by the instants at which an attack packet arrives at the server, $a_i, 1 \leq i \leq n, n = ceil[t_{ontime}/\Delta]$, and a special point, a_0, which is situated a time $\overline{RTT}$ before the reception of the first attack packet in the observation period, that is,

$$a_0 = a_1 - \overline{RTT} \tag{1}$$

If an output is generated at an instant t, and assuming that

$$\Delta = a_i - a_{i-1} \leq \overline{RTT} \tag{2}$$

the associated *idle time* is

$$T_i(t) = \begin{cases} \overline{RTT} & \text{if } t < a_0 \text{ or } t \geq a_n \\ a_i - t & otherwise, 0 \leq i \leq n \end{cases} \tag{3}$$

The reasoning for Eq. 3 is as follows. If an output is generated before a_0 of after a_n, the time during which a queue position is available, in absence of user traffic, will be $\overline{RTT}$, as an attack packet will (automatically) be emitted by the intruder upon the reception of the output. Thus, the timeline will be as follows. At time t, the output is generated at the server, at time $t + \overline{RTT}/2$, this output is received by the intruder, which sends a new attack packet as response, at time $t + \overline{RTT}$ the attack packet is received at the server, and the *idle time* ends. It is important to note that we are assuming the RTT to be always its mean value. This is appropriate as the final objective is to evaluate the mean idle time. On the other hand, if an output is generated by the server at any other instant, the next scheduled attack packet will be received by the server before the automatically generated response if and only if the round trip time is lower than the *interval*. In this case, the instantaneous *idle time* is the time till the next scheduled attack packet. Otherwise, the *idle time* is again $\overline{RTT}$. Nevertheless, this case is not relevant in most situations as will be explained later.

Thus, for the case in which $\Delta \leq \overline{RTT}$, the *mean idle time* in a period of attack can be obtained through integration as:

$$\begin{aligned} \overline{T}_{idle} &= \frac{1}{T_p} \cdot \Bigg[\int_{-\infty}^{a_0} \overline{RTT} \cdot f(t)dt \\ &+ \int_{a_0}^{a_1} (a_1 - t) \cdot f(t)dt \\ &+ \ldots + \int_{a_{n-1}}^{a_n} (a_n - t) \cdot f(t)dt \\ &+ \int_{a_n}^{\infty} \overline{RTT} \cdot f(t)dt \Bigg] \end{aligned} \tag{4}$$

where $f(t)$ is the probability function for the generation of an output at the instant t and T_p is the duration of a period of the attack, that is, $T_p = t_{offtime} + t_{ontime}$.

The assumed distribution for $f(t)$ will affect to the values obtained from the model, although the model itself is independent of the proposed distribution.

Eq. (4) have been deducted under the condition that the value of Δ should be low enough to accomplish the condition in Eq. (2). However, the model could be easily adapted to the opposite condition, that is, $\Delta > \overline{RTT}$. Nevertheless, this case is out of the scope of this work due to the fact that a good attack design should preserve the case in Eq. (2). If this condition is not meet, the attack would mainly become an almost naive attack in which a request is made every time a response from the server is received. This way, the ON/OFF behavior of the attack waveform will become useless and an important increase in the number of total attack packets will appear.

3.2 Model for the User Perceived Performance

The *user perceived performance* (UPP) is an indicator defined for a system where both legitimate users and the attacker are simultaneously trying to access to the target system. According to MF06, the packet arrivals from the legitimate user are modeled by a Poisson distribution. This implies that the probability of packet reception from a legitimate user during a period of time T is given by the exponential distribution function:

$$F(T) = 1 - e^{-\lambda T} \tag{5}$$

where λ is the arrival rate of packets from the legitimate users.

The UPP can be evaluated by estimating the probability for a legitimate user to queue its request during a period of the attack, P_u^k. This probability can be obtained through the *mean idle time*, that is, the more time a position is free the more probability of a capture by the user. According to the proposed distribution, the probability of an user capture in the k-th interval (Fig. 2), that is (a_{k-1}, a_k), denoted as P_u^k, is given by the expression:

$$P_u^k = 1 - e^{-\lambda T_{idle}^k} \tag{6}$$

where T_{idle}^k represents the generated *idle time* during the k-th interval, that is:

$$T_{idle}^k = \frac{1}{a_k - a_{k-1}} \int_{a_{k-1}}^{a_k} T_i^{(a_{k-1}, a_k)} f(t) dt \tag{7}$$

By summing up the probability of capture for all the intervals, the total probability for a legitimate user to seize a position in the service queue for a complete period of the attack, P_u, is

$$P_u = \sum_{k=0}^{n+1} \left(1 - e^{-\lambda T_{idle}^k}\right) \tag{8}$$

where n is the index of the last calculation point.

The Eq. (7) does not consider the presence of traffic coming from legitimate users. Some approximations have to be made in order to take into account users' traffic. The details and justification for them exceeds the purposes of this paper and can be found in (Maciá-Fernández et al., 2006a). On the other hand, the experimental results shown in Section 5 confirm the goodness of these approximations. Anyway, a set of equations relating P_u and the *mean idle time* can be derived from those reasonings, yielding

$$\begin{aligned} T_i(t) &= \overline{RTT} \cdot (1 - P_u) \\ &+ min\left[\frac{1}{\lambda}, \overline{t_s} - t_{ontime}\right] \cdot P_u \end{aligned} \tag{9}$$

The calculation of the expressions for T_{idle}^k and P_u should be made recursively, due to the fact that there is a cross-dependency between them.

Once a stable value for P_u is obtained, the final expression for the UPP, for an attack of duration T, with C seizures, is given by:

$$UPP = \frac{P_u \cdot C}{T/\lambda} \tag{10}$$

3.3 Effort of the Attack

The *effort* of the attack, E, corresponds to the number of packets sent to the server by the intruder during the attack. Two sources of attack packets exists: the activity period, *ontime*, during which packets are generated at a rate $1/\Delta$; and the packet sent as a response to the reception of an output by the intruder.

For the calculation of the *effort* an assumption will be made: the intruder will receive the answers from the server after sending all the packets corresponding to the *ontime* interval. This is similar to suppose that the attack period is not going to be restarted during *ontime*, being the number of packets generated equal to $ceil(t_{ontime}/\Delta)$. On the other hand, not all the outputs are received at the intruder side. Therefore, no attack packet is sent in this case. The percentage of attack periods at which an output is not received from the server is given by UPP.

All these considerations lead to:

$$E = ceil\left(\frac{t_{ontime}}{\Delta}\right) + (1 - UPP) \tag{11}$$

4 MODEL DISCUSSION

In summary, the described model allows to obtain the expected value of the *mean idle time* for any specific setting of the attack and server/network characteristics. Both aspects are considered in the model as:

- Server characteristics: they are considered in the $f(t)$ term, which is univocally defined by the mean value and the variance of the inter-output time observed by the intruder.
- Network characteristics: as discussed above, the main network factor that affects the attack is the round trip time. As it can be seen from the final expressions of the model, it is clear the affection of this parameter. Both the mean value ($\overline{RTT}$) and its variance, $var[RTT]$, through the distribution $f(t)$, affects the value obtained for $\overline{T}_{idle}$.
- Setting of the attack: the configuration of the attack is reflected on the calculation points of the expression. In effect, their position depends on the parameters of the attack, that is, $t_{offtime}$, t_{ontime}, and the considered value for the *interval* Δ.

Some conclusions concerning the behaviour of the attack can be deducted from this model:

- As the value of $\overline{RTT}$ increases, a lower efficiency is obtained.
- Lower values of T_i are obtained when the output is generated during the reception of the *ontime* interval, if the design of the attack complies the condition $\Delta \leq \overline{RTT}$. Moreover, in this case, as the value of interval is lower, a lower value for $\overline{T}_{idle}$ is also expected. Outputs generated out of the *ontime* interval imply a higher value for the *mean idle time*.
- Variations in the mean service time does not affect the behavior of the attack period. This implies that, for an attack period, the total time during which a free position is available will be the same. As the *mean idle time* is defined as a percentage, lower values are expected when $\bar{t}_s$ increases.

On the other hand, the model also provides the expected values for the *user perceived performance* and the *effort*, which determine the performance of the attack.

For the UPP, two conclusions can be drawn from Eqs. (10) and (8). First, the tendencies of the values for UPP and $\overline{T}_{idle}$ are similar, that is, an increase in the value of UPP is expectable when the *mean idle time* becomes higher. Second, there is no linear dependency between the traffic rate of the legitimate users and the value obtained for UPP, as it can be seen from expressions (4), and (8).

Finally, it is clear from Eq. (11), that the effort is mainly affected by the duration of *ontime* and the setting of the *interval* value.

5 EXPERIMENTAL VALIDATION OF THE MODEL

The main purpose of this paper is to validate the theoretical framework presented in the above Sections. This objective has been accomplished through experimental results obtained from simulations made within Network Simulator 2 (NS2) (Fall, 2006).

In a first step, the results from the simulations are going to be compared with those from the proposed mathematical models to check its validity. Thereafter, the conclusions derived from the model concerning the expected behavior when some attack parameters are changed are tested.

5.1 Attack Performance Indicators

The first step in validating the models is, as previously stated, to check whether the expected values for the performance indicators are in accordance with those obtained by simulation. For this purpose, we have evaluated the behavior in a set of scenarios with different configurations for both the attack and server parameters. The results from these experiments have been compared to the values derived from the mathematical model, obtaining a very good approximation between them.

First, we examine the *mean idle time*, as the other measurements are deducted from it. Fig. 3.a) shows a comparison between expected and simulated values for this parameter in 13 different scenarios from the test set. The maximum variation given by the model is 3.77%, with a mean value of 1.71%, what means a very good approximation.

Next, the remaining parameters are tested. Fig. 3.b) shows a comparison for 13 different scenarios, where the *user perceived performance* is evaluated. The results are shown both in absolute and relative values. As can be seen, the obtained values from the model approximate well to the simulated ones, with a mean variation of 0.4% and a maximum of 1.46%.

Finally, Fig. 3.c) depicts the comparison for the *effort* between both the simulation and the model values, for 13 different scenarios. It can be observed that the model approximates well the simulated values, with a mean variation of 1.42% and a maximum of 4.02%.

As a conclusion, the approximations made in the mathematical model can be considered accurate enough, providing validity to the model as a tool to evaluate the potential effect of an attack starting from the knowledge of its design parameters.

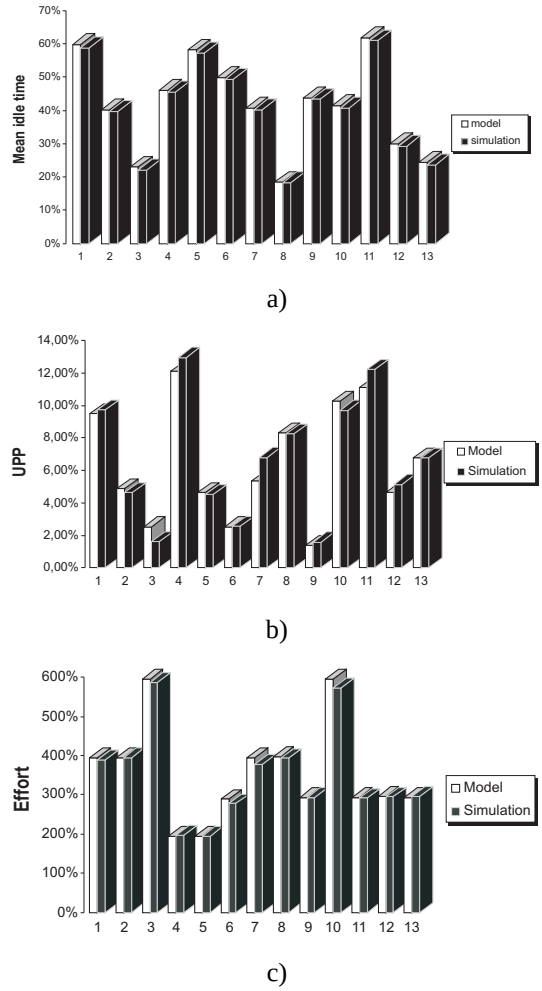

Figure 3: Comparison between the values obtained from simulation and from the mathematical model for 13 different scenarios: a) *mean idle time*; b) *user perceived performance*; and c) *effort*.

5.2 Attack Dynamics

To check the different conclusions extracted from the mathematical models about the behavior of the attack (attack dynamics), some simulations have been made. The purpose is to study the variations of the proposed indicators that measure the efficiency and the rate of the attack, as the parameters of the attack are adjusted. In a first approach, these parameters are supposed to be independent on each other. Therefore, only one of them is modified on each set of experiments.

The results obtained are shown in Fig. 4. It shows the effectiveness of the attack, in terms of UPP and T_{idle}, and also the effort for those attacks.

Some conclusions can be extracted from these experiments:

- The tendencies in the behavior of UPP and $\overline{T}_{idle}$ are similar. That is, an increase in the value of UPP also involves a higher value for $\overline{T}_{idle}$ -see Fig. 4-.
- As the *ontime* period becomes longer or the $interval$ takes a shorter value, both UPP and $\overline{T}_{idle}$ are reduced, and a higher $effort$ is obtained – see Figs. 4.a), b), c) and d)–.
- An increase in the value of $\overline{RTT}$ affects only to the efficiency and not to the rate –Figs. 4.e) and f)–. This implies that a higher value of UPP and $\overline{T}_{idle}$ is obtained when the $\overline{RTT}$ increases.
- The increase of the $var[\tau_{int}]$ implies a lower efficiency –Figs. 4.g) and h)–. This is because more outputs are generated out of the boundaries of the *ontime* interval. Moreover, when an output arrives to the intruder before the *ontime* interval is sent, the attack period is restarted. Due to this fact, the *effort* decreases when the variance is higher.
- The value of the efficiency decreases as soon as the mean value for the service time ($\overline{t_s}$) is higher –Figs. 4.i) and j)–. The reason is that the *mean idle time* represents a percentage of the time during which the service queue has a free position and this time is equal independently of the mean service time. Moreover, the *effort* is not affected by this kind of variations in the setting of the attack.
- There is no linear dependency between variations in the rate of the legitimate users traffic and the efficiency –see Figs. 4.k) and l)–, as expected from expression (10).

All these conclusions are coherent with those obtained from the mathematical models in Section 4, which allow us to conclude that these models are valid for an approximated analytical study of the behavior of the attack.

6 CONCLUSIONS

Low-rate DoS attack against iterative server (Maciá-Fernández et al., 2006b) has appeared as a threatening way of attacking simple applications. In this paper, the fundamentals of the design for these kind of attacks are presented.

A framework for evaluating the behaviour of the attack is contributed. Leaning on it, a mathematical model that let us to adjust the design parameters to get an specific efficiency and rate is proposed. Finally, by means of simulation, all the design rules proposed are contrasted.

As a future work, the comprehensive understanding of the behaviour of this kind of attacks will lead to the development of response and detection mechanisms. We are now working in this field, obtaining promising results.

ACKNOWLEDGEMENTS

This work has been partially supported by the Spanish Government through MYCT (Project TSI2005-08145-C02-02, FEDER funds 70%)

REFERENCES

Axelsson, S. (2000). Intrusion detection systems: A survey and taxonomy. Technical Report 99-15, Department of Computer Engineering, Chalmers Univ., Goteborg.

Douligeris, Christos; Mitrokotsa, A. (2004). DDoS attacks and defense mechanisms: classification and state-of-the-art. *Computer Networks*, 44(5):643–666.

Fall, Kevin; Varadhan, K. (2006). *The ns manual*. Retrieved from http://www.isi.edu/nsnam/ns/.

Ferguson, P.; Senie, D. (2001). Network ingress filtering: defeating denial of service attacks which employ ip source address spoofing. RFC 2827.

Geng, X.; Whinston, A. (2000). Defeating distributed denial of service attacks. *IEEE IT Professional*, 2(4):36–42.

Kuzmanovic, A.; Knightly, E. (2003). Low rate TCP-targeted denial of service attacks (The shrew vs. the mice and elephants). In *Proc. ACM SIGCOMM'03*, pages 75–86.

Maciá-Fernández, G., Díaz-Verdejo, J., and García-Teodoro, P. (2006a). Evaluation of a low-rate dos attack against iterative servers. *Submitted to Computer Networks*.

Maciá-Fernández, G., Díaz-Verdejo, J., and García-Teodoro, P. (2006b). Low rate dos attack to mono-process servers. *Lecture Notes in Computer Science*, 3934:43–47.

CERT Coordination Center (2003). *Denial of Service attacks*. Retrieved from http://www.cert.org/tech_tips/denial_of_service.

SANS Institute (2000). Special notice - egress filtering. global incident analysis center.

Shevtekar, A., Anantharam, K., and Ansari, N. (2005). Low rate tcp denial-of-service attack detection at edge routers. *IEEE Communications Letters*, 9(4):363–365.

Sun, H., Lui, J., and Yau, D. (2004). Defending against low-rate tcp attacks: Dynamic detection and protection. In *Proc. of the IEEE Conference on Network Protocols (ICNP2004)*, pages 196–205.

Weiler, N. (2002). Honeypots for distributed denial of service. In *Proc. of the Eleventh IEEE International Workshops Enabling Technologies: Infrastructure for Collaborative Enterprises 2002*, pages 109–114.

Williams, M. (2000). *Ebay, Amazon, Buy.com hit by attacks, 02/09/00*. Retrieved from http://www.nwfusion.com/news/2000/0209attack.html.

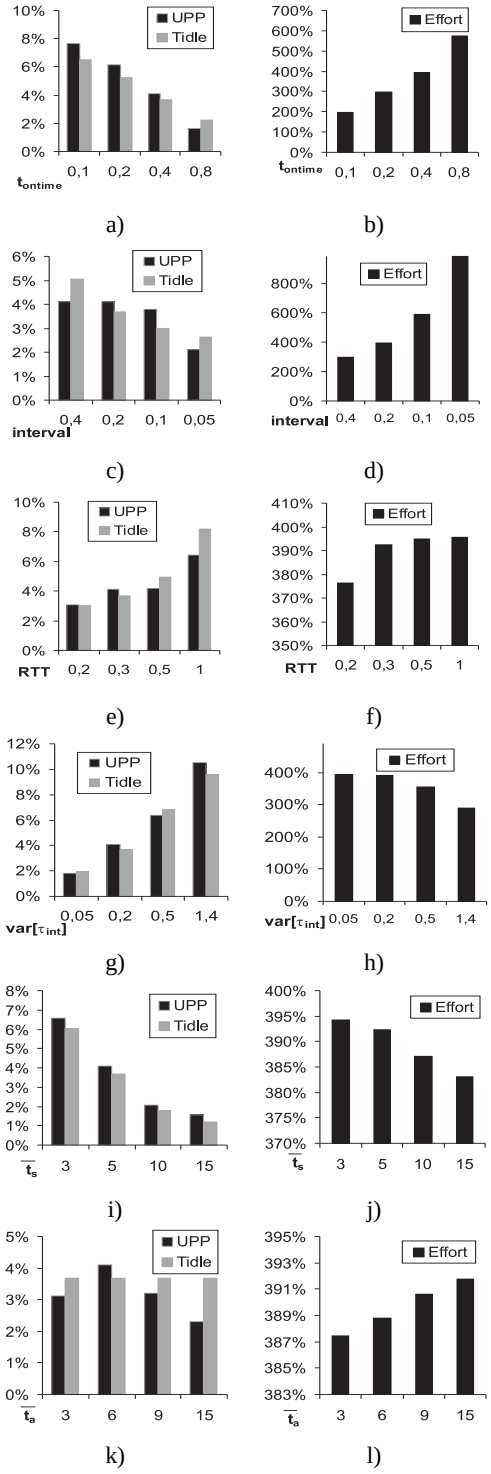

Figure 4: Attack dynamics. Variations on the indicators of the attack for different settings of the parameters: a) & b) *ontime interval*, c) & d) *interval* Δ, e) & f) mean round trip time, g) & h) variance of the inter-output time perceived by the intruder, i) & j) mean service time, and k) & l) legitimate users traffic rate.