EVALUATING THE SURVIVABILITY AND SECURITY OF COMPLEX WEB SYSTEMS

Costel CIUCHI[1], Angelica BACIVAROV[2], Ioan BACIVAROV[3], Laura IANCU[4]

The strategy of an organization should include as a main objective the insurance of an optimal performance level for its information systems, which implies the need to define their core capabilities and fundamental quality attributes. The early development, since the design phase, of specific performance evaluation techniques and survivability capacities in different critical situations (attacks) for system operation, adds to the managerial decision-making process a powerful tool for maintaining the system to the expected performance level. This article analyses the survivability (ability to survive) of an application that uses Web technology in a 3-tier architecture through systematic evaluation, at different levels, of the availability in terms of cyber-attack and survival properties of the application.

Keywords: strategy, decision, complex systems, infrastructure, Web design architecture, security, software security, survivability

1. Introduction

The technological complexity, the large area of distribution of data and information, and the large number of threats and incidents to the security of a computer system are factors to be considered when developing an information system. Designing systems without taking into account the above mentioned factors generates a delay in the management of the system, making it practically impossible to conduct a proper decision making process to reduce security risks.

In recent years it was found that the number of failures and incidents is in constant growth; a strong interest for the study and development of survivability system was therefore revealed.

The security concerns acknowledged by service providers and manufacturers include software and hardware errors, software bugs, attacks, human error operation / maintenance and natural disasters. The ability of a system

[1] PhD student, Faculty of Electronics, Telecommunications and Information Technology (ETTI), University POLITEHNICA of Bucharest, Romania, e-mail: ciuchic@yahoo.com
[2] Prof., EUROQUALROM Laboratory, Faculty of ETTI, University POLITEHNICA of Bucharest, Romania, e-mail: angelica@euroqual.pub.ro
[3] Prof., EUROQUALROM Laboratory, Faculty of ETTI, University POLITEHNICA of Bucharest, Romania, e-mail: bacivaro@euroqual.pub.ro
[4] PhD student, Faculty of ETTI, University POLITEHNICA of Bucharest, Romania, e-mail: laura_yln@yahoo.com

to continue providing services (availability) in the presence of threats or vulnerabilities (flaws, attacks etc.) represents the survivability of that system.

The growing impact of unavailability of applications and services, with implications for the public safety and commercial transactions, has made the assessment of availability and performance in presence of undesirable events an essential step for system testing and validation. Consequently, two requirements were emphasized as indispensable for most systems: business continuity and information security. The coordination of a decision-making process on providing security for complex network systems is difficult in terms of technological diversity or large number of users. Without enhancing security, the cost of protection may be much higher and more substantial, while separate use of equipment and security solutions can "generate" new security gaps.

2. About survivability

The main requirement for survivability is the ability of a system to provide essential services and preserve its main associated properties, even if several components of the system are in a state of failure. Survivability requirements may differ considerably depending on the purpose and the mission of the system, on the critical situations and subsequent consequences in case of failure and service interruption. Defining and analyzing survivability requirements by encompassing all aspects related to the use, development, operation and evolution of the system is a first step towards the development of the survivability attributes of a system.

The requirements of survivability attribute for various systems are different, depending on the purpose, use, development and evolution of the system, as well as on the extent of the consequences of a failure or service interruption. Survivability focuses on providing key services and preserving the essential components of the system. Basic services and components are critical system capabilities to meet the mission’s objectives.

Survivability is defined by three key elements: **resistance**, **recognition** and **recovery** (Table 1) [1]. The development of a system that meets for an extended period of time the 3 properties of survivability at once is difficult due to the continuous emergence of new threats and security breaches. Adapting and learning from previous attacks is one of the most important strategies for updating the protection mechanisms of a system.

The identification of critical services and the maintenance of an optimal capacity of their delivery emphasized four fundamental elements depicting survivability.

Table 1

System survivability strategies

PROPERTY	DESCRIPTION	STRATEGIES
RESISTANCE to attacks	Strategies to reject attacks	• Log in • Access control • Encryption • Filter messages • Systems diversification • Functional isolation
RECOGNITION of attacks and evaluating the damage	Strategies for detecting attacks and damage assessment	• Intrusion detection • Integrity check
Complete RECOVERY of essential services after attack	Strategies for limiting damage, compromising or functional information recovery, maintenance or recovery of essential services within time constraints, full recovery service	• Redundant components • Data duplication • System backup and restore • Continuity plans
ADAPTABILITY and evolution of the system (*lessons learned*)	Survivability strategies for improving the system based on knowledge gained from previous intrusions	• Recognition of new intrusion signatures

For each specific property, a number of survivability strategies is identified that can be adopted and applied in order to neutralize the threats of attack on a system.

The most difficult part of survivability is to create a system as robust as possible to withstand attacks or intrusions which are not known. Because attackers are improving their attack models and are constantly looking for possible security breaches, system administrators need to build a defense based on previous attacks experience as well [2], in order to be able to anticipate those potential directions where the attacks may come from.

3. Survivability, security and dependability

Security is generally defined as a combination of characteristics of availability, confidentiality and integrity, focusing on "recognizing attacks" and "resistance to attacks." Survivability concept is broader than security concept and focuses on "adapting and overcoming the attack" [3].

A parallel between the concepts of dependability and survivability was developed in [4].

The main features of survivability in relation to the two main attributes of systems (dependability and security) are:

- wide range of failures handling (from attacks to natural disasters);

- transitory behavior of the system, just after a fault has occurred (attack or natural disaster) until the system stabilizes itself (through restoration or recovery) [5].

Table 2

Parallel between the concepts of dependability and survivability

	DEPENDABILITY	SURVIVABILITY
OBJECTIVE	1) The ability to deliver services that can justifiably be considered reliable; 2) The ability of a system to avoid frequent or severe failures in a way that is acceptable to the user(s).	Ability of a system to fulfill its mission even if some of its components are in state of failure.
THREAT	1) Design faults - defects in software, hardware errors (errata), malicious acts; 2) Physical defects - manufacturing defects, physical damage; 3) Defects in interaction - interferences, improper or inconsistent data entry, attacks, including viruses, worms and intrusions.	1) Attacks (e.g. intrusions, attack attempts, denial-of-service); 2) Errors (due to internally generated events, such as software design errors, hardware degradation, human errors, corrupted data); 3) Accidents (externally generated events such as disasters).

In the context of cyberspace security, a **threat** can be defined as the presence of a potential event that could have negative effects in violation of a security mechanism. Information systems have as sources of failure two main causes: natural phenomena and human actions.

In IT (Information Technology) field, a security incident may be associated with any action taking place within a system (including those related to the performance of any component, network, calculation system, software etc.) which may compromise integrity or cause the loss of confidentiality of the respective component (network, computer, applications and services, data).

In addition to traditional threats in the field of IT, an important form of computer incident is the software attack, as an induced or accidental act.

Cyber-attacks are a serial of actions performed by an attacker to obtain an unauthorized result. An attacker is using **means** to exploit a **vulnerability** in order to perform an **action** on a **target** with the aim to obtain an **unauthorized result**. To be successful, an attacker must find ways (attacks) that can facilitate access through repeated trials or by forcing the security systems (brute force). **Means and vulnerabilities** are used to cause an **event** in a computer system.

Cyber-attacks have as result the loss of integrity, confidentiality and data availability, in violation of security policies and security systems; the most recurrent are user account violation, administrator roles usurpation, data capture packets, service denial, deceiving, malicious programs use, attacks on infrastructure.

Several analysis techniques were developed for evaluating attacks or for defense, such as Attack Graphs [6][7], Attack Trees [8][9], Attack-Defense Tree

(ADTree) [10], Stochastic Petri-Net [11], Reliability Block Diagram (RBD) [12], Waiting Networks and Continuous-Time Markov Chains (CTMC) [13].

The above techniques were used to evaluate server architectures and software architecture of data flow related to reliability, availability and performance.

4. System modeling - Web-based systems approach with 3-tier architecture

The 3-tier architecture is a client-server architecture where the logic operation model, data access, database and user interface are developed and optimized as independent modules on different hardware and software platforms. This type of architecture arose from practical considerations software design process, being a fundamental framework for modeling logical systems and becoming a basic model in software development.

The basic components of this architecture are [14][15]: the **Presentation** level (the user service level provides access to the application), the **Logic** level (realizes process and data modeling) and the **Data** level (interacting with data from databases or data warehouses). Architectures with more than 3 levels are generally called *n-tier type architectures*.

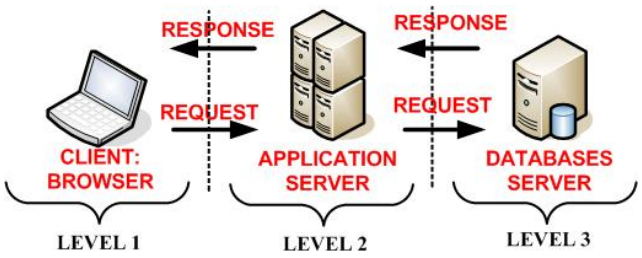

Fig. 1. 3-tier architecture in Web technology

If development is based on Web technologies, 3-tier architecture is used especially in e-commerce applications. In general, applications using 3-tier architecture have the following structure:

1. one user interface level: in Web-based applications, the interface (front-end) represents the content extracted by a specialized program (browser);

2. an intermediate level of processing and generating content achieved through various technologies such as Ruby on Rails, Java EE, ASP.NET, PHP, ColdFusion, Perl, a Web server providing static or dynamic content distribution;

3. a support level of database or data warehouse including data collection, RDBMS (*Relational Database Management System*) that manages and provides data access.

Projecting an additional level meets the operation needs and the use of distributed type applications on a wide area. The main benefits of tiered models of N-tier/3-tier type are [16]:

- Maintainability - each level is independent of the others; updates or modifications can be made without affecting the entire application;
- Scalability - levels are based on implementation in layers; scaling an application is quite simple;
- Flexibility - each level can be managed or scaled independently; flexibility is high;
- Availability - applications can exploit the modular architecture through the use of scalable components, which improve availability.

5. Survivability evaluation model of an information system (Web applications)

The development of applications using Web-based technology has impressively evolved over time by using client-server architectures, especially the 3-tier model that allowed separate-type implementation of different components by levels of work.

In this paper we will elaborate on the study of survivability for a Web-based application organized along the lines of the 3-tier architecture where each level is independent and is implemented for simulation on different virtual machines.

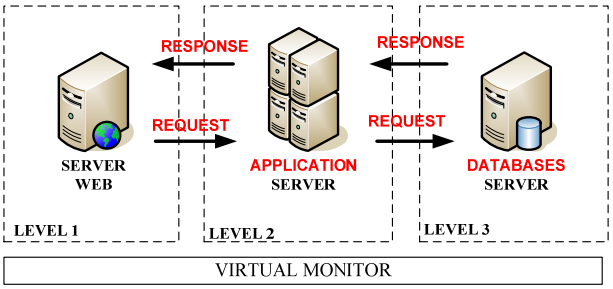

Fig. 2. 3-tier architecture model for a Web-based application

In a 3-tier architecture model, each level of the architecture and its function within the architecture (service) can be installed on different virtual machines. A level can have associated several virtual machines installed for providing the same service or different services; the number of virtual machines is usually fixed by the importance and complexity of the service provided by the respective level. Each level can have a number of virtual machines for different

purposes (backup, load balance, and so on). There is also a possibility to have installed on a certain level several services on the same machine.

Survivability calculation for the whole system is made using the algorithm proposed in [17], combined with a computational model developed for each level [18]. The modeling of survivability by level calculation brings a very important contribution for collecting realistic data on survivability system.

Model analyzes a system's response to incidents and defines a general method to evaluate the survivability of a system [17]. Thus, survivability is defined as the degree of resistance of the system when an attack occurs, and its capacity to provide services at a certain level in the new weakened state, after the attack. This new state S is, in general, a compromised state, when the system stops before any recovery or reparation attempts to return to the normal state.

At conceptual level, survivability will be calculated as [17]:

SURV = (level of performance in state s) / (normal level of performance)

We consider **P(s, k)** as the degree of survivability of service (P), where **k** service has survived in state **S**, and **w(k)** is the level of importance of the service. Thus, we can consider the possibility to express survivability function as the expression of:

$$SURV(s) = \sum_{k} w(k) * P(s,k) \tag{1}$$

where $0 \leq w(k) \leq 1$ and $\sum_{k}[w(k)] = 1$.

Survivability of a system can be defined by two essential attributes of security: **availability** (the system can respond to all requests) and **integrity** (responses meet functional specification of the system and all its components are in normal operation). Web architecture requires 3 servers' levels: Web Server, Application Server and Database Server.

When a system is under attack, as long as the system can respond (availability) and the responses are correctly generated by the three levels (integrity), we can say that the system can survive the attack; otherwise, the whole system is compromised.

Due to the probabilistic nature of different states through which the system evolves, one can accept that the evolution of the process is described by a random process. The evolution of the respective process is defined by a set of variables describing the development process.

To know the various states of the system in consecutive periods t_1, t_2,...,t_n, prior to period t, helps to find out the state at time t by collecting information on the condition of the system during previous periods, but all included in the latest state, that is t_n respectively.

It should be noted here that, generally, a system can reach a certain state after crossing several successions of states, the way how that system reached the respective state influencing its subsequent operation, and therefore also the indicators of the reliability of the system at time t_n. Such a development process characterized by the fact that the state that will be reached by the system depends on its initial state, as well as on the way the system got into this state, is called a *Markov process*.

The study of behavior and implicitly of survivability of the system is made in terms of attacks on the system, for a constant number of attacks and a large number of consecutive attacks, using 2 variables: *attack success rate* (compromise) and *attack response rate* (recovery). Both variables, compromise rate and recovery rate, can be modeled by a Poisson probability distribution. Based on the above assumptions, the system's state possibilities become a finite one, of Continuous-Time Markov Chain type [18].

We assume that the system has **n** installed services and each level of 3-tier architecture (application) has implemented a service (provided by a virtual machine). We consider that services at all 3 levels of the application architecture behave identically under attack, regardless of the type of service they provide (Web server, application, and database). In these conditions, the probability of penetrating the system through a single vulnerability, noted by $\mathbf{P_b}$, is the same for all levels. Then survivability can be expressed as: $P_s = 1 - P_b$ and given the considerations of [18]:

$$P_b = \frac{n}{|S|} \tag{2}$$

where **n** is the number of services on that respective level and **S** - the number of states associated to all n services of a level.

Associated states of a service can be: normal, under attack, compromised, recovering or inoperable.

$$P_s = 1 - \sum_{i=1}^{m} (P_b)^i (1 - P_b)^{m-i} = 1 - \sum_{i=1}^{m} (\frac{n}{|S|})^i (1 - \frac{n}{|S|})^{m-i} \tag{3}$$

$$P_s = 1 - \sum_{i=1}^{m} (\frac{n}{\Pi_{j=1}^{k} |S_j|})^i (1 - \frac{n}{\Pi_{j=1}^{k} |S_j}})^{m-i} \tag{4}$$

where **n** nodes can have no more than |S| variations and **m** is the total number of intrusion attempts, provided that m>1 (at least one service must be compromised).

6. Analysis / results - impact of attacks on a Web-based application

Survivability analysis for a level of a 3-tier application can be achieved in the following situations, static (in the conditions of a constant number of attacks on the application) and dynamic (a growing number of attacks on the application).

Static analysis involves the consideration of a constant number of incidents (attacks) and the survivability calculation is done in terms of system architecture and depending on how this is influenced by the number of services performed on a certain level of a Web application. A single attack cannot affect more than one virtual machine, regardless of the technique used and the number of services per level. Once a service is compromised, it is confined to its assigned virtual machine and can be recovered by restarting the system. The defense mechanism of the virtual machine has its own procedure to recover the virtual machine and reduces the number of restarts of the system depending on the severity of the attack. A level is compromised when all virtual machines or all installed services on a virtual machine on that level are in inoperable condition.

Static Calculation

Depending on the security level chosen for the application, in terms of quality attributes (confidentiality, integrity and availability), the importance of different levels is determined by the general security level assessed for the entire application, as well as by the purpose, applicability area and criticality of services at every level. Survivability for Web applications of 3-tier architecture type is a combination of architectural model proposed in [18] and the calculation algorithm proposed in [17].

Based on a minimum number of services for each level of the 3-tier architecture (at least one service) the calculation of survivability can be made according to the total number of services in the system:

- individual calculation of survivability for the 3 levels of 3-tier architecture using (4);
- calculation of the survivability of the system based on the significance of the level using (1).

Survivability calculation for each level of Web application with a minimum number of N = 1 services / each level, ranging the number of services from a single level, with S = 50 and m = 5:

Table 3

Number of services per level: NIV1 = 1…10 services, NIV2=NIV3 = 1 service

Surv(NIV1)	0.903	0.815	0.733	0.659	0.590	0.527	0.470	0.418	0.370	0.327
Surv(NIV2)	0.903	0.903	0.903	0.903	0.903	0.903	0.903	0.903	0.903	0.903
Surv(NIV3)	0.903	0.903	0.903	0.903	0.903	0.903	0.903	0.903	0.903	0.903

To calculate survivability for the whole system (application) using (1):

$$SURV(s) = \sum_{k} w(k) * NIV(s,k) \quad (5)$$

The level of importance of services for a system can have the following values: w_k=0.13, 0.33, 0.53.

Table 4

Level of importance of services for a system

	LEVELS IMPORTANCE	IMPORTANCE
w1	[w_{NIV1}=0.13, w_{NIV2}=0.33, w_{NIV3}=0.53]	DATA INTEGRITY, w_{NIV3}= database level
w2	[w_{NIV1}=0.33, w_{NIV2}=0.53, w_{NIV3}=0.13]	CONFIDENTIALITY, w_{NIV2}= application level
w3	[w_{NIV1}=0.53, w_{NIV2}=0.13, w_{NIV3}=0.33]	AVAILABILITY, w_{NIV1}= Web server level

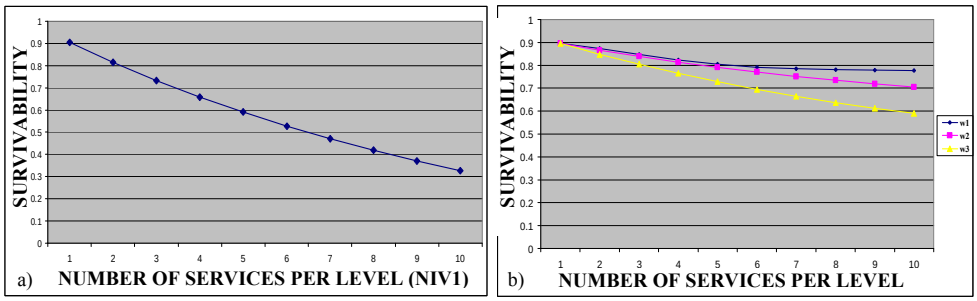

Fig. 3. Survivability evolution: a) in case of increase of services in a certain level, b) depending on the importance of the system levels

Remarks:

- the level that has several services, but less importance, has a better survivability (w1) comparing to the situation when on a highly important level more services are located (w2);

- when the number of services on one level rises (is higher), the likelihood of a successful attack from a single test is growing. Thus, if the number of services on one level increases, survivability decreases.

Dynamic calculation

To calculate survivability, we will consider the case of a variable number of attacks. The transition graph of the states associated to the system (to the application) is:

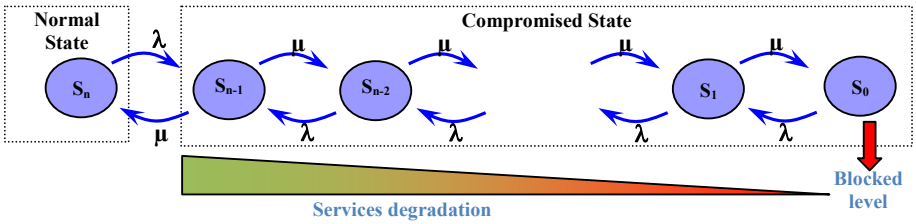

Fig. 4. System state transition graph

According to the architecture graph state, the attacker breaks up services, one by one, with a **λ** compromise rate. The system can retrieve services on a virtual machine (by restarting the system, the service or other approaches), with a **μ** recovery rate.

State $\mathbf{S_i}$, where 0<i<n, indicates that the system has **i** uncompromised services and **n-i** compromised services. In the application architecture proposed above, once a service is compromised, the integrity of the service is compromised too; thus, the only normal state of the system is $\mathbf{S_n}$, if the system is not compromised.

CTMC (Continuous-Time Markov Chain model)

The process that has an evolution characterized by the fact that the state which the system will reach depends on its actual state, as well as on how the system got into this state, is called *Markov process*. If we assume that the λ compromise rate and the μ recovery rate fulfill the conditions of a Poisson distribution, the transition states of the system become a model of Continuous-Time Markov Chain model [13] which can be determined using the matrix of transition probabilities of states Q = (q_{ij}), where:

- q_{ij} is the transition rate from state i to j, with the probability of leaving state i:

$$q_{i,i} = -\sum_{j \neq i} qi, j \tag{6}$$

- the initial state probability vector (normal state) has the following form π(0) = (0, 0, ..., 1).

State transition probability matrix associated with the system:

$$Q = \begin{array}{c} \\ G_0 \\ G_1 \\ \vdots \\ G_n \end{array} \begin{array}{c} \begin{array}{cccccc} G_0 & G_1 & G_2 & \dots & G_{n-1} & G_n \end{array} \\ \begin{pmatrix} -\mu & \mu & 0 & \dots & 0 & 0 \\ \lambda & -\lambda-\mu & \mu & \dots & 0 & 0 \\ \vdots & & & \dots & & \vdots \\ 0 & 0 & 0 & \dots & \lambda & -\lambda \end{pmatrix} \end{array} \tag{7}$$

By using Continuous-Time Markov Chain model, the equilibrium state as well as the transition state of the system can be calculated. The equilibrium state of a system is the state where all features of the system do not change after a long period of operation.

Probability vector in equilibrium state fulfills the following conditions:

$$\begin{aligned} \pi Q &= 0 \\ \sum_j \pi_j &= 1 \end{aligned} \tag{8}$$

$$\pi Q = [\pi_0 \quad \pi_1 \quad \pi_2 \quad \pi_3] \begin{bmatrix} -\lambda & \lambda & 0 & 0 \\ \mu & -(\lambda+\mu) & \lambda & 0 \\ 0 & \mu & -(\lambda+\mu) & \lambda \\ 0 & 0 & \mu & -\mu \end{bmatrix} = 0 \tag{9}$$

$$\pi_0 + \pi_1 + \pi_2 + \pi_3 = 1 \tag{10}$$

CASE 1: λ=1 - the attack rate is constant over the entire level; μ=1...10 - recovery rate (the number of defense mechanisms); n=10 - the number of services.

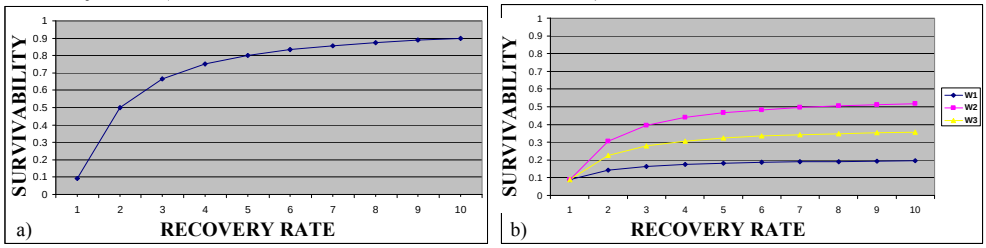

Fig. 5. a) Survivability evolution on a level and b) System survivability evolution in case of an increasing number of defense mechanisms on a certain level

The levels of importance of different services within a system have the same values from Table 4.

While the recovery rate (number of defense mechanisms) μ increases, survivability has an increasing trend for all levels of the system, regardless of their related importance. The lowest survivability factor is met where the first level has a high importance, regardless of the number of associated security mechanisms, as this is the most exposed level, and therefore its survivability is the lowest.

CASE 2: λ =1...15 - variable attack rate over the entire level; μ=5 - recovery rate (the number of defense mechanisms); n=10 - the number of services.

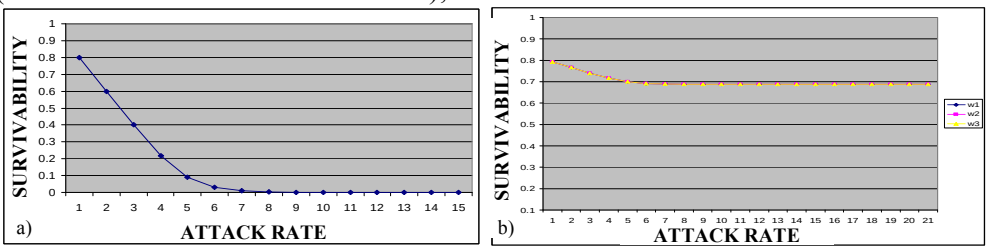

Fig. 6. a) Survivability evolution on a level and b) System survivability evolution in case of a variable number of attacks on a certain level

The levels of importance of different services within a system have the same values from Table 4.

Regardless of the importance associated to a level within the architecture of the system and of the number of defense mechanisms located on a certain level, in case of sustained attacks, survivability decreases equally throughout the whole system.

7. Conclusions

The modeling software of a Web application has a major impact on performance and security. The development of testing methodologies, complemented by a proper experimental basis, will support a coherent and realistic assessment of defense mechanisms projected to reduce widespread attacks. Due to the diversity of survival features, this approach is used to get quantitative measures to approach the survivability skills, but also to provide an insight into modeling the behavior of a system.

In this paper we have developed, using the analytical methods proposed in [17] and [18] for an application in Web technology 3-tier architecture, a survivability study model for a critical service of a system and how to ensure for the base system a decision support framework for the survivability attribute.

This approach can be used to quantitatively compare survival characteristics for different architectures and can be extended to a wide range of systems with different degrees of complexity. Future research will include the extension of the methodology by:

- applying various hybrid architectures;
- diversifying responses to the application by considering some specific defense mechanisms for each level in case of attack;
- extending the study from 3-tier to n-tier model.

An alternative research direction would require significant progress to be made in modeling network attacks in addition to studying interactions between attacks and their context, as well as the afferent defense technology, topology, protocols and applications used.

R E F E R E N C E S

[1] *R.J. Ellison et al.,* "Survivable Network Systems: An Emerging Discipline", Tech. Report CMU/SEI-97-TR-013, Pittsburgh, Penn., Software Engineering Institute, Carnegie Mellon University, Nov. 1997 (revised May 1999)

[2] *I. Bacivarov and I. C. Mihai,* "The Survivability Analysis of the Informational Systems", Proceedings of the 11th International Conference of Quality and Reliability – CFF2008, Sinaia, 24-26 September, 2008, ISSN: 1842-3566, pp. 151-158

[3] *R. J. Ellison, D. A. Fischer, R. C. Linger, H. F. Lipson, T. Longstaff, and N. R. Mead*, "Survivable network systems: an emerging discipline. Technical report", Technical Report CMU/SEI-97-TR-013, November 1997, revised May 1999

[4] *A. Avizienis, J.-C. Laprie, and B. Randell*, "Fundamental concepts of computer system dependability", IARP/IEEE Workshops on Robot Dependability: Technological Challenge of Dependable Robots in Human Environments, Seoul, Korea, May 2001
[5] *Y. Liu and K. S. Trivedi*, "Survivability Quantification: The Analytical Modeling Approach", Int. Journal of Performability Engineering, **volume** 2(1), 2006, pp. 29-44
[6] *R. E. Sawilla and X. Ou,* "Identifying critical attack assets in dependency attack graphs", ESORICS '08: Proceedings of the 13th European Symposium on Research in Computer Security, Berlin, Heidelberg, Springer-Verlag, 2008, pp. 18–34
[7] *L. Krautsevich, F. Martinelli, and A. Yautsiukhin*, "A general method for assessment of security in complex services", Proceedings of 4th European Conference ServiceWave, Springer, 2011
[8] *Bruce Schneier*, "Attack Trees", Dr. Dobb's Journal of Software Tools 24, (December 1999), pp. 21–29
[9] *S. Mauw and M. Oostdijk*, "Foundations of Attack Trees", ICISC, **volume 3935 of LNCS**, Dongho Won and Seungjoo Kim, editors, Springer, ISBN 3-540-33354-1, 2005, pp. 186–198
[10] *Barbara Kordy, Marc Pouly, and Patrick Schweitzer*, "Computational Aspects of Attack-Defense Trees", Security & Intelligent Information Systems, **volume 7053 of LNCS**, Springer, 2011, pp. 103–116
[11] *M. A. Marsan,* "Stochastic Petri nets: an elementary introduction", 1990, pp. 1-29
[12] *R. A. Sahner, K. S. Trivedi and A. Puliafito,* "Performance and reliability analysis of computer systems: an example-based approach using the SHARPE software package", Kluwer Academic Publishers, Norwell, MA, USA, 1996
[13] *H.C. Tijms,* "Stochastic Models", Wiley & Son, New York, USA, 1994
[14] www.msdn.microsoft.com/en-us/library/windows/desktop/ms685068(v=vs.85).aspx
[15] *Gary P. Schneider,* "Electronic Commerce", 3rd Annual Edition, Thomson Learning, 2002
[16] www.msdn.microsoft.com/en-us/library/ee658117.aspx
[17] *Soumyo D. Moitra, Suresh L. Konda, Joanne E. Spriggs*, "A Simulation Model for Managing Survivability of Networked Information Systems", 2000
[18] *Meng Yu, Alex Hai Wang, Wanyu Zang, and Peng Liu*, "Evaluating Survivability and Costs of Three Virtual Machine based Server Architectures", Proceedings of 5th International Conference on Security and Cryptography (SECRYPT), July 2010, Athens, Greece