Security Assessment of Web Based Distributed Applications

Catalin BOJA[1], Mihai DOINEA[2]
[1] Academy of Economic Studies, Bucharest, Romania
[2] University of Amsterdam, Amsterdam, The Netherlands
catalin.boja@ie.ase.ro, l.m.doinea@student.uva.nl

This paper presents an overview about the evaluation of risks and vulnerabilities in a web based distributed application by emphasizing aspects concerning the process of security assessment with regards to the audit field. In the audit process, an important activity is dedicated to the measurement of the characteristics taken into consideration for evaluation. From this point of view, the quality of the audit process depends on the quality of assessment methods and techniques. By doing a review of the fields involved in the research process, the approach wants to reflect the main concerns that address the web based distributed applications using exploratory research techniques. The results show that many are the aspects which must carefully be worked with, across a distributed system and they can be revealed by doing a depth introspective analyze upon the information flow and internal processes that are part of the system. This paper reveals the limitations of a non-existing unified security risk assessment model that could prevent such risks and vulnerabilities debated. Based on such standardize models, secure web based distributed applications can be easily audited and many vulnerabilities which can appear due to the lack of access to information can be avoided.

Keywords*: Security, Risks and Vulnerabilities, Distributed Applications, Audit Process*

1 Web-based distributed applications

Since earliest stages of distribute systems, auditing was used to prevent unwanted intrusion into the systems or other kinds of attacks [1]. The exposure of distribution in the sense of various points of accessing resources has made from this field an uncertain bay of processes which could easily be exploited.

Web-based distributed applications represent complex software applications based on client-server architecture that deliver information and services using HTML and XHTML visual interfaces. This type of applications integrates various components:

- multimedia elements like sound, video and interactive media clips;
- server-side processing routines developed in various programming languages like ASP, ASP.NET, PHP, JSP or CGI scripts;
- client-side processing routines that use JavaScript syntax;
- logical structures that define the framework in which input data are processed in order to give needed results;
- network structures that defines different roles for various application components in order for the system to function properly.

In a Web-based distributed applications the user has access to remote resources through client components implemented by the application. The resources reside and are distributed on other machines and this remote processing mode is transparent for users.

A web-based distributed application is broken up into several components, accordingly to three-tier architecture, as described in figure 1:

- the presentation layer is the client, which is represented by a browser application; at this level, the client displays data received from the sever using XHTML forms and also accepts user input and sends it back to the server; using JavaScript scripts or Java applets, some processing can be done at this level;
- the application layer handles the processing between the client and the server; the server, Web server, Content server, Streaming media server, FTP or Email server provides access to different services and resources; the core component of a Web-based distributed application is the Web Server which is a service that serves up web content; typically, this service listen on port 80 for Hyper Text Transfer Protocol (HTTP), RFC 2616, or 443 for Hyper Text Transfer Protocol Secure (HTTPS), RFC 2818;

- the data layer separate data processing and offers access to data through different services;

Many distributed applications use in the development stage a Web-based application type framework. The reason is given by the:

- extended variety of instruments, programming environments and languages, techniques and methods used on a large scale;
- open software technologies that reduce the costs for proprietary tools;
- great number of on-line communities and free code libraries that reduce the cost of development from start;
- easiness to combine multimedia components into an application.

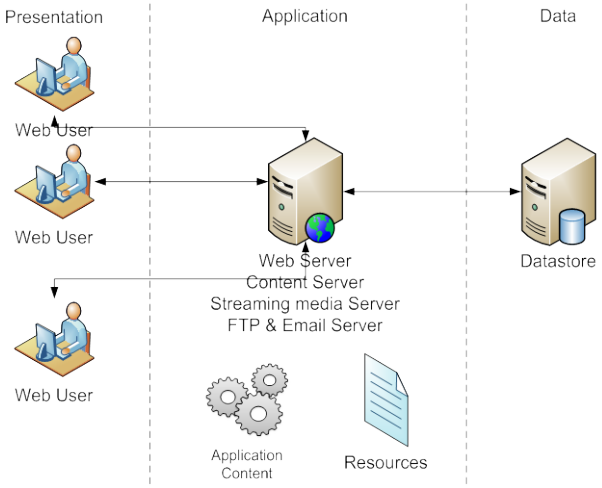

Fig. 1. Three-tier architecture for Web-based distributed applications

There are many criteria to classify Web-based distributed applications. Taking into consideration the structural viewpoint, largely described in [2], these applications can be classified in:

- Web-based distributed applications with linear structure;
- Web-based distributed applications with tree like structure;
- Web-based distributed applications with graph structure.

Linear Web-based distributed applications are constructions formed from components that are launched in execution one after another. A component Pg_i of the E-learning application is characterized by the input data-set ID_i and by the intermediary results or output data-set, OD_i. Linear Web-based distributed applications have components that are executed one after each other and the complete execution of the application is considered to be completed if all its components have been activated, as described in [3].

A Web-based distributed application that is developed with a linear structure is the application that performs a virtual quiz or exam based on numerous questions. For this application, the data is grouped in multiple sets, which define the problem dimension, the questions, the matrix for answers and questions, the points of the exam session.

The tree structure of Web-based distributed applications is best used in the case when at input data designing stage there are identified the parameters that help user to select various methods of data processing. The typology defines different ways of processing data and allows the selection of a way composed from modules that interact between them. The modules are placed on different calling levels. The components' calling is determined by user's options.

The graph structure is the most common because it reduces redundancy generated by code and components duplication. The modules structure of the Web-based distributed applications describes an oriented graph with repeating cycles.

By the number of users criteria, Web-based distributed applications are:

- local informatics applications with one user; the software product allows only one work session and permits access to only one user;
- network applications with more users simultaneously; the software application manages series of work sessions attached to a group of users; these use the same resources and have concurrent access to a common set

of application's functions; an example of this software is client – server applications.

The input data for Web-based distributed applications is composed by:

- user input; applications allow user to enter data like text, voice and video capture that will be processed in order to evaluate him; because the application is based on a Web-based application framework users could interact with various controls and select his options;
- automated input; applications allow users to load preformatted data resulted from previous working sessions or from other compatible applications.

A Web-based distributed application manages a multi-user environment and a multi-role functionality, [4] using all or some of the software components:

- user authentication that verifies secret and personal information as username and passwords that each specific user provides at login;
- user session management that authenticates a particular user request;
- user permissions that define the rules regarding user behavior;
- role level enforcement that assures that in this multi-ser multi-role environment each user can access only the functions allowed by its privileged role;
- data access;
- data processing represents the application itself.

2 Risks and vulnerabilities

The terms used in this paper are the ones that are generally accepted through definition in RFC2828 [5]. *Security* is viewed as a tool used to defend against unpredicted and unwanted actions that are made by a malicious user. An *attack* could exploit a *vulnerability*, which can be interpreted as a flaw or weakness in a system's design, or may regard the normal functions of a system for taking control over an *asset*, in this way damaging the normal working parameters. The *risk*, on the other hand, is defined as the combination of the likelihood that a threat can happen and its impact upon the system's assets. The impact on the security assets is measured in terms of evaluating the impact upon the security features of every affected asset like confidentiality, availability and integrity.

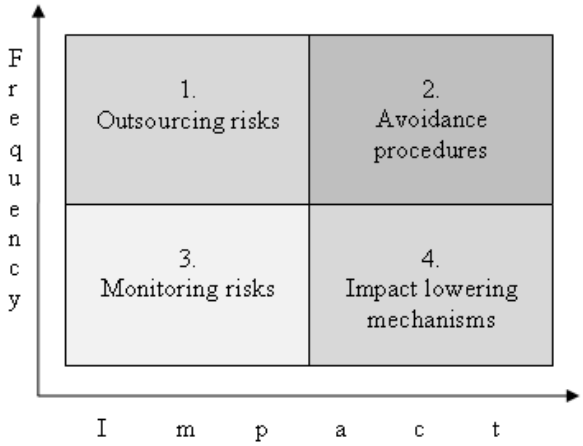

Fig. 2. Risk evaluation diagram

The risk assessment process is designed to enhance the audit overall process by means of aligning the internal audit objectives with the strategic goals of the organization.

Whether we are speaking of financial area or IT field, which are important for the business process of an organization, risk assessment has proven to be useful in the auditing process [6].

A distributed system may confront various types of vulnerabilities [7] and risk problems. Some of those could have low or high impact that could affect the system's functionality. If this scale is aggregated with the frequency scale, then it will result the following combination of factors and risk evaluation [8], presented in figure 2, which can explain how risks could be managed based on the evaluation of this two factors.

In this scenario presented in figure 2, depending on the intensity of the impact factor upon the distributed system and on the frequency of happening, four situations could be deducted:

1. high frequency and low impact presumes that risks could be given for handling to another party which is specialized on managing risks and who can implement risk preventing procedures more efficient and less expensive;
2. high frequency and high impact zone means that those risks categorized like that are to be avoided or if it's not possible, highly monitored and rigorous approached;
3. low frequency and low impact; is the category of risks that do not threat the distributed system security often and their impact is minimum; these risks should be monitored and inexpensive measures for dealing with them could be implemented;
4. low frequency and high impact; these risks suppose a well defined policy to be applied for them and measures for lowering the impact should be implemented even if their

frequency of producing stands low; it's better to prevent the risks from happening than to treat the consequences.

Table 1. Categories of Risks and Vulnerabilities for Web-based distributed applications

Security aspects	Possible actions to undertake
Password	Enforce password changes
	Log passwords reset operations
	Add structural constraints on creating password
	Prevent from sending passwords via email to users
	Have a consistent password recover policy
	Force users to change password on regular bases
Account	Implement a solid privilege policy
	Review account periodically
	Monitor unused accounts
Session	Lockout repeated failures
	Create logs about the session behavior
	Terminate inactive sessions
	Prevent roles interferences between different class of users
	Prevent elevation of privileges within the same role
Access	Delay repeated attempts
	Log unauthorized attempts
	Enforce access process with CAPTCHA (Completely Automated Public Turing Test To Tell Computers and Humans Apart) usage
	Trace account access from different IP addresses
Data	Protect information from unwanted changes – integrity
	Providing means of knowing the true origin of data – authenticity
	Keep sensitive data unreachable for unwanted viewers – confidentiality
	Unequivocal association between originators and data – non-repudiation
	Implementing mechanisms for data availability
Communication	Enforce secure communication via VPN or encrypt sensitive information
	Perform hardware and software communication regular maintenances and updates
	Use effective firewalls and protective systems for monitoring data traffic

Table 1 presents a list of components related to the security aspects along with the actions that could be undertaken for increasing the security level of a distributed system.

Presented list is meant to help users evaluate the security level of a web-based application and eventually reinforce their security implemented measures, based on the existence or not of such characteristics.

This evaluation process can be done using vulnerabilities notification frameworks [9] that can create queues of messages depending on the severity level and delivering them to administrator for further attention.

3 Security metrics for web based distributed applications

The software metric is a mathematical model developed based on an equation that has the form $y = f(x)$, where x is a variable or set of variables, that are associated with the model influence factors, and y is the result [10].

A mathematical model contains one or more equations, inequations and has one or more objective functions. Its role is to describe, to measure, the state of associate system. The role of software metric is to measure a certain characteristic of a software application including all factors that influence the level of measured characteristic. Being applied to all software application from a homogenous set, the metrics become the instrument that helps making classifications and hierarchies of analyzed software applications.

Based on the mathematical model, the process of defining a metric consist of two different stages:

- define the objective of the metric, the y variable; the objective must clearly describe what to measure and this requires a measurable element;
- identify and define the influence factors, x variables, that are independent and by their behavior or actions determine the metric objective value.

The metric complexity depends on its model and this may affect its quality. An increased complexity requires difficult utilization

procedures and result interpretations. In practice, a simple and well defined metric is more appropriate because it is simple to use and its results are less prone to be misinterpreted.

According to [11], security metrics represent measurable standards that are used to monitor the effectiveness of goals and objectives established for IT security.

To facilitate understanding and easiness integration into an organization for all the security metrics proposed it is good that an audit-based approach should be implemented in order to verify the compliance with the internal and external standards [12].

Metrics are different from measurements because, accordingly to [14], measurements provide single-point-in-time views of specific, discrete factors, while metrics are derived by comparing to a predetermined baseline two or more measurements taken over time.

Good metrics are those that are SMART, meaning specific, measurable, attainable, repeatable, and time-dependent, according to [13]. In [11] there is added another attribute, comparable, that makes metrics useful tools in comparing different security measures or different values in the time evolution of a security measure. These characteristics define a metric as a valuable and useful tool in security assessment process. Metrics should be defined based on the risks and vulnerabilities analysis. The role of the metric is to analyze the measured level of the vulnerability or risk and to indicate the degree to which:

- the security policy is implemented
- known vulnerabilities have been solved
- the application has been tested for unknown vulnerabilities

In [14] there are described seven key steps to be used in the process of defining a security metrics program:

- define the metrics goals and objectives
- decide which metrics to generate
- develop strategies for generating the metrics
- establish benchmarks and targets
- determine how the metrics will be reported
- create an action plan and act on it, and
- establish a formal program review/refinement cycle

In [15] the authors examine different frameworks for developing security requirement and metrics of information systems security. The International Systems Security Engineering Association (ISSEA) has defined the System Security Engineering – Capability Maturity Model (SSE-CMM) [16] with the goal to define, improve and assess security engineering capability. This model defines characteristics for a 22 steps security engineering process that is explicitly defined, managed, measured and controlled. The model represents a guide for organizations that need to define a security assessment process used to measure security aspects regarding operations, information, networks, personnel, communications and computers. Table 2 describes the SSE-CMM 22 process areas that require metrics definition.

Table 2. Security process and security metrics areas defined by SSE-CMM [13]

Process areas
PA01 Administer Security Control
PA02 Assess Impact
PA03 Assess Security Risk
PA04 Assess Threat
PA05 Assess Vulnerability
PA06 Build Assurance Argument
PA07 Coordinate Security
PA08 Monitor Security Posture
PA09 Provide Security Input
PA10 Specify Security Needs
PA11 Verify and Validate Security
PA12 Ensure quality
PA13 Manage configurations
PA14 Manage Project Risks
PA15 Monitor and Control Technical Efforts
PA16 Plan Technical Efforts
PA17 Define Organization Systems Eng. Process
PA18 Improve Organization Systems Eng. Process
PA19 Manage product line evaluation
PA20 Manage Systems Eng. Support Environment
PA21 Provide ongoing skills and knowledge
PA22 Coordinate with suppliers

Another widely security metrics standard is the NIST 800-55, [17]. This standard defines a metrics development process that consists of two major activities:

- define the security goals of the IT security program;
- define and select metrics to measure implementation, efficiency, effectiveness and the impact of the security controls.

The NIST 800-55, [17] standard defines the quality of a security metrics accordingly to a metric detail form that describes:

- performance goal represents the objectives that are measured by the metric;
- performance objective describes the actions

required to reach the performance goals;
- the metric that represents the quantitative measurement;
- purpose describes the metric functionality;
- implementation evidence;
- frequency of using the metric;
- the formula describes the mathematical model to be applied in order to obtained a numerical value;
- data source
- indicators represent ways of interpreting the metric value.

The quality of security metrics is analyzed, also in [18], from the viewpoint of seven *myths*:
- metrics must be objective and tangible;
- metrics must have discrete values;
- metrics does not require to use absolute measurements;
- metrics should not to be costly;
- metrics are useful because "you can't manage what you can't measure and you can't improve what you can't manage", [18];
- it is important to define metrics to measure the process of information security and not just their results;
- complex security metrics need to be further analyzed by defining metrics for their input factors.

The NIST 800-55 standard, [17], defines ten metrics used to measure 17 Information Technology (IT) security topics described in 800-26, *Security Self-Assessment Guide for Information Technology Systems*:
- Percentage of systems that had formal risk assessments performed and documented;
- Percentage of total systems for which security controls have been tested and evaluated in the past year;
- Percentage of total systems that have the costs of their security controls integrated into the life cycle of the system;
- Percentage of total systems that have been authorized for processing following certification and accreditation;
- Percentage of current security plans;
- Percentage of systems that have a contingency plan;
- Percentage of systems for which contingency plans have been tested in the past year;
- Percentage of employees with significant security responsibilities who have received specialized training;
- Percentage of agency components with incident handling and response capability;
- Number of incidents reported externally to law enforcement.

Another framework for vulnerabilities metrics is the Common Vulnerability Scoring System (CVSS), [19], that provides a hardware and software independent measurement system. The CVVS system goals are to define a standardized vulnerability scores that allows prioritizing risks. The metrics are grouped in three categories, described in figure 3.

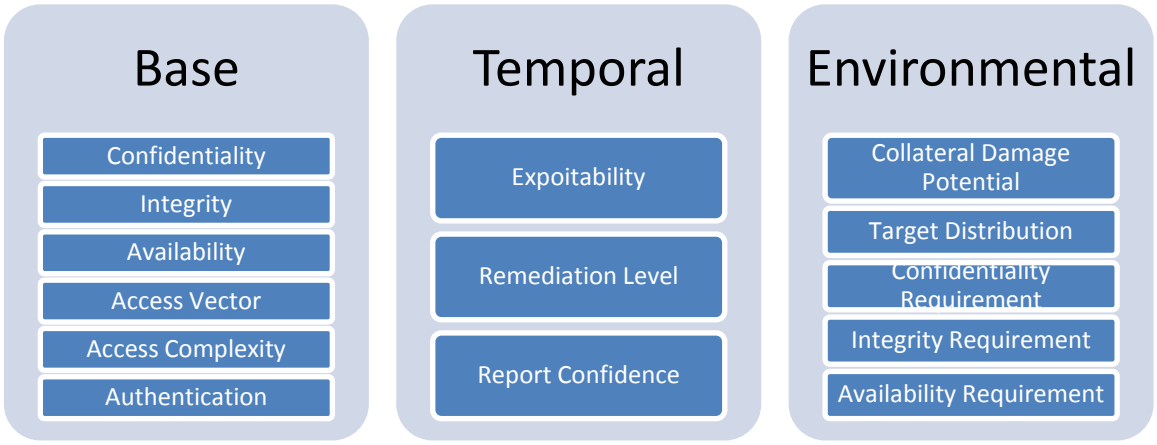

Fig. 3. CVSS Metric Groups [16]

The *Base* category contains metrics that measure fundamental qualities of a vulnerability that are constant over time and environments. The *Temporal* metrics group contains vulnerability characteristics that evolve over the lifetime of vulnerability but not among user environments. The *Environmental* category contains vulnerability characteristics that are dependent on a specific user's environment.

For the risks and vulnerabilities describes in the second section, there are some possible metrics that can be used to measure them, presented in table 3.

Table 3 Identified Security Metrics

Area	**Metrics**
Users	▪ Password strength; ▪ Number of accounts with passwords that do not expire ▪ Number of accounts with manufacturers default passwords still being used ▪ Percentage of Tier 1, 2 & 3 logon environments that do meet password complexity requirements ▪ Number of distinct user accounts
Cookies	▪ Expire time ▪ Not in-clear sensitive data
Services (Email, FTP and others)	▪ Total number of malware stopped at the e-mail gateway ▪ Total number of messages dropped as spam ▪ Percentage of total e-mail secured
Communication	▪ Use post requests ▪ Use encryption ▪ The strength of the public-key system ▪ Number of potentially dangerous open ports on workstations; ▪ Vulnerability port Scan of inside demilitarized zone (DMZ); ▪ Vulnerability port Scan of outside demilitarized zone (DMZ); ▪ High Risk Network Traffic ▪ Number of shared resources ▪ Number of unsecured communication nodes
Input validation	▪ Use validation controls ▪ Use client side validation ▪ Use server side validation
System devices	▪ Total number of devices with antivirus software installed and current ▪ Percentage of all devices with all appropriate patches installed

4 Security assessment process

ICT Security is a well documented field with hundreds of standards that encompasses many important aspects starting with identifying resources, documenting security algorithms, evaluating performances and giving new visions for keeping up with the immense progress of ICT area.

In [20] a description is presented on how international institutions come together for dealing with all the aspects which are enclosed in the security area. ISO is a well known network that is focused on elaborating standards. ISO involves the cooperation of almost 150 countries that are working in collaboration with international organizations, the business industry and government sectors. ISO technical work is divided in a hierarchy very well organized in three major categories:

- technical committees, TCs;
- subcommittees, SCs;
- working groups, WGs.

For ICT standardization ISO formed a joint with the IEC forming the ISO/IEC JTC 1 Information Technology which is dived in the following subcommittees that are dealing with security standards:

- SC 17 Cards and Personal Identification;
- SC 27 IT Security Techniques;
- SC 37 Biometrics.

From these subcommittees, SC 27 had become the primary resource for international standards relating ICT security area. The technical work of SC 27 is divided in several workgroups [21]:

- Security Management;
- Security Algorithms;
- Security Assessment.
- Security controls and services;
- Identity management and Data protection;

This aspect is also analyzed by the System Security Engineering – Capability Maturity Model (SSE-CMM) [16], which defines what security assessment means [16]:

- impact assessment;
- security risk assessment;
- threat assessment;
- vulnerability assessment;
- security verification and validation.

In figure 4 is presented a diagram of this hierarchy of communities that are working together for standardizing all the aspects that appear and persists in a well defined environment adding value to the ICT security community.

The first workgroup WG1 deals with regulations and guidelines for Information Security Management Systems.

The second workgroup WG2 handles the matters of standardizing IT security techniques and

mechanisms within JTC1.
The third workgroup WG3 provides useful information about the regulations and procedures of the audit process, evaluation criteria.

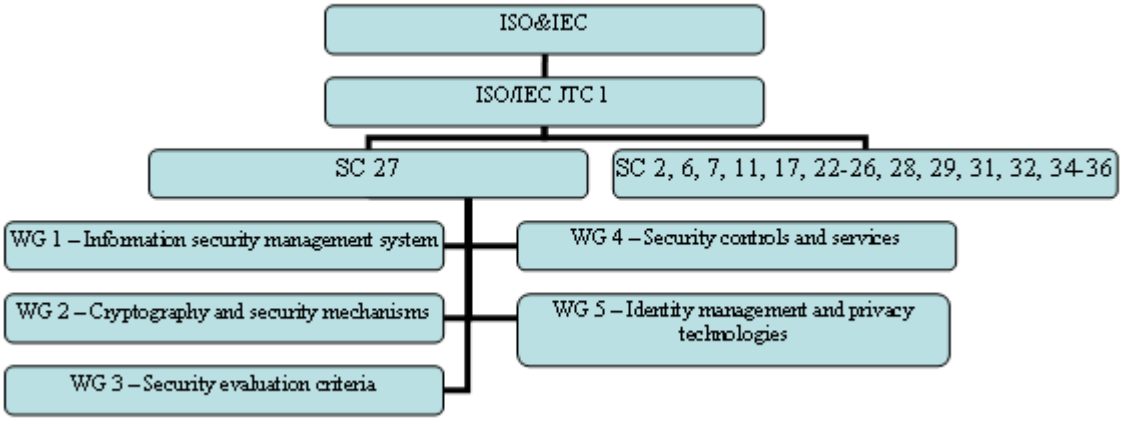

Fig. 4. Hierarchy diagram of standardization committees.

The fourth workgroup WG4 covers the development and maintenance of standards and guidelines concerning security services and controls.
The fifth workgroup WG5 is dealing with identity management, biometrics and protection of data.
The audit is a term which stands for listening the facts and figures provided from the activity of an organization. Initially, the term was used for the accounting and financial domain, but gradually spread in every area who needed a safe and impartial point of view on the audited activities.
Auditors work with the full knowledge of the organization in order to understand the resources that they must check and the complete set of external connections with these assets.
The audit process can be classified by its purpose in two different categories:

- internal audit – is conducted by specialists part of the organization having the role of making an internal radiography of the target area with the purpose of knowing the real state of the organization aspects implied;
- external audit – used by companies for getting external and reliable feedback about the current state of the audited domains and gaining trust in front of their possible clients; is made by third parties, independent from the audited organization which certifies through special means the quality of the audited processes.

The audit process for ICT field is conducted on several levels in which hierarchy, security stands on top. A computer security audit is a systematic, measurable technical assessment of how the organization's security policy is used in an information system. Its main objective is analyzing ICT area of the organization, to test the security levels for the respective information systems. In auditing the ICT field, auditors must approach the following directions:

- the organization ICT resources;
- the processes enrolled in the ICT activity;
- the information security aspects;
- the computer security aspects.

The audit process follows a certain pattern to stand up to the existing standards and regulations. In order to conduct an audit process first of all, the auditors must find out which are the resources that they must evaluate and gather information about all the aspects involved along the activity line. After a well and rigorous documentation in the field examined the next stage is based on making qualitative and quantitative measurements and confront them with the figures provided by the company itself. At the end of the evaluation process they must provide a full documented report about what are the deficiencies of the analyzed field and what are the improvements that can be made to adjust the identified vulnerabilities and risks.
The head management must implement the audit report suggestions and make a re-audit process to analyze its efficiency.
In auditing a web based distributed system the following aspects should be analyzed:

- the communication process;
- the overall information process;
- the security constraints;
- the quality of the results provided by the system.

For undertaking an audit process for the information system of an organization, the organization itself must minimize the possible interferences between the audit and the business process. ISO's standards describe a list of measures that could prevent the interference between this to processes as follows:

- audit requirements supervised by appropriate management;
- the actions taken should be only for check purposes;
- the checks should be limited to read-only access of data;
- back-up copies of vital data should be made before any audit operation;
- all audit operations must be logged for any possible failures;
- the whole process of auditing must be documented in order to provide detailed information about all the aspects involved.

In order for obtaining the best possible results, auditors must be qualified for their activity and also must not be implied in the activities that they are caring out.

As the systems are growing in complexity and many are the relations that come to substantially trigger chain of decisions from the organization management, the common ways of auditing such systems are beginning to be inefficient. In this direction, automated auditing processes had been developed for diminishing the need of human supervision. Although these kinds of systems could not carry the audit process all the way and manually auditing procedures must be enrolled, they somehow tend to add a plus value to the overall process, doing repeatedly automated procedures that facilitates and improve the audit end results.

Many studies revealed that the need of audit automation is increasingly growing based on the exponential growth of the complexity level of such systems. In [22] are presented ways of achieving audit automation by the use of security design patterns. But entering this field is very tricky, because of the imperious necessity of having well defined standards accepted across multiple different systems. Table 1 could be mapped on such security patterns. ISO 27002 and its related documents have contributed to the standardization of security features.

Patterns, as part of information systems, add extra knowledge and help the process of communication as common language helps people understand each other. Security patterns could be created based on such entities presented in table 1. Patterns communicate with other patterns and are structured based on different criteria such as:

- patterns for security integrity and confidentiality called protected system patterns;
- patterns for security availability – available systems patterns.

Patterns are developed based on a hierarchy of patterns, clustered in patterns catalogues, patterns systems and patterns languages.

These security entities identified are meant to be part of a recurring evaluation process described as the audit automation in which the following checking could be pursued:

- checking the existence of operation logs for vital processes in the system;
- checking for the existence of software and hardware protective equipments such as authentication systems, firewalls, antivirus programs, antispyware, anti-phishing, rooters with encryption capabilities, et cetera;
- verifying protection of sensitive information by means of encryption;
- identifying the existence of mechanisms that allow redundancy in case of unavailability, back-up systems & policies;
- testing for techniques and algorithms used in identifying losses of data integrity such as hash functions;
- searching for vulnerabilities that could have passed undetected by the IT security department.

All these tests that must be referenced by an audit process could be automatically implemented and launched to bring efficiency and reliability to the overall process of auditing web based distributed systems.

The security assessment could be compared with a test process that analyzes the application behavior when it is tested against known vulnerabilities. From this point of view, the security assessment process is divided in two type of analysis white box and black box.

Black box security assessment process is based on the fact that the auditor does not have knowledge about the source code, documentation or server logs. In this case, the assessment process is conducted based on general attacking methods that require:

- Analyze application interface
- Gather information about the application
- Knowledge about known vulnerabilities
- General methods of attacking security

The black box assessment process is done by

experienced auditors or security testers that simulate real attacks. Depending on the results, the audit process can validate the application security level or may reveal vulnerabilities that were not considered.

White box security assessment process is conducted based on application documentation and specifications. Its goal is to test and validate that the implementation of proposed and described security measures has been done accordingly to the specified requirements. The assessment tests and procedures are designed accordingly to the security specifications. Their objective is to validate the implementation of security rules and procedures.

5 Conclusions

As the systems tend to increase in complexity and organizations develop more their structural degree, reliability becomes a primary concern for all parties involved in the process. More aspects coming and being added to the systems, usually means more correlations to make and more things to manage so that the systems wouldn't go unstable. For this reason audit comes to help organizations certify that their products are reliable and of good use. An auditor stands as factum on behalf the organization, attesting that the information system used for the business process is trustful and no drawbacks influence the quality implied by its products. As parts of the audit process tend to become automated, the question that arises is: "Is the audit automation a process that needs to be audited as well?" Those facts must seriously be a primary concern for both the audit community and IT developers because of the implications that may come along with an unreliable product.

Acknowledgements

This article is a result of the project „Doctoral Program and PhD Students in the education research and innovation triangle". This project is co funded by European Social Fund through The Sectorial Operational Programme for Human Resources Development 2007-2013, coordinated by The Bucharest Academy of Economic Studies (project no. 7832, "Doctoral Program and PhD Students in the education research and innovation triangle, DOCECI").

References

[1] S. Ravi and S. Pierangela, "Access Control: Principles and Practice," *IEEE Communications Magazine*, September, 1994

[2] I. Ivan and A. Felician, *Structuri HTML (HTML structures)*, ASE Publishing House, Bucharest, 2005.

[3] I. Ivan, C. Boja and A. Felician, "Web Applications Optimization," *Proceedings of the InfoBUSINESS'2005, International Symposium*, October 14-15, 2005, „Al. I. Cuza" University of Iaşi, Iasi, Volume edited by Ioan I. ANDONE, PIM Publishing House, Iasi, Romania, 2005, pp. 21 - 30.

[4] M. Cross, S. Kapinos, H. Meer et al, "Web applications vulnerabilities, Detect, Exploit, Prevent," *Syngress*, 2007.

[5] R. Shirey, Internet Security Glossary, 2000, Available at: http://www.ietf.org /rfc/rfc2828.txt

[6] A. Marco and D.O. Giuseppe, "Internal Auditing and Risks Assessment in Large Italian Companies: an Empirical Survey," *International Journal of Auditing*, No.7, 2003, pp. 191-208.

[7] T. Bakhshi, M. Papadaki and S. Furnell, "Social engineering: assessing vulnerabilities in practice," *Information Management & Computer Security*, Vol. 17, No. 1, 2009, pp. 53-63.

[8] A. Jones, "A framework for the management of information security risks," *BT Technology Journal*, Vol. 25, No. 1, January 2007, pp. 30-36

[9] A. Al-Ayed, S. M. Furnell, D. Zhao and P. S. Dowland, "An automated framework for managing security vulnerabilities," *Information Management & Computer Security*, Vol. 13, No. 2, 2005, pp. 156-166.

[10] I. Ivan, M. Popescu, P. Sinioros and F. Simion, *Metrici software (Software metrics)*, Inforec Publishing House, Bucharest, 1997.

[11] V. Patriciu, I. Priescu and S. Nicolaescu, "Security Metrics for Enterprise Information Systems," *Journal of Applied Quantitative Methods*, Vol 1, No. 2, pp. 151 – 159, 2006, accessed January 2009, Available at: http://www.jaqm.ro

[12] L. Jennifer, "Information Security Metrics: An Audited-based Approach", *NIST and CSSPAB Workshop*, Washington, D.C., June 2000. URL: http://csrc.nist.gov/csspab/june13-15/Bayuk.pdf (10 July 2001)

[13] G. Jelen, "SSE-CMM Security Metrics," *NIST and CSSPAB Workshop*, Washington, D.C., 13-14 June 2000, accessed 10 July

2001, Available at: http://csrc.nist.gov/csspab/june13-15/jelen.pdf.
[14] S. C. Payne, "A Guide to Security Metrics," *SANS Security Essentials GSEC Practical Assignment Version 1.2e, SANS Institute, InfoSec Reading Room*, 2006, accessed January 2010, Available at: http://www.sans.org/reading_room/whitepapers/auditing/a_guide_to_security_metrics_55?show=55.php&cat=auditing
[15] J. A. Chaula, L. Yngström and S. Kowalski, *Security Metrics and Evaluation of Information Systems Security*, accessed January 2010, Available at: http://citeseerx.ist.psu.edu
[16] The Systems Security Engineering Capability Maturity Model (SSE-CMM), *Model Description Document, Version 3*, 2003, accessed December 2009, Available at: http://www.sse-cmm.org/model/model.asp
[17] M. Swanson, N. Bartol, J. Sabato, J. Hash and L. Graffo, NIST Special Publication (SP) 800-55, *Security Metrics Guide for Information Technology Systems*, accessed January 2010, Available at: http://csrc.nist.gov/publications/PubsSPs.html
[18] G. Hinson, "Seven myths about information security metrics," *originally published in ISSA Journal***,** July 2006, Accesed Feb. 2010, Available at: http://www.noticebored.com/html/metrics.html
[19] P. Mell, K. Scarfone and S. Romanosky, "A Complete Guide to the Common Vulnerability Scoring System Version 2.0," *Forum of Incident Response and Security Teams*, accessed January 2009, Available at: http://www.first.org/cvss/cvss-guide.html
[20] *Joint Technical Committee of the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC)*, Available at: http://en.wikipedia.org/wiki/ISO/IEC_JTC1
[21] *BSI Group eCommittees system*, http://ecommittees.bsi-global.com
[22] T. Tryfonas and B. Kearney, "Standardising business application security assessment with pattern-driven audit automations," *Computer Standards & Interfaces*, No. 30, 2008, pp. 262-270.

Cătălin BOJA is Lecturer at the Economic Informatics Department at the Academy of Economic Studies in Bucharest, Romania. In June 2004 he has graduated the Faculty of Cybernetics, Statistics and Economic Informatics at the Academy of Economic Studies in Bucharest. In March 2006 he has graduated the Informatics Project Management Master program organized by the Academy of Economic Studies of Bucharest. He is a team member in various undergoing university research projects where he applied most of his project management knowledge. Also he has received a type D IPMA certification in project management from Romanian Project Management Association which is partner of the IPMA organization. He is the author of more than 40 journal articles and scientific presentations at conferences. His work focuses on the analysis of data structures, assembler and high level programming languages. He is currently holding a PhD degree on software optimization and on improvement of software applications performance.

Mihai DOINEA received a PhD scholarship from the Academy of Economic Studies, Bucharest, Romania in Economic Informatics at the UvA Research Center. He has a master diploma in Informatics Security (2006). He is also a lecturer assistant and he teaches data structures and advanced programming languages at the Academy of Economic Studies. He published more than 20 articles in collaboration or as single author and co-published two books in his area of interest. His research interests are given as follows: informatics security, distributed applications, optimization criteria, databases, artificial intelligence, information management, security policies, mobile devices, networking and wireless communication.