MODELLING OF CENTRAL PROCESSING UNIT WORK
DENIAL OF SERVICE ATTACKS

Simona Ramanauskaite1, Antanas Cenys2
1Siauliai University, Department of Information Technology,
Vilniaus st. 141, Siauliai, Lithuania, simram@it.su.lt
2Vilnius Gediminas Technical University, Department of Information Systems,
Sauletekio st. 11, Vilnius, Lithuania, ac@fm.vgtu.lt

Abstract. Denial of service attacks is one type of cyber attacks, when attacker tries to make some kind of service not available to its legitimate users. It becomes more relevant for service providers, however, for estimating of the potential threat it is simpler to use mathematical/programmable models instead of the real experiments. DoS attacks can vary in type, as well as its models. In this paper, we propose central processing unit work depletion DoS attack model and its characteristics. Also we use this model to examine attack success probability dependencies on the used queries arrival rate function as well as the optimal queries arrival rate function for the overall attack time period. The proposed model should enrich the other type of DoS models in the way of incorporating them. Meanwhile, the implemented modelling experiments can be used as guidelines for central processing unit work depletion DoS attack efficiency optimization.

Keywords: DoS, Denial of Service, modelling, CPU usage.

1 Introduction

The number of different services in the Internet is increasing every day. Some of these services become more and more popular, sometimes even necessary for some users, functionality of the other systems. Therefore the quality of service becomes an important factor for its proper functioning, not mentioning functioning itself.

Attacks which try to disturb some kind of internet services by denying this service to its legitimate users are called Denial of Service (DoS) attacks. According to CERT [1], in 2010 Denial of Service attacks were in the 6-th place considering how often it occurs in the word. Office of the State Chief Information Security Officer in United States of America ((State of Texas)) stated that DDoS attacks as the way to make ransom attacks will also be one of the most popular attacks in future [5]. Meanwhile, looking into Lithuania in the 3-rd quarter of 2010, this type of attacks was in 4-th place considering occurrence [2].

Usually good preparation can help to avoid DoS attack effect on the certain system and the service it is supplying, however, to estimate the effectiveness of some kind of prevention properties can be quite complex. The aim of this work is to suggest a mathematical model for modelling of central processing unit (CPU) work Denial of Services, and using it, to examine the possible attack success changes when different strategies of the incoming attack flow are used.

2 Denial of Service attacks

Denial of Service attack is a type of cyber attacks. In a denial-of-service (DoS) attack, an attacker attempts to prevent the legitimate users from accessing information or services. By targeting your computer and its network connection or the computers and network of the sites you are trying to use, the attacker may be able to prevent you from accessing email, websites, online accounts (banking, etc.) or other services that rely on the affected computer [11]. In order to increase the attack power, many controlled computers can be used. This kind of attack is also known as Distributed Denial of Service (DDoS) attack.

Practical experiments with DoS and DDoS attacks are difficult because of these reasons:
- The attack sources area spreads in the wide geographical area, and experiments in the local network can be insufficient to illustrate the real situation;
- DDoS attacks require a lot of controlled computers and can cause difficulties getting the sufficient amount of infected and ready to attack computers;
- Real experiments in the Internet can cause problems for third parties, disturb the work of innocent Internet users or even services;
- Execution on DoS attack in the Internet can be illegal.

Examination of the attack properties without the real execution of DoS attacks can be done using different modelling methods and tools. However talking about modelling of DoS attacks, there can be different types of DoS attacks, so different models must be used to model the desired attack situation.
There are three main categories in DoS attack classification concerning resource, which can be exhausted to cause the DoS effect:

- Bandwidth exhaustion DoS attacks;
- Memory depletion DoS attacks;
- Central processing unit work depletion DoS attacks.

2.1 Bandwidth exhaustion DoS attack

Bandwidth exhaustion DoS attack happens when an intruder consumes all the available bandwidth on a certain network by generating a large number of packets directed to your network. Typically, these packets are ICMP ECHO packets but in principle they may be anything [8].

Qiang Huang, Hisashi Kobayashi and Bede Liu suggested two models for modelling bandwidth exhaustion DoS attacks. One of them is for DoS attacks in the global network [9], and the other one is for the wireless networks [10]. These two models offer the methods allowing finding the minimal number of agents necessary to execute the successful DDoS attacks, however, the models do not pay enough attention to the properties of all attacks. More attack properties are taken into account in the paper focusing on the modelling of DoS attacks using stochastic methods [14]. Also this paper represents mathematical expression for quantitative DoS attack success calculation using the known data on the attacks, normal flow and other victim’s properties.

2.2 Memory depletion DoS attack

DoS resource depletion attacks involve the attacker sending packets that misuse network protocol communications or sending malformed packets that tie up network resources so that none are left for legitimate users [15].

Memory depletion DoS attacks are the most common because of the noticeable effect and quite low attack expenses. This is why there is a quite big range of proposed memory depletion DoS attacks models:

- Q. Huang and other authors of the paper “Analysis of a New Form of Distributed Denial of Service Attack” [9] apply the simplified Engest loss model $G(N)/G/m(0)$. This model enables to estimate the success of SYN flooding attack but there are no characteristics of the legitimate users in this model, only the attack itself is characterised;
- The author of “Defending against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial” [3], uses $G/D/\infty/N$ model to calculate the minimal attach flow, which is necessary to make a successful TCP SYN attack. However, in this work the model is not described in detail, and only the results of the experiment are given.
- Y. Wang and the other authors use the two-dimensional embedded Markov chain model in the paper “A queuing analysis for the denial of service (DoS) attacks in computer networks” [17]. This model takes into account legitimate and attack flow characteristics as well as the buffer size. But this model is difficult to use because of the complex calculations.
- Authors of the paper „Modelling of SYN Flooding Attacks“ [13] suggest the SYN flooding attack model which can be used for any kind of the memory depletion DoS attacks and allows to estimate the success probability of the attack, taking into account both victim and attacker properties.

2.3 CPU work depletion DoS attacks

CPU work depletion DoS attacks differ from memory depletion attacks considering the exhausted recourse type. This type of attacks attempts to make victim to do some kind of job which pretends to seem more important than the legitimate user requests or requires more time to finish it. For these reasons, the legitimate user requests are served slower and even the impression of not execution at all may appear as the attack requests use up CPU processing cycles to prevent the legitimate processes from being processed.

This is one of the oldest known forms of denial of service and mature operating systems most often have the defence, nevertheless, some of them are still vulnerable to this type of DoS attacks [6].

Maybe because of variety of different systems and varying request service strategies in them, CPU work depletion DoS attacks do not have a huge effect on it. It is the same as with the models of this type of DoS attacks. There exist different models for CPU performance and scheduling algorithm analysis [4, 7, 12], however the impact to CPU work during DoS are not analyzed widely. This is why we suggest a CPU work depletion DoS attack mathematical model, which would let judge the success of this type of DoS attacks with certain attacker and victim properties configurations.
2.3.1 Conceptual model of CPU work depletion DoS attack

The basic idea of CPU work depletion DoS attack model should be that there exist the criteria allowing judging what might be the success of this type of attack. It would be wrong to judge the success of attack considering the processor utilization percentage because the legitimate user does not care about the system itself but just about the service he gets. This means the success of attack should be judged considering the time necessary to process the legitimate user query.

Thinking about CPU work, it is also important to remember the fact that one CPU can serve just one process at one time moment and the illusion of multiprocessing is achieved by scheduling small parts of certain tasks. In this work, only those situations are analyzed where the non-priority scheduling algorithms are used and where the legitimate and attack requests are not distinguished. Also we assume that CPU work depletion DoS attacks do not block incoming requests, and the data buffer to store requests is infinite. This is because the control of incoming requests blocking is done by memory – not CPU work – management systems.

To fully define the CPU work model we use these properties:

- CPU takes one query and process it for a $k$ cycles within one take;
- Legitimate queries require approximately $ct$ takes to CPU;
- Attacks queries require approximately $ca$ takes to CPU;
- At one time moment there are $M$ queries in the buffer;
- There are $p_l$ percent of legitimate and $1-p_l$ percent of attack queries in the system;
- The processor frequency is $f$, which means how many cycles can be processed by the processor in one time moment (second).

We assume that legitimate and attack flows are distributed evenly. This allows estimation of the average number of takes into CPU for one query (how many times query should be processed by CPU to finish it):

$$c = c_l \cdot p_l + c_a \cdot p_a$$  \hspace{1cm} (1)

For deeper analysis of query processing in CPU, we should know what scheduling algorithm is used. One of the most populars is FIFO (First In – First Out). Using this scheduling algorithm queries are not divided into small blocks and they all are processed instantly. Immediately after the processing of one query, the other is processed – then CPU finishes processing of the query, and the query is removed from the buffer.

![Figure 1. Conceptual model for serving of FIFO query](image1)

Another popular non-priority scheduling algorithm is “Round-robin”. Using this scheduling algorithm, all the queries are processed in small parts. When one small part of query is executed – and if it was not the last part of query – this query “goes” back to the end of the queue (buffer). And just after executing the last part of it, the query is removed from the queue (buffer).

![Figure 2. Conceptual model for serving of Round-robin query](image2)
To estimate the average time of certain query, we should count how many queries are in front of the watched one every time the query goes back to the end of queue (buffer). So we have to express the size of buffer as a time function $M(i)=M(i-1)-M_{done}(i-1)+M_{new}(i-1)$, which would be estimated by queue size one moment ago, adding newly coming queries and subtracting number of queries finished in the previous time moment. So the average service time could be expressed as following:

$$T = \sum_{i=1}^{\infty} \frac{c \cdot M(i)}{f}$$

(3)

Using this formula the average query service time can be estimated quite precisely even when size of queue changes dynamically. But usually the buffer size is quite constant, for example, in those situations, when attacker sends so many queries that the buffer is fully filled and even some of them are blocked (especially in DDoS attacks). In such situations the average service time formula can be simplified and would be the same as in FIFO scheduling algorithm.

2.3.2 Estimation of attack success probability in CPU work depletion DoS attack model

No matter FIFO or “Round-robin” scheduling algorithm is used, to estimate the attack probability there must be known the critical request service time limit $T_{max}$. It shows how long the legitimate user is really to wait until it is assumed that the request did not reach the service or was rejected. So the attack success probability can be expressed as ratio of average and critical service times. And in these situations when the average service time is greater than the critical service time, the attack success probability is equal to 100%:

$$P = \begin{cases} \frac{T}{T_{max}} , & T \leq T_{max} \\ 1 , & T > T_{max} \end{cases}$$

(4)

2.3.3 Modelling experiments

The proposed CPU work depletion DoS attacks model formulas and its modelling result show the dependencies between the attack and victim properties. The average attacks success probability linearly depends on the average query service time (number of cycles necessary to finish legitimate and attack query and its quantity distribution in the queue) and queue size in the buffer and is inverse to CPU frequency and critical service time. Similar dependencies can be noticed in experiments of the other authors [16, figure 11].
Using these query arrival functions with the proposed CPU work depletion DoS model, we noticed such tendencies:

- When arrival rate is the same as service, attack success probability stays the same all the time;
- When arrival rate is less than service time, the success probability decreases (but slower), and then it increases with higher arrival rate than the service rate;
- When arrival rate is constant, the attack success probability changes linearly;
- When arrival rate changes in time linearly, the success probability increases exponentially;
- When attacker starts with high arrival rate and decreases it linearly in time, the success probability is increasing until the rate reaches 0 value and then start to decrease.

![Figure 5. Attack success probability changes in time, depending on queries arrival function](image)

In addition to the examination of the success probability curve changes in time depending to queries arrival rate, the analysis of the arrival rate efficiency considering attack success was done as well. These tests showed that the most efficient are the arrival rate functions when the number of queries in the buffer is big at the start point even if latter it can decrease. Meanwhile, if attack would use a linearly increasing arrival rate, the attack success would change more rapidly, however, it would be more sufficient only at the end of attack, and not during the overall attack time.

3 Conclusions

1. There are many different models for memory depletion and few for bandwidth exhaustion DoS attacks, however, practically none for CPU work depletion DoS attacks. This may be because of not popularity of this type of DoS attacks and quite little success probability; nevertheless, it is necessary to make deeper analysis of the different type of DoS attacks and more precise value estimation of the other types of DoS attack properties.
2. CPU work depletion DoS attack model is assumed considering that the buffer size is infinitive. This means this model should not be used separately for real situation modelling, and it should be combined with memory depletion DoS attack model. Such combination would work more realistically for both – CPU work and memory depletion DoS attack – models.
3. The average execution period for user queries in CPU work depletion DoS attacks depends not only on time necessary to execute the query but on the queue size as well. So if the incoming traffic increased adequately considering both parameters – legitimate and attack – the success probability would increase linearly as well because of the increasing queue size. But if attack traffic increased and the legitimate traffic remained constant, the success probability should increase not linearly, but exponentially.
4. Attack success probability in time changes accordingly to the arriving query rate function. To get the most effect considering the overall attack time, the instant big query arriving rate should be used, and what can be achieved choosing a certain query arriving rate or combining some of them in time is the desired attack effectiveness.
References


