# СИНТЕЗ И АНАЛИЗ СИММЕТРИЧНЫХ КРИПТОПРИМИТИВОВ

**UDC 681.3.06**

*R.V. OLIYNYKOV, PhD, D.V. MURIN, S.V. KYLYPKO*

## MODELLING OF BOTNET BASED DDOS ATTACKS TARGETED TO WEB APPLICATIONS

### Introduction

On present level of technology development dependency between efficiency and protection of communication information systems becomes more and more obvious. Availability (opportunity to have immediate access to resources, if you have appropriate access rights) is one of the basic requirements to such systems. Availability violation is possible either as cause of hard\soft-ware errors or as a consequence of attackers activity. Also, malicious software is a big problem. It is very expensive to get rid of all malicious software in the network. So, it is question of current interest to find and use methods which allow to make balanced decisions for network protection(cost-benefit balance).

### Analysis and characterization of load

Distributed system performance depends a lot on workload. So, as the first step to measure performance, we have to analyze and characterize the load As a system workload we should mean a set of all stimuluses received from external sources for fixed period of time. For example for database server, workload is a set of transactions ( update-search requests).

Actual load contains bigger amount of components. To measure system performance we need to analyze load information and choose options which describe it in the best way. In other words we need to build a model of the load, which reflects the behavior of actual load. Load models have some advantages compared with actual loads, or tracers (where options are being fixed for monitored period of time) First of all, it is enough to change model options to see changes of the system or load. For example: to increase the model of load we need to change only one option: either time between requests , or “user delay” .In order to improve reflection of behavior of the system we can introduce correlated options into the model. Using models of the load allows to get a great improvement of developing system.

Characteristics and options for describing of the load are being chosen depending on the type of research. For example in cost-benefit analysis for installation of cache-server for web-site we should choose following options: document request frequency, request distribution, size of requested documents, time between requests. So we can determine the profit from installing cache-server analyzing percentage of documents which are demanded by the most of requests to the site.

For measuring influence of faster processor on web-server reply we need a different set of load characteristics: average time for service of one request , average number of i\o operations for one request, average response time for one request.

So for every system we need different set of options to build a load model. Despite that there are common principles for any system. Load characterizing requires nest steps:

1. Specify a point of view from which the system will be analyzed
2. Choose a set of options which will be the most significant for research targets.
3. Explore the system to get elementary data about its performance
4. Primary statistics analysis and data unification.
5. Build the model of the load
6. Evaluate actuality and reliability of a model.

### Research of impact on the load by attacker

The weak spot of any web application is the link which provides information and computation power to the end-user. Attacks which implement this vulnerability is called DoS/DDoS (Distributed Denial of Service). Nowadays one of the most dangerous source of this attack is botnet. A botnet is a collection of Internet-connected programs communicating with other similar programs in order to

perform tasks. Most of all , bot in botnet is a program that is being installed secretly on victims computer and attacker is able to use victims computer for his own purpose. So, attacker may create traffic in the network using all the elements of botnet. This traffic may prevent normal operation of application. Crucial botnet parameter is a number of its elements (bots). Measurement and prediction of bot amount increase is a question of current interest.To analyze increase in the number of botnet elements we can use method described in previous chapter.

Lets define three classes of elements in system which is being attacked: A – clean elements without malicious software; B – elements which are being attacked; C – infected elements. Malicious software makes the element to send illegitimate requests to application.

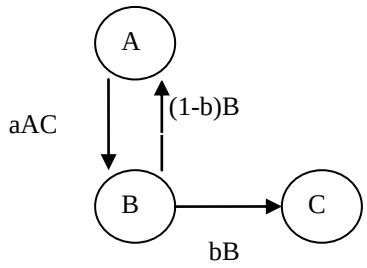


Figure 1. Model of propagation medium of malicious software

Probabilities: a - probability of successful implementation of vulnerability; b – probability of successful infection after implementation of vulnerability.

Basic model might be introduced in this way:

$$A` = -aAC + (1-b)B$$

$$B` = aAC - B$$

$$C` = bB$$

Current system state can be introduced this way: $S_0 = \begin{bmatrix} A \\ B \\ C \end{bmatrix}$

Basic interaction model can be introduced by Jacobian: $J = \begin{bmatrix} -aC & 1-b & -aA \\ aC & -1 & aA \\ 0 & b & 0 \end{bmatrix}$

For example lets build model with following parameters: $S_0 = \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} 999 \\ 0 \\ 1 \end{bmatrix}$

$a = 0.0001; b = 0.7$

So we get following diagram of changing of the quantitative indicators for each class in system(figure 2):

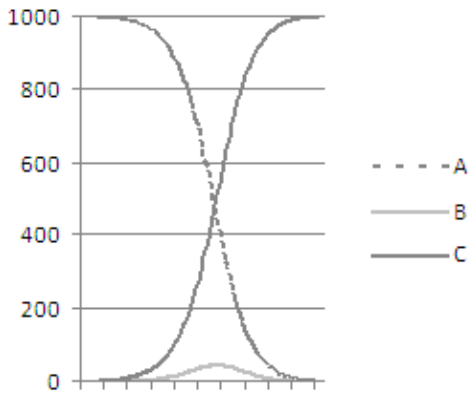


Figure 2. Changing of the quantitative indicators in system

Given the fact that illegitimate request are sent by infected computers only and knowing requests per second, attacker will create a load on the system shown on figure 3: $W = \begin{bmatrix} 0 \\ 0 \\ 5000 \end{bmatrix}$

In unprotected network malicious software propagate quite fast. Without any protective software attacker gains control of elements fast and greatly increases load on network (figure 3).

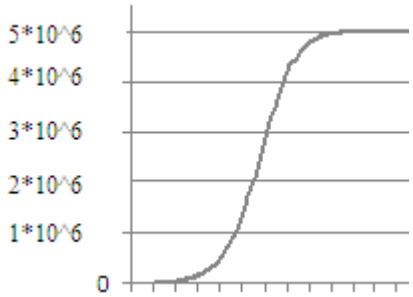


Figure 3. Increase of the load on application during DDoS-attack by botnet

Comparing loads created by botnet and by users, we can see that amount of illegitimate requests increases noticeably faster. That leads to impossibility to provide application availability(figure 4)

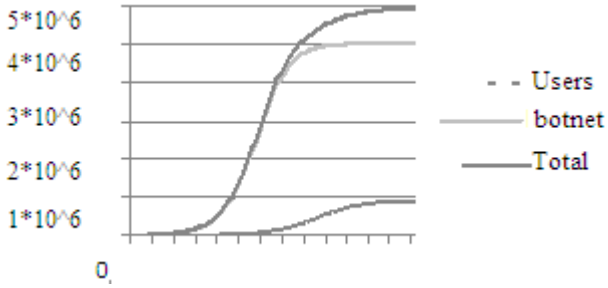


Figure 4. Load on application during DDoS attack

To provide protection from botnet it is necessary to use different means of protection. Choosing the most effective means would allow to stop propagation of malicious software in network timely and noticeably decrease load on the application.

Consider the impact on propagation of malicious software if we use protection software with cure $q = 0.05$ (Figure 5)

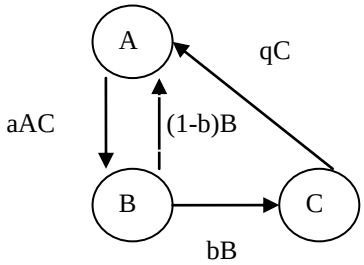


Figure 5. Model of propagation medium of malicious software in system with antivirus

$$A` = -aAC + (1-b)B + qC$$

Basic model might be introduced in this way: $B` = aAC - B$

$$C` = bB - qC$$

Basic interaction model can be introduced by Jacobian: $J = \begin{bmatrix} -aC & 1-b & -aA=qC \\ aC & -1 & aA \\ 0 & b & -qC \end{bmatrix}$

So we get following diagram of changing of the quantitative indicators for each class in system(figure 6). There are still some infected elements in system , but their number is less, illegitimate load on application is also less and it increases slower (figure 7).

Usage of newer and more effective means of protection would allow to increase probability of curing the element and decrease amount of infected elements even more.

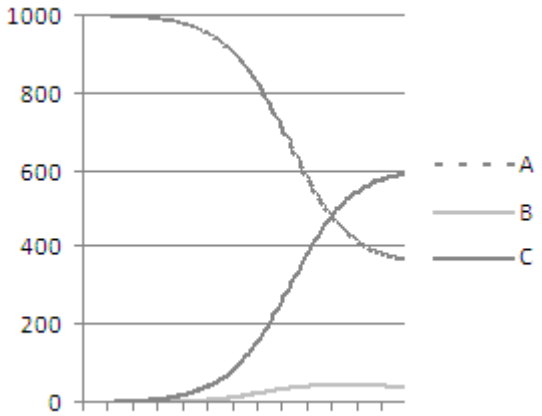


Figure 6. Increase of the load on application during DDoS-attack by botnet, system with antivirus

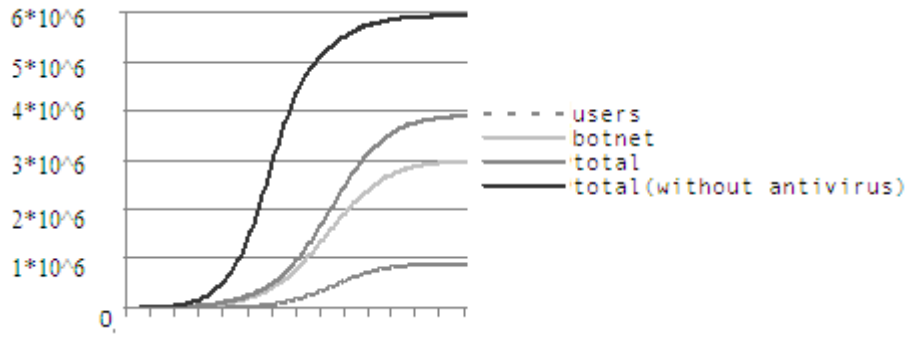


Figure :7. Load on the application during DDoS attack (antivirus software is used)

**Measurement of application availability**

If we consider web application as many-linked queuing system theory with limited queue and limited waiting in the queue, we can take a probability of service for system request (Q)as an availability of application. Calculations are made on known formulas from queuing system theory. Lets take application from part 2 of this article with following parameters to illustrate availability $n = 100; m = 150; \overline{T}_{service} = 0.8\,\text{sec}; \overline{T}_{waiting}$

As we can see on figure 8, application availability is on very high level (99.9%), but after a certain moment availability decreases very fast. Consequently , it is possible to get the limit of requests which application can service with given availability. When amount of requests gets to critical limit , system resources should be scaled in order to provide sufficient availability.

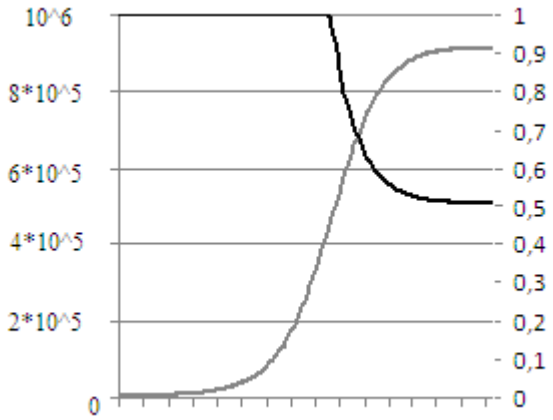


Figure 8. Application availability with network effect

Consider imply of different scaling methods on application availability. Supposably , given system is a physical server which can service $n = 100$ threads simultaneously.

If we use horizontal scaling method (increase amount of servers) , we get multiply threads. Load limits for one and two servers with 99.9% availability: $R_1 \approx 420300$ ; $R_2 \approx 900000$

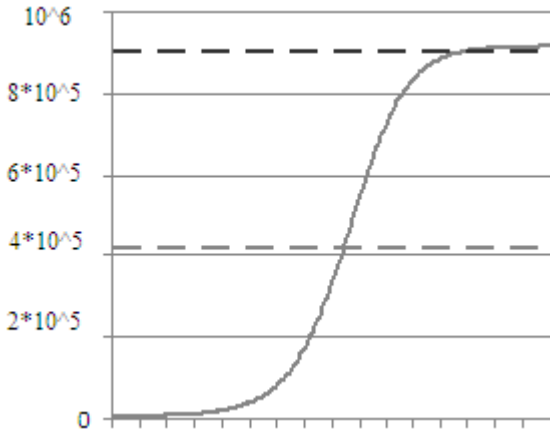


Figure 9. Horizontal scaling load limits

As we can see, two servers increase availability, but it is not enough. At least three servers are needed to provide sufficient availability.

If we use vertical scaling method and double system productivity, we get 30% speed increase. So time for service of one request is : $\overline{T}_{service} = 0.8\,\text{sec}$

We get following load limits: R≈620000

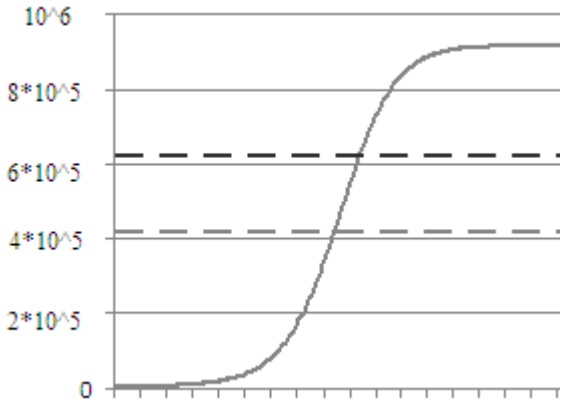


Figure 10. Vertical scaling load limits

As we can see on figure 10 , this method gives relatively small increase to load limits.

Thus, usage of horizontal scaling is more effective in given system. But, if you analyze any system, it is necessary to model both methods .

Usage of proposed method allows to predict load increase and calculate system efficiency in order to scale system timely.

**Conclusion**

Usage of proposed a method allows to predict load increase and calculate system efficiency in order to scale system timely. It is possible to use more complicated mathematical models in order to get more accurate data. We need to describe nonlinear properties of the network effect using this models and model work of a system using queuing system theory.

Proposed method allows making prediction of web application load increase during given period of time with condition of DDoS attack possibility for effective implementation of protection means. Defense of such type of attacks is quite important for modern information society for effective implementation of the most valuable information services like e-government, production control, logistics, etc.

**References :** 1. *J. M. Almeida, J. Krueger, D. Eager, and M. K. Vernon*, "Analysis of Educational Media Server Workloads," Proc. Wth Int'l. Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV 2001), Port Je.erson, New York, June 25-26, 2001.2. *M. Arlitt and C. Williamson*, "Web Server Workload Characterization: Тдо Search for Invariants," Proc. ACM 1996 SIGMETRICS Conf. Measurement Comput. Syst., Philadelphia, Pennsylvania, May 1996, pp. 126-137. 3. *G. Banga and P. Druschel*, "Measuring the Capacity of a Web Server," Proc. USENIX Symp. Internet Technol. Syst., Monterey, California, Dec. 1997. 4. *M. Chesire, A. Wolman, G. Voelker, and A. Levy*, "Measurement and Analysis of a Streaming-Media Workload," Proc. 3rd USENIX Symposium on Internet Technologies and Systems, San Francisco, California, March 2001..

*Харьковский национальный*
*университет радиоэлектроники*

*Поступила в редколлегию 10.12.2013*