Scenario-Based Markovian Modeling of Web-System Availability Considering Attacks on Vulnerabilities

Vyacheslav Kharchenko[1], Yurij Ponochovny[2], Artem Boyarchuk[1] and Anatoliy Gorbenko[1]

[1] National Aerospace University KhAI, Kharkiv, Ukraine
V.Kharchenko@khai.edu
[2] Poltava National Technical University named after Yurij Kondratyuk, Poltava, Ukraine
pnch1@rambler.ru

Abstract. In the paper we simulate web-system availability taking into account security aspects and different maintenance scenarios. As a case study we have developed two Markov's models. These models simulate availability of a multi-tier web-system considering attacks on DNS vulnerabilities in additional to system failures due to hardware/software (HW/SW) faults. Proposed Markov's model use attacks rate and criticality as initial simulation parameters. In the paper we demonstrate how to estimate these parameters using open vulnerability databases (e.g. National Vulnerability Database). We also define different vulnerability elimination (VE) scenarios and examine how they affect system availability.

Keywords: web-system availability, security, vulnerability, Markov's models, scenario of vulnerability elimination

Key terms. MathematicalModeling, MathematicalModel, SoftwareSystems

1 Introduction

Efficient implementation and operation of multitier web-systems using COTS components depend on accuracy of security assessment and quality of attacks prevention and recovery activities. Security of web-system can be estimated by analyzing web-components vulnerabilities and predicting attacks affecting system availability and other security attributes. System availability and accessibility of the provided ser-vices depend on the used maintenance strategy. This strategy can implement various vulnerability prevention and elimination scenarios [1]. Thus, assessing web-systems availability taking into account both system failures due to HW/SW faults, and hacker attacks on components vulnerabilities is important.

To estimate system availability and security researchers develop various simulation models [1, 2]. Most of them are based on attack tree analysis [3,4], Markov's [5,6] and

semi-Markov's chains [7,8] or use of Petri nets [9,10] as a mathematical apparatus. However, known models do not explicitly consider attacks on system vulnerabilities causing inaccessibility of the provided services (accessibility vulnerabilities) and do not take into account different security policies and vulnerability elimination strategies.

In the paper we analyze web-system availability considering failures caused by HW/SW faults as well as attacks on system vulnerabilities. With this purpose we propose and examine a set of Markov's availability models implementing different scenarios of vulnerability elimination. This paper continues research described in [6] using scenario-based approach.

The rest of the paper is organized as follows. In the second section we suggest a set of scenarios to assess web-system availability taking into account different vulnerability elimination procedures. In the third section we discuss a technique of estimating input parameters of Markov's models by use of information about software component vulnerabilities from the open vulnerability databases. The forth section presents a case study and the set of Markov's models and also examines simulation results.

2 The Scenario-Based Approach to Web-System Availability Modeling with Regards to System Vulnerabilities and their Elimination

Attacks on vulnerabilities of web-systems can be simulated using Markov's models [5-7]. However, for that we should take into account that parameters of the vulnerabilities (numbers and types) are changed as a result of elimination and patching procedures.

In the Fig. 1 we propose a set of common state-transitional models capturing different attack and recovery scenarios. The scenarios are differed by a number of attacked vulnerabilities: one (a-f) or several (g); with (b-g) or without (a) vulnerability elimination; with vulnerability elimination after system been successfully attacked (b-d) or during (e,f) preventive maintenance actions.

We have marked model states as following: double circles correspond to up-states, single line marked circles correspond to maintenance states, thick line marked circles correspond to down-states after attacks.

The simplest scenario is shown in Fig. 1,a. After successful attack a web-system is recovered (e.g. rebooted) without vulnerability elimination. However not all attacks can be successful and lead to web system unavailability. This is why we consider two transitions from up-state S0: the first transition with the rate $\lambda attack*Da$ leads to down (unavailable)-state Sd; the second one with the rate $\lambda attack*(1-Da)$ returns back to up-state S0 (Da is a probability of attack to be successful).

The second scenario (Fig. 1,b) illustrates vulnerability elimination during system recovery after successful attack. We assume that during recovery action it is possible to eliminate from 0 to all (nv) vulnerabilities. Hence, web-system may return from the down-state Sd to the initial state S0 without vulnerability elimination with the rate $\mu'a*(1-Dp)$, where Dp is a probability of successful recovery and vulnerability elimination, or may transit to the next up-state Su with the rate $\mu'a*Dp$.

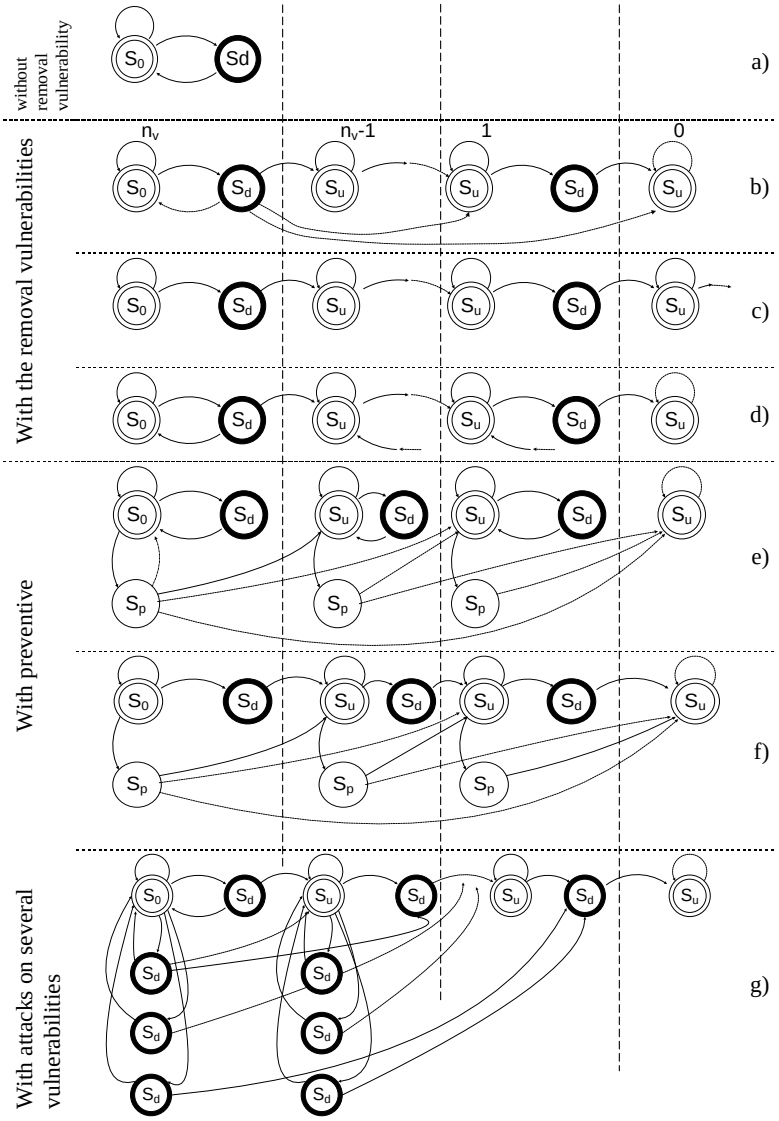

Fig. 1. Graph models of scenarios of web-system availability considering different options of vulnerability elimination

The third scenario (Fig. 1,c) describes graduate vulnerability elimination only after successful attacks on these vulnerabilities. In this scenario the total number of vulnerabilities in the system may be unlimited $nv \to \infty$.

The step by step vulnerability elimination is described by the next scenario (Fig. 1,d). In this case it is assumed that restart of web-system is possible without elimination of vulnerability which was attacked.

According with the fifth scenario (Fig. 1,e) vulnerabilities can be detected and eliminated from the system only during the periodic maintenance actions (i.e. security audits) only. After the successful attack a web-system is restarted or reboot without vulnerability elimination. Vulnerabilities can be detected and eliminated from the system only during periodic security audits. The probability of eliminating the i-th vulnerability is equal to αi, $\Sigma \alpha i = 1$.

The sixth scenario (Fig. 1,f) assumes that vulnerabilities can be detected and eliminated from the system both after successful attacks or during periodic security audits. The seventh scenario takes into account possibility of attacks on several vulnerabilities (Fig. 1,g). The scenario describes sequential chains of attacks on sever-al (four, in our example) services of a web-system. In this case an intruder continues to attack the next services. After successful attack a web-system can transit to a new up-state where vulnerabilities are eliminated from the system or can return back to the initial state by system restarting or rebooting.

Described set of scenarios is not complete. This set includes some basic scenarios. However, other scenarios can be developed considering different procedures of maintenance and vulnerability elimination or patching.

3 Estimation of Input Parameters for Markov's Web-System Availability Models

3.1 Vulnerabilities Sampling

In this section we discuss how parameters of Markov's models simulating web-system availability can be estimated using existing vulnerability databases like NVD.

The whole set of vulnerabilities stored in NVD can be downloaded as an XML file «NVD/CVE XML Feed with CVSS and CPE mappings (version 1.2)» [11,12]. Then we need to select those vulnerabilities of Web-system components (DNS-server, HTTP-server, application server, etc.) affecting system availability. It is can be done by analyzing vulnerabilities availability impact and vector of access using, for instance, common vulnerability scoring system (CVSS) [13] provided by NVD:

- Availability impact, A, which can be equal one of three fuzzy values "None" (N), "Partial" (P) and "Complete" (C);
- Vector of access, value "Network" (N).

For example, Table 1 presents a subset of vulnerabilities detected during 2013 and causing unavailability of DNS (CVSS_vector – contains – AV:N, A:C и A:P; ns1:descript – contains – DNS (an example for analysis attacks on DNS) including their publishing dates and score.

3.2 Estimation of Attack Rates

In order to parameterizes state-transition models we need to evaluate a rate of the attacks exploiting system vulnerabilities.

This rate obviously depends on different factors including number of system

vulnerabilities, their criticality, availability impact and vector of access. However, vulnerabilities define only the capability of a system to be attacked. On the other hand, unlike random system failures, vulnerabilities are exploited by various intended (hacker, computer criminals, industrial espionage, insiders, etc.) and unintended (viruses, worms, malware, etc.) threat agents.

Table 1. Subset of vulnerabilities causing DNS unavailability (01.2013 – 10.2013)

#	name	published	base_score	CVSS_vector
1	CVE-2013-0198	05.03.2013	5,0	(AV:N/AC:L/Au:N/C:N/I:N/A:P)
2	CVE-2013-2266	28.03.2013	7,8	(AV:N/AC:L/Au:N/C:N/I:N/A:C)
3	CVE-2013-2494	28.03.2013	4,9	(AV:N/AC:H/Au:S/C:N/I:N/A:C)
4	CVE-2013-1152	11.04.2013	7,8	(AV:N/AC:L/Au:N/C:N/I:N/A:C)
5	CVE-2013-2052	09.07.2013	5,1	(AV:N/AC:H/Au:N/C:P/I:P/A:P)
6	CVE-2013-2053	09.07.2013	6,8	(AV:N/AC:M/Au:N/C:P/I:P/A:P)
7	CVE-2013-2054	09.07.2013	5,1	(AV:N/AC:H/Au:N/C:P/I:P/A:P)
8	CVE-2013-4854	29.07.2013	7,8	(AV:N/AC:L/Au:N/C:N/I:N/A:C)
9	CVE-2013-4115	09.08.2013	7,8	(AV:N/AC:L/Au:N/C:N/I:N/A:C)
10	CVE-2013-5479	27.09.2013	7,8	(AV:N/AC:L/Au:N/C:N/I:N/A:C)
11	CVE-2013-5480	27.09.2013	7,8	(AV:N/AC:L/Au:N/C:N/I:N/A:C)

Motivation of intended threat agents is also depended on the system itself (its value and interest for the attacker). Last two factors are really difficult to define quantitatively. Thus, in the paper we propose to define the attack rate by the average per year frequency of vulnerability disclosure in the system components.

Criticality of attack is determined as an average value of basic CVSS estimation. We propose the following technique to estimate attack rate:

1) development of availability block diagram (ABD) of web-systems as a sequentially-parallel connection of components influencing on accessibility (similar to RBD);

2) extraction from NVD the vulnerability subsets for all components of ABD;

3) calculation of average per year frequency of vulnerability disclosure in these subsets;

4) determination of attack rate as the maximum of these frequencies of vulnerability disclosure;

5) calculation of attack criticality as an average value of basic CVSS estimation for selected set per year.

According with Table 1, average attack rate on DNS vulnerabilities causing unavailability could be estimated in 2013 as 1,26*10–3 1/h while the average criticality equals 6,75.

4 Web-System Availability Models for Different Vulnerability Elimination Scenarios

4.1 Initial Model and its Parameters

Let us examine a web-system based on three network services: DNS, DHCP and Routing. Reliability block diagram (RBD) and Markov's model (the marked Markov's chain) of the web-system are shown in Fig. 2.

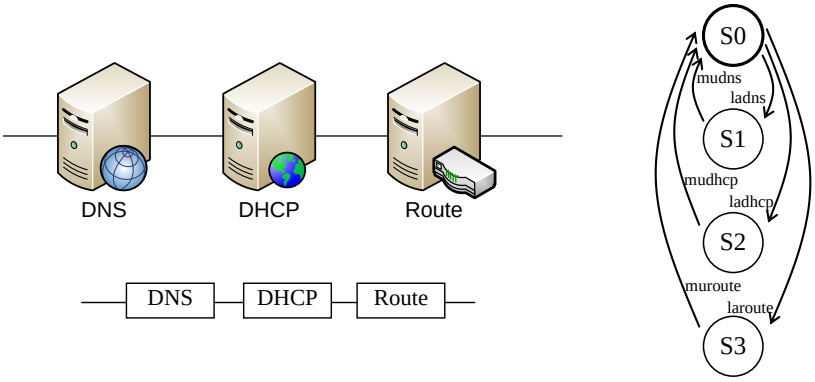

Fig. 2. Reliability block diagram and Markov's model of the web-system without considering system vulnerabilities

Table 2. Values of input parameters for availability models

#	Name	Symbol	Value	Unit
1.	DNS service software failure rate	*ladns*	3e-5	1/hr
3.	DHCP service software failure rate	*ladhcp*	1.5e-5	1/hr
4.	Route service software failure rate	*laroute*	5e-4	1/hr
5.	DNS service software recovery rate	*mudns*	0.67	1/hr
6.	DHCP service software recovery rate	*mudhcp*	1	1/hr
7.	Route service software recovery rate	*muroute*	0.33	1/hr
8.	Attack rate on availability (accessibility) of DNS service	*laatdns*	6.3e-3	1/hr
9.	Criticality of attack on availability of DNS service	*d1dns*	0.77	
10.	Restart (recovery) rate after attack on availability	*mureboot*	0.5	1/hr
11.	Restart (recovery) rate after attack on availability with VE	*murecovery*	0.22	1/hr
12.	Rate of maintenance (security audit)	*laprof*	4.5e-4	1/hr
13.	Recovery rate of service after security audit	*muprof*	0.5	1/hr
14.	Probability of successful recovery with VE	*d2p*	0.5	
15.	Probability of vulnerability elimination during security audit	*p* ($p=\alpha_1$)	0.7	

The RBD consists of three consequently connected components and failure of any components causes failure (unavailability) of the system. In this section we study two

availability models taking into account attacks on DNS vulnerabilities and different maintenance operations including security audits [7]. The first model (MA-1) corresponds to scenario with vulnerability elimination during security audits only (Fig. 1,e). The second one (MA-2) implements scenario with vulnerability elimination after successful attack on a system and also during security audits (Fig. 1,f).

Initial values of model parameters are presented in Table 2. The models itself have been implemented as Matlab programs.

4.2 The Model MA-1

This model describes a web-system with attacks on DNS vulnerabilities and periodic maintenance activities (security audits) including detection and elimination of vulnerabilities without complication of code (*ladns* =const).

Table 3. Probabilities of detection of j vulnerabilities

j	1	2	3	…	nv–1	nv
α_j	p	q*p	q^2*p	…	q^{nv-2}*p	$1-\sum\alpha_j$

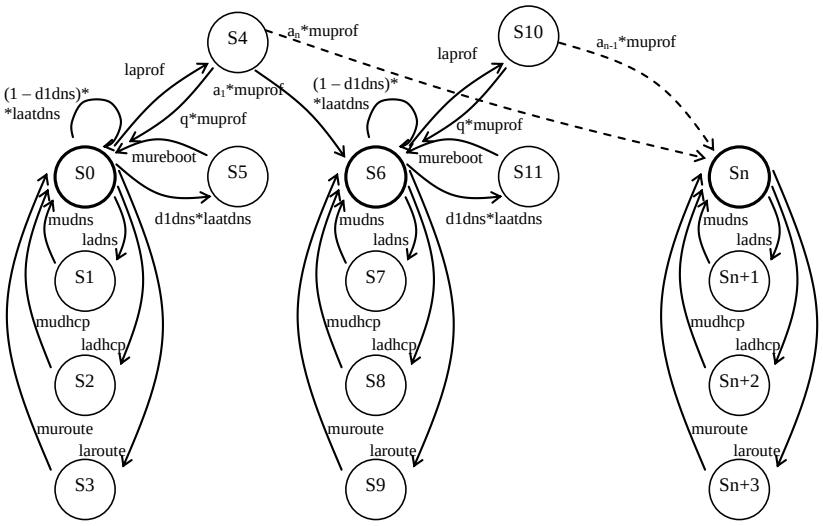

Fig. 3. Marked Markov's graph for MA-1

Marked Markov's graph is shown on Fig. 3. As during these activities it is possible to detect and eliminate more than one vulnerability [1…nv], we use a special parameter αj which defines probability of detection of j-th (j ∈ [1…nv]) vulnerabilities. Apparently, Σαj = 1, and values α1, α2,… αj,… αnv are distributes on discreet law. For calculation of geometrical distribution law αj was used with parameters: p=α1=0.7 (probability of detection of the single vulnerability) and q=1–p=0.3 (Table 3).

Initially (state S0) web-system works considering failures and recovering of DNS, DHCP и Routing services (states S1- S3). After attack on DNS (transition to state S5

with the rate d1dns*laatdns) the system fails and can be recovered by restart without vulnerability elimination with rate mureboot. Periodically maintenance activities are performed (state S4) during which 0, 1,…nv vulnerabilities can be eliminated (transitions from state S4 to states S0, S4… Sn). These transitions are weighted using parameter αj*muprof. Further process is continued in the same way (states Sn…Sn+3).

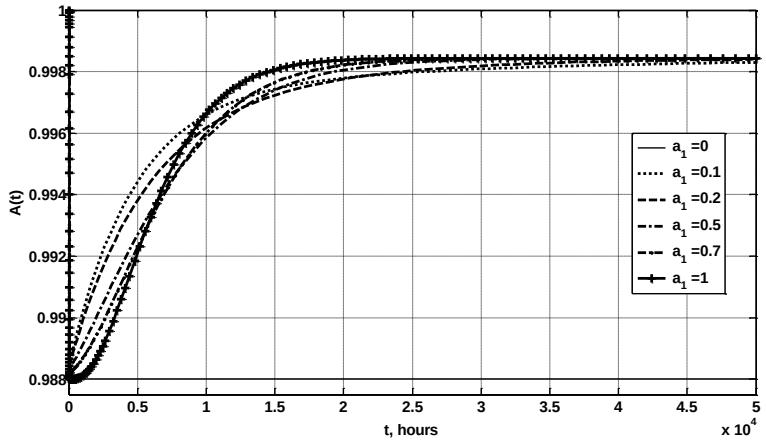

Fig. 4. Diagram of dependency of availability function for the model MA-1 on different probabilities α_1

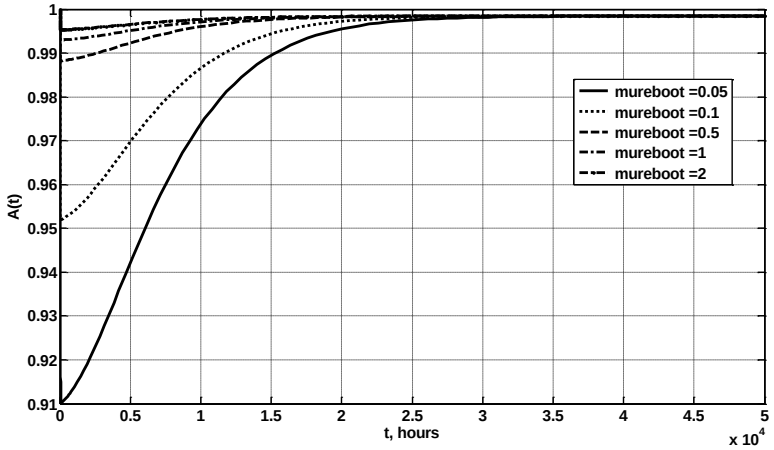

Fig. 5. Diagram of dependency of availability function for the model MA-1 on different recovery rate after attack on vulnerabilities, *mureboot*

The research results of availability function depending on parameters p=α1 and mureboot are shown on Fig. 4 and Fig. 5.

The greater value α1 causes more fast transition of the function A(t) to stationary state (Fig. 4). A value of mureboot influences on a value of availability function minimum, location of minimum on the time axis and time of transition to stationary

state (Fig. 5). If mureboot=2 (1/hour) availability function minimum equals 0,9953 for t=17 hours; if mureboot=0.05 (1/hour) availability function minimum equals 0,9103 for t=119 hours.

4.3 The model MA-2

This model describes scenarios whish in addition to MA-1 assumes detection and elimination of vulnerabilities both during security audit and right after attack (without complication of code (ladns =const). Marked Markov's graph is shown on Fig. 6.

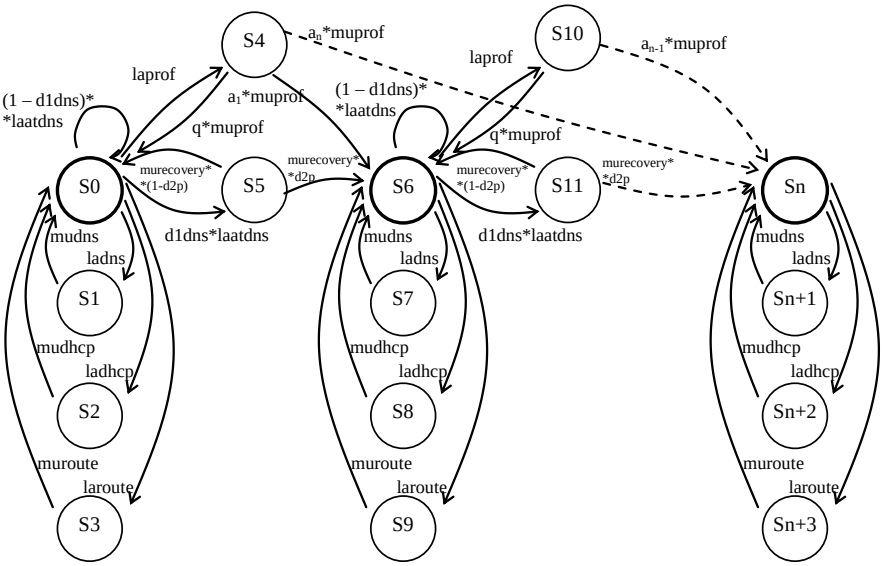

Fig. 6. Marked Markov's graph for MA-2

After attack on DNS and transition to state S5 with rate d1dns*laatdns system fails and can be recovered by restart without eliminating vulnerability with the rate (1–d2p)*murecovery or with elimination with the rate d2p*murecovery.

The results of availability function analysis depending on parameters d2p and laprof are shown on Fig. 7 and Fig. 8. The increasing of probability of vulnerability elimination d2 during maintenance activities causes more fast transition of the function A(t) to stationary state (Fig. 8). Changing the availability function depending on the rate of maintenance laprof is dual. On the one hand the rare maintenance activities are carried on the more minimum of availability function on non-stationary phase. On the other side the more often maintenance activities are carried on the faster the function transits to stationary state (Fig. 8).

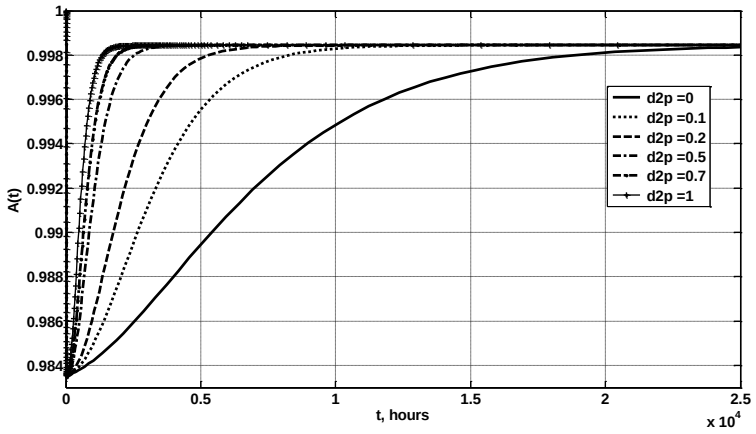

Fig. 7. Diagram of dependency of availability function for the model MA-2 on different probabilities of vulnerability elimination after attack *d2p*

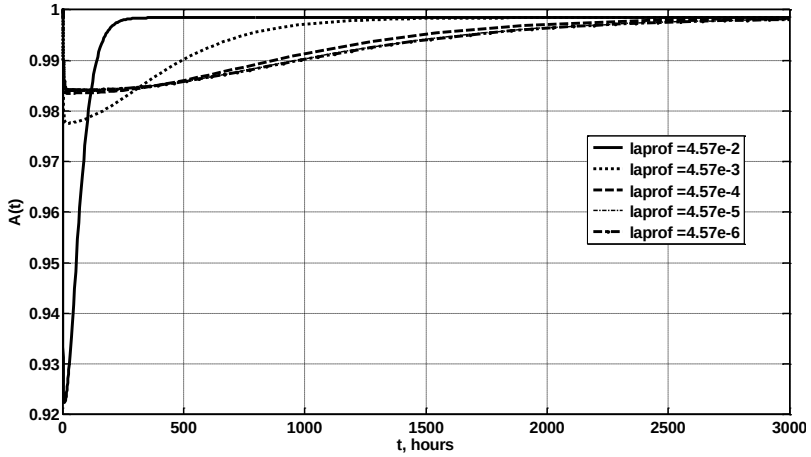

Fig. 8. Diagram of dependency of availability function for the model MA-2 on rate of maintenance *laprof*

4.4 Combining of the Models MA-1 and MA-2

The scenarios corresponding to the models MA-1 and MA-2 can be superposed to increase availability due to increasing of minimum and duration of system transition to the stationary state of availability function. To combine these two scenarios we have developed a set of Matlab programs. Filing of coefficient matrixes was done according with the same initial data (Table3). To solve systems of Kolmogorov-Chapman's differential equations the method *ode15s* for time span [0…20000] hours. The results of solving are shown on the Fig. 9.

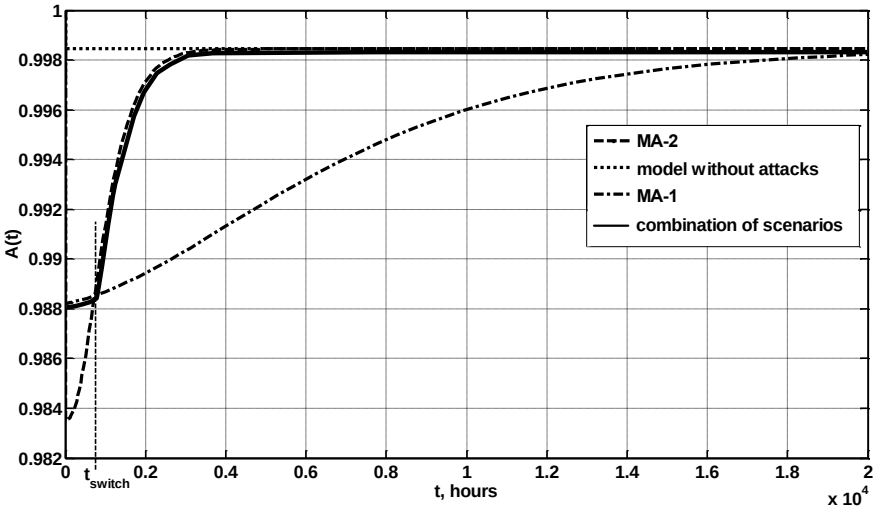

Fig. 9. Combining of the models MA-1 and MA-2 (solid line)

According to Fig. 9, the vulnerability elimination scenario MA-1 is better to use till tswitch =750 hours, after this time the scenario MA-2 ensures better availability. Hence at the beginning recovering a system after attack (without vulnerability elimination) is preferable. Then, taking into account increase of the number of failures caused by attacks other scenario (when vulnerabilities are detected and eliminated both after attacks and during maintenance) becomes preferable. It allows increasing the value of availability from 0.984 (MA-2) to 0.988 (MA-1) and decreasing time transition to stationary state from 20000 (MA-1) to 3000 (MA-2) hours.

5 Conclusions

We analyzed a set of web-system behavior scenarios in conditions of attacks on component vulnerabilities. Quantitative assessment and research of availability for such systems can be based on Markov's models using statistic data about vulnerabilities contained in open databases and described sequence of evaluating of attacks rates and criticality.

We proposed and discussed two models of web-system availability considering attacks on DNS vulnerabilities and different scenarios of vulnerability elimination. There is possibility and reasonability of scenario changing taking into account values of availability function allowing increase minimum one at the non-stationary stage and decrease time of transition to stationary state. This approach allows selecting VE scenario to improve resilience of web-system.

The future research efforts may be concentrated on development of integrated strategies for maintenance and security policies selection taking into account physical, design and interaction faults, and implementation of dynamically reconfigurable web- and cloud-systems with embedded monitor and solver to select the optimal strategy of

maintenance.

Besides, other types of the vulnerabilities for confidentiality and integrity issues and more detailed model taking into account routing processes can be researched.

References

1. Dong Seong Kim, Machida, F., Trivedi, K.S.: Availability Modeling and Analysis of a Virtualized System. In: 15th IEEE Pacific Rim International Symposium on Dependable Computing, pp.365--371, IEEE Press, Shanghai (2009)
2. Zheng Wu, Yang Ou, Yujun Liu: A Taxonomy of Network and Computer Attacks Based on Responses. In: International Conference on Information Technology, Computer Engineering and Management Sciences, pp.26-29, IEEE Press, Nanjing (2011)
3. Roy, A., Dong Seong Kim, Trivedi, K.S.: Cyber security analysis using attack countermeasure trees. In: Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research (CSIIRW '10), pp.1--4, ACM, New York (2010)
4. Ping Wang, Jia-Chi Liu: Threat Analysis of Cyber Attacks with Attack Tree+. Journal of Information Hiding and Multimedia Signal Processing 5(4), 778--788 (2014)
5. Alaa Mohammed Abdul-Hadi, Ponochovny, Y., Kharchenko, V.: Development of basic Markov's model research availability of commercial web services. Radioelectronic and computer systems (64), 186-191 (2013)
6. Kharchenko, V., Alaa Mohammed Abdul-Hadi, Boyarchuk, A., Ponochovny, Y.: Web Systems Availability Assessment Considering Attacks on Service Configuration Vulnerabilities. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) Advances in Intelligent Systems and Computing. vol.286, pp. 275--284, Springer International Publishing, Switzerland (2014)
7. Nicol, D., Sanders, W., Trivedi, K.S.: Model-based evaluation: from dependability to security. IEEE Transactions on Dependable and Secure Computing 1(1), 48-65 (2004)
8. Trivedi, K.S., Dong Seong Kim, Roy, A., Medhi, D.: Dependability and security models. In: Proceedings 7th International Workshop on the Design of Reliable Communication Networks (DRCN 2009), pp. 11-20, IEEE Press, Washington, DC (2009)
9. Kizza, J M.: Guide to Computer Network Security. 2nd edition. Springer, London (2013)
10. Al-Kuwaiti, M., Kyriakopoulos, N., Hussein, S.: A comparative analysis of network dependability, fault-tolerence, reliability, security, and survivability. IEEE Communications Surveys & Tutorials 11(2), 106--124 (2009)
11. NVD - Advanced Search, http://web.nvd.nist.gov/view/vuln/search-advanced
12. NVD - Data Feeds, http://nvd.nist.gov/download.cfm#XML
13. Recommendation X.1521. Common vulnerability scoring system. ITU-T, Geneva, The Switzerland (2012)