J. Appl. Environ. Biol. Sci., 4(8)109-114, 2014

ISSN: 2090-4274
Journal of Applied Environmental and Biological Sciences
www.textroad.com

Provide an Approach for Evaluating the Security of Software Products

Roya Vaezi[1], Nasser Modiri[2]

[1,2] Department of Computer, Zanjan Branch, Islamic Azad University, Zanjan, Iran

Received: March 19, 2014
Accepted: July 8, 2014

ABSTRACT

In this paper, an approach is proposed which tries to consider security at the beginning of software development. In proposed approach, during life cycle of web-based applications, security measures have been taken to improve security in these applications. By evaluating security in each phase of software development process, we ensure that vulnerabilities properly identified and reduced at each phase; then the next phase is started.
In this approach, security verification OWASP ASVS standard, which is a measure in four levels, is the base of activities classification. After that, we have presented all requirements, activities, security controls and countermeasures from the beginning of software development lifecycle systematically and organized.
KEYWORDS: Security Evaluation, Web based Applications, Vulnerability, Security controls, Countermeasure

I. INTRODUCTION

Increasingly development of software as a main element in our daily lives and enormous costs of software failure has caused the emergence of activities and programs to evaluate software [Modiri and et al 2010]. The software programs are written for years and the security, more or less, is mentioned in them. Even, many customers may not have enough knowledge about its quality and reliability. With expanding web and web-based applications, as well as being more intelligently the attacks and frauds, software security has grown increasingly important. Security is a need that should be considered in any type of software. It seems that those approaches that denoted the security injection in software to post-production step would not have been effective. One of the best ways to avoid security problems is to consider security in each phases of software development lifecycle. Integrating security in lifecycle can improve security of web-based applications [OWASP, 2008]. The consensus is that whatever software vulnerabilities are discovered and fixed in early stages of production, the final cost of production will be less. According to a study published by B. Boehm, and V. Basili [Boehm and Basili, 2001] troubleshooting a software error after installing is a hundred times when the error was discovered and corrected in early stages of software development. This figure will be higher in the case of security flaws, because in addition to above spending, compensation costs resulted from abuse of these defects in order to steal information, sabotage and other attacks would also burdened the producer with high responsibility. Therefore, the aim of this paper is presenting an approach which knows the software security arrangements as appropriate throughout software development life cycle.

In following and in second part of the paper, concept and necessity of security in software are expressed and how it is provided. In third section, the term vulnerability is defined and some of possible vulnerabilities are discussed in applications. In fourth section, security test, its purpose and necessity as well as some security testing methods would be presented. ASVS standard is explained in the fifth section. In sixth section, the ISO / IEC12207 standard is described. In seventh part, approach of evaluating proposed security with procedures, details and activities within each phase is represented. Finally, the conclusion is given.

II. CONCEPT OF SECURITY IN SOFTWARE

A. Definition of Security

Software security is a young system which is addressed security features of software when it is developing, testing, implementing, and using. This includes many security activities at different stages of software development, such as threat modeling, risk management, and security tests. The first security mission is to protect an organization's assets, which may include tangible items (such as a web page or a customer database) or intangible items (such as reputation of an organization). Security is a path not a destination; in parallel with analysis of existing infrastructure and applications, you need to identify the threats and dangers of them. In fact, security refers to risks management and implementing a system in order to accountability and deal with threats.

B. Security Requirement in Software

With increasingly growth of web-based applications and their interaction with different people, maintaining security is one of requirements. However, we are still witnessed several attacks to web-based applications, with the aim of financial abuse, or stealing sensitive information [Lucca and Fasolino, 2006].

Web-based software can be accessed by a network for different users with different intentions at all around the world. It is very difficult or even impossible to limit the population of users who have access to system. For this reason,

***Corresponding Author:** Roya Vaezi, Department of Computer, Zanjan Branch, Islamic Azad University, Zanjan, Iran.
E-mail: Vaezi.r@gmail.com, Tel: (+98)243-4221003.

in order to protect sensitive data and to make secure of transferring data, more powerful security structures should be used than other software.

Web-based programs are the most common targets of online hackers, because they may be the most visible entry points into your company, and they are obviously fraught with vulnerability. On the other hand, most companies are forced to be present in cyberspace to bring their business to result, hence risk confronting is undeniable.

Being active in the field of web application security must become a high priority. When a web-based application is at risk it leads to lose the capital. In these conditions, at least for reputed businesses, to lose money is not the most inflicted damage, but it is more displeasing when their reputability is damaged.

Long cuts of web-based applications are resulted in anger of customers and company‘s manager. Also, regardless of whether the attack was preventable or not, the IT department is blamed.

C. How Provide the Security

One of the ways for ensuring software security is applying security test and evaluation over software lifecycle. For this reason, we, in this paper, present an approach to assess the security of software products.

III. IDENTIFICATION ON EXISTING VULNERABILITIES IN SOFTWARE

A. Definition of Vulnerability

Vulnerability is a defect or weakness in design, implementation and/or performance of a system that can cause the violation of security policies of the system. The term vulnerability is very wide in web. ISO defines vulnerability as follows:

It is a weakness in a capital or a group of assets that can be exploited by one or more threats [ISO / IEC, 2008].

An asset can be everything that is valuable for organization, including business operations and information resources that support the goals of organization [British Standard Institute, 2004].

ENISA defines vulnerability as: “weakness, design or implementation error that can lead to unexpected and undesirable event causing the security of computer systems, networks, software, or involved protocols being at risk”.

In fact, vulnerabilities are weaknesses in a system that provides early potential of emerging a harmful event. Weaknesses in design, misconfiguration, and use of unsafe coding methods are the most important reasons for creating a weakness in the system. Weakness in verification of input data by user is an example of vulnerable layer in a program that can provide the occurrence of attack.

B. Browsing Some of Possible Vulnerabilities in Software

In identified vulnerabilities in web-based applications the highest number is related to malicious code injection (26%). In following, lack of information and errors for each session management (16%), authentication and access authorization (13%), CSRF (8%), SQL injection (6%), web server version (5%), remote code running (5%), web server configuration (3%), and access to unauthorized directory (2%) are all vulnerabilities [Cenzic, 2013].

IV. WHAT IS SECURITY TEST?

A. Definition

Security test measures the features of system that are related to availability, integrity and confidentiality of data and system services. Users / customers should be encouraged by this subject that their security demands are clearly characterized when the requirements are determined. So, security issues are considered by designers and testers.

The goal of security test is evaluating the performance of web system‘s defense mechanisms against unwanted access of unlicensed users, maintaining system resources against inappropriate users, and also providing access to users who have authorization.

B. The Purpose and Need for Security Test

Security test is performing to ensure that the software under test is strong enough, even software performance remains in an acceptable condition when an attack is occurring.

Objectives of security test can be the following:

- To ensure that sufficient accuracy is applied to identify security risks.
- To ensure that there is a real mechanism for defining access to system.
- To ensure that there is sufficient expertise to test security.
- To conduct reasonable tests to verify proper operation of implemented security measures.

Some of the reasons that suggest the importance of security test are mentioned below:

- Downtime: Most of security shortcomings lead to service failure and loss of income.
- Legal issues: security issues could lead to legal problems and complexities.
- brand damages: confidentiality of company‘s data can be severely damaged by frequent downtimes or shortage of credit and may result in loss or stall of business.
- Cost: the expense of each extracted security issue is several times higher than the cost of identifying and repairing them during development and testing phases of software.

C. Major Methods of Security Testing

With growing concern about software security, research on security testing has made some progress. In the following we will refer Examples of security testing methods derived from the [Tian-yang and etc, 2010].

1) *Formal Security Testing:* The basic idea of formal method is to build a mathematical model of the software, and provides software form specification supported by some formal specification language. Formal security testing methods can be classified into theorem proving and model checking.

2) *Model-Based Security Testing:* Model-based testing constructs a model by the behavior and structure of software, then derives test cases from test model. Finally drive software to run the test cases. The behavior of Software system can be described by input and output sequence, activity diagram, sequence diagram, collaboration diagram, condition or data stream. Software behavior model and structure model is the exact description of the tested software, which can be used to generate test cases. Software testing models are commonly used , such as finite state machine, UML model, Markov chain.

3) *Fault Injection-Based Security Testing:* Wenliang Du used fault injection technique for software security testing, which established fault mode of Environment-Application Interaction, EAI. Fault injection focuses on the interaction points of application and environment, including user input, file system, network interface, environment variable.

4) *Fuzzy Testing:* Fuzzy Testing is effective to discover security vulnerability, which gets more and more attention. Fuzzy testing would inject random data into program to test whether it can running normally under the clutter input. Fuzzy testing is illogical, just creates clutter data. Fuzzy testing would find flaws of tested software, which are difficult for the other logical testing method.

5) *Vulnerability Scanning Testing:* Vulnerability scanning, as an important method to find software security risks, includes testing space scanning and known defects scanning. Testing space scanning deals with network port, string, procedure data, network data and other elements scanning, for example, through network port scanning, it can be found whether the port of software is opened which should not open. Known defects scanning finds known flaws usually basing on the defect library.

6) *Property-Based Testing:* The method transforms security property of software into specification described by TASPEC language. It would extract the code in relation to specific property by program slicing technology, and discover violation of the code against security property specification. Property-based testing focuses on some specific security properties, which can meet requirement of classification and priority.

7) *White Box-Based Security Testing:* Static analysis, as one of common white-box based testing methods, is good at finding security bug, such as buffer overflow. The main technologies of static analysis are deducing, data flow analysis and constraint analysis.

8) *Risk-Based Security Testing:* risk-based security testing combined the risk analysis, security testing with software development lifecycle, as early as possible to find high-risk security vulnerabilities.

V. OWASP ASVS STANDARD

Application Security Verification Standard (ASVS) has been provided by OWASP in 2009. The ASVS defines four levels of verification that increase in both breadth and depth as one moves up the levels. For each of the verification levels, specified verification requirements related to those and the methods of do any of Verification.

The four levels are presented as follows:

- Level 1 - Automated Verification
- Level 2 - Manual Verification
- Level 3 - Design Verification
- Level 4 - Internal Verification

VI. ISO / IEC12207 STANDARD

The ISO/IEC12207 Standard is a framework associated with software engineering that meets the needs of software lifecycle from the beginning to end. In fact, ISO/IEC12207 standard is involved set of processes which, in turn, the processes contain a set of activities; then the activities are included the sequence of tasks.

This standard is employed for defining process, controlling, and improving software lifecycle processes [Li andetc, 2009]. ISO/IEC12207 Standard has been created for software lifecycle processes (SLPs). The purpose of this standard is flexibility, modular and compatible. Software lifecycle in this standard is divided into two main processes:

- Software specific processes (SSPs)
- System context processes (SCPs) [Portillo-Rodriguez and etc, 2010]

Although it is special view to software life cycle, with a wider look, it considers software as a part of system. Hence, the standard includes two basic parts: the processes involved in software and processes involved in the system.

The ISO/IEC12207 is composed of two series of processes, 7 main process groups and 43 processes.

VII. PROPOSED APPROACH TO IMPLEMENT THE EVALUATION OF SECURITY

A. Steps of Proposed Approach

The proposed approach represents a method of evaluating the security of software by using the processes in ISO / IEC12207 standard and security measures based on ASVS standard. In each part of approach, different stages are explained to integrate evaluation requirements of ASVS standard. The proposed approach consists of the following steps:

1. Doing software development process based on ISO / IEC 12207.
2. Identifying main security measures in each phase based on the requirements of ASVS.

3. evaluating security by reviewing requirements of ASVS standard that can be implemented in each phase.
4. Preparing safety reports on software development process in each phase.

As mentioned above, the approach proposed in this paper tries to consider the security at the beginning of software production. In this approach, security measures have been taken to improve security in these applications throughout production life cycle of web-based applications. By evaluating security in each phase of software development process, we ensure that vulnerabilities properly identified and reduced at each phase; then the next phase started.

B. Presenting the Proposed Approach in the Form of Picture

Fig. 1 shows the schema of proposed idea. Sections and sub-sections of approach are shown in this figure. Moreover, the processes that are employed from ISO / IEC12207 standard and phases of proposed approach are presented. Then, you can see security measures, security evaluation, and providing report that are based on ASVS standard and should be performed in all phases of proposed approach.

Steps of proposed approach along with details and activities within each phase and consistently with fig. 1 are presented in table I.

It is worth noting that phases can be implemented in proposed approach in orderly way; so in each phase it is possible to go back. Security measures in each phase will be conducted in parallel with processes of 12207. At the end of each phase, the documents of its activities are examined; if approved, security report created according to ASVS and next phase of software development is started.

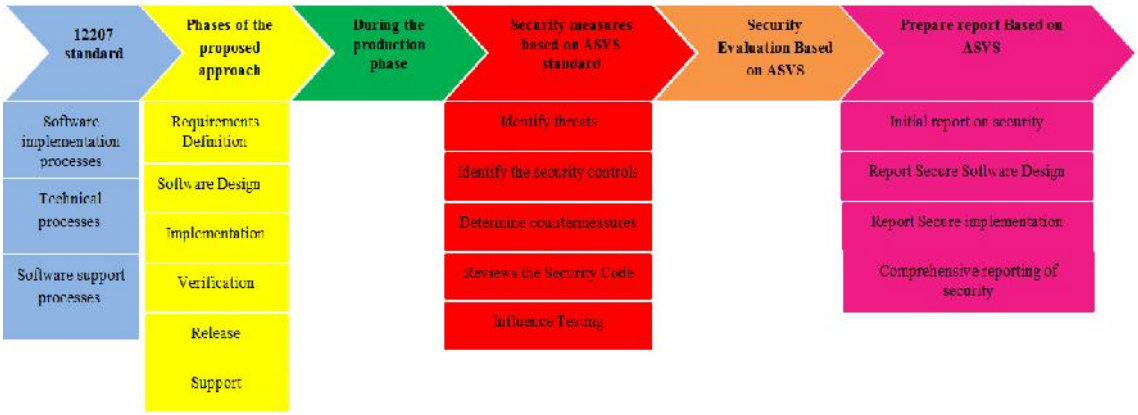

Fig. 1. Abstract schema of proposed approach

C. The Output of Proposed Approach Phases

1) *Requirements Definition Phase Output:* The outputs of Requirements Definition Phase activities be expressed as follows:

- Provide a comprehensive explanation of what the user of software systems are expected to perform.
- Explain what the user wants to do with the software system.
- Define all constraints that the user wishes to impose any solution on it.
- Determining the functional and non-functional requirements and prioritize them all.
- Planning for software project management for the next phases.
- Software configuration management planning for the next phases.
- Validation and verification plan for the next phases.
- Software quality assurance plan for the next phases.
- User Acceptance Test Plan.
- Documentation of the software and user requirements.

TABLE I

PHASES OF PROPOSED APPROACH WITH DETAILS OF THE WORK BREAKDOWN

Phases of the proposed approach	**Processes used from ISO / IEC 12207 standard**			**Security measures based on ASVS standard**	**Security Evaluation Based on ASVS**	**Prepare a report Based on ASVS**
	Software implementation processes	**Technical processes**	**Software support processes**			
Requirements Definition	Software requirements analysis	Stakeholder requirements definition, system requirements analysis	Software documentation management	•Identify security requirements •Identify security objectives •Identify threats •Identify the security controls	Security Evaluation Based on ASVS	Initial report on security
Software Design	Software architectural design , Software detailed	System architectural design	Software documentation management	•Determine countermeasures and reduce the impact of threats		Report Secure Software

	design			•Secure Architecture Design •security controls on coding • reviews the security of Design and Architecture		Design
Implementation	Software construction	System integration	Software configuration management, Software documentation management	•reviews the Security Code		Report Secure implementation
verification	Software qualification testing, Software integration	Software operation, system qualification testing	Software verification, Software validation, Software review, Software audit, Software documentation management	•influence Testing		Comprehensive reporting of security
Release		System installation	Software quality assurance			
Support		Software acceptance support, software maintenance, software disposal	Software problem resolution			

- Describe and analyze their requirements and security objectives.
- Describe the threats and security controls of the software system.
- Provide information on threat modeling.

2) *Software Design Phase Output:* The output of the software design phase activities be expressed as follows:

- software implementation plan
- Details of the implementation of software components designed according to the requirements expressed.
- Determine countermeasures and strategies to reduce the impact of threats
- Described security controls on coding
- Review the security of Design and Architecture

3) *Implementation Phase Output:* The output of the implementation phase activities is expressed as follows:

- system Code generation
- Implementation of software components
- reviews code Security
- Identify vulnerabilities in the code
- Identify Countermeasures to fix vulnerabilities
- implementation Security reviews

4) *Software Verification Phase Output*: The output of the verification phase activities is expressed as follows:

- Software Tested
- Deliverable test
- Test plan
- Detailed test cases
- Results
- Test Report

VIII. CONCLUSION

So far, methods that have been proposed for testing the security of web-based applications postponed it to later stages of software development. But in evaluation approach of proposed security, we tried to identify necessary security activities according to security issues provided by OWASP ASVS and to test the security at all phases of software life cycle. In the proposed approach, security problems are identified in initial phases. After that by detecting attacks and countermeasures the safety of application is improved.

In this paper, we attempted to achieve an approach of proper security evaluation and reach to appropriate level of security by using security evaluating of lifecycle activities.

Make a move in the direction of presented levels, observing them and implementing each level based on standard and existing methods can help manufacturer to obtain a proper safe product. The proposed approach is suggested to evaluate the safety of web-based applications. It considered ASVS security evaluation, which is a security verification standard of web-based applications, as a base of security evaluations and activities. It determined the appropriate security measures in every phase of software development lifecycle based on this standard.

Such an approach could be considered for the following reasons:

- Being standard-based
- Ease of applying due to its simplicity and being systematic

- Evaluating security during the software development lifecycle
- Reducing risk because of security evaluation based on ASVS standard during production process of software product.
- Clarification of processes
- Considering security measures at all stages of software development, not only after implementation.

REFERENCES

[1] N. Modiri, F. Davvami, E. Alimohammadmalayeri, Software Metrics, 1st ed., ED. Tehran, Iran: Gange Nafis Publications, 2010.

[2] OWASP Testing Guide, Version 3.0, published by Open Web Application Security Project, [Online]. Available: https://www.owasp.org/images/5/56/OWASP_Testing_Guide_v3.pdf, 2008.

[3] B. Boehm, V. Basili, "Software Defect Reduction Top 10 List", IEEE Computer, Vol. 34, No. 1, pp. 135-137, January 2001.

[4] Di. Lucca, G.A., Fasolino, A.R., "Testing Web-based applications: The state of the art and future trends", Elsevier Information and Software Technology, Vol 48, pp. 1172-1186,2006.

[5] ISO/IEC, Information technology—Security techniques-Information security risk management, ISO/IEC FIDIS 27005, 2008.

[6] British Standard Institute, Information technology--Security techniques- Management of information and communications technology security - Part 1: Concepts and models for information and communications technology security management BS, ISO/IEC 13335-1, 2004.

[7] Cenzic Managed Security team, Application Vulnerability Trends Report, [Online]. Available: http://www.cenzic.com, 2013.

[8] G. Tian-yang, S. Yin-sheng, F. You-yuan., "Research on Software Security Testing", World Academy of Science, Engineering and Technology 69, 2010.

[9] G. Li, H. Dong, Q. Zheng, M. Zhou, Y. Guo, "Research on National and International Software Engineering Standard Bodies", in International Conference on Computational, 2009.

[10] Portillo-Rodriguez, J., Vizcaino, A., Ebert, C., Piattini, M.,"Tools to support global software development processes: a survey, in: Global Software Engineering (ICGSE)", 2010 5th IEEE International Conference On. pp. 13-22, 2010.