Evaluation security of web-based electronic payment systems

Mehdi mehdizadeh [1], Dr.Nasser modiri [2]

[1-]MSc, Department of Electrical, Computer & IT, Zanjan Branch, Islamic Azad University, Zanjan, Iran
[2-] Associate Professor in Department of Electrical , Computer & IT, Zanjan Branch, Islamic Azad University, Zanjan, Iran
post_mehdizade@yahoo.com

Abstract: Secure web-based electronic payment system is one of the main concerns of governments and individuals and entities throughout the world. However, determining the degree of safety of the system is still one of the most challenging and important issues in the world of computers. The method presented in this paper for evaluating the safety and security of electronic payment systems, e-commerce enterprise can set mail and other electronic payment systems to be used.

[Mehdi mehdizadeh, Nasser modiri. **Evaluation security of web-based electronic payment systems.** *N Y Sci J* 2014;7(6):37-48]. (ISSN: 1554-0200). http://www.sciencepub.net/newyork. 6

Keywords: electronic payments, network security, e-commerce

1. Introduction

Due to the intense competition between manufacturers and suppliers of products, e-commerce has provided a global opportunity that producer and supplier or customer irrespective of geographical distances, so in a wider range of through the service, mail between the international to find each other. Manufacturers or suppliers of products and services can use e-commerce spending less money anywhere in the world; have the appropriate client for their goods. On the other hand customers exactly according to your taste and budget easily and in the shortest possible time they desired service by international mail and in other ways achieve. Now, every one of his vision to define the e-commerce deals. But can be summarized briefly and the definition of e-commerce can be defined:

"Electronic commerce is the buying and selling of goods, services and information via communications networks."

This definition can be studied from four perspectives:

1- From the perspective of communication: e-commerce, i.e., to deliver the goods, services or payment via a computer network or any other electronic means.

2- From the perspective of business processes: e-commerce, i.e., application of technology in the automation of business transactions.

3- Services vision: e-commerce means that the aspirations of our customers and companies to achieve better quality, higher speed and better services at lower costs met.

4- The Online vision: e-commerce, the ability to buy, sell and exchange goods and information through the Internet or any other network Online.

Companies whose products and services traditionally offered forced to survive on its own technologies and new technologies bring among these technologies, e-commerce online using the online system also offers online banking and online payment methods.

Therefore, reliability and acceptability of e-commerce has an important role in the world and this is very important in the field of payment and payment security. Safety assessment process should be considered during the software lifecycle. It requires methodical process management and security testing during the production of engineered software. Considering that this is the web-based secure electronic payment systems, secure web application in this application will be considered. In this regard, web application security verification (ASVS) from OWASP, one of the most important standards in the security evaluation of these systems. So the way from the start of production software, security concerns and safety controllers in the software lifecycle to consider and assess security in all phases of the software implementation is needed.

26 percent of the test programs in 2012, including the vulnerabilities have been. Vulnerability to inject malicious code significantly in 2012 compared to 2011 increased. Many of the programs have been tested in several locations, including the vulnerabilities.

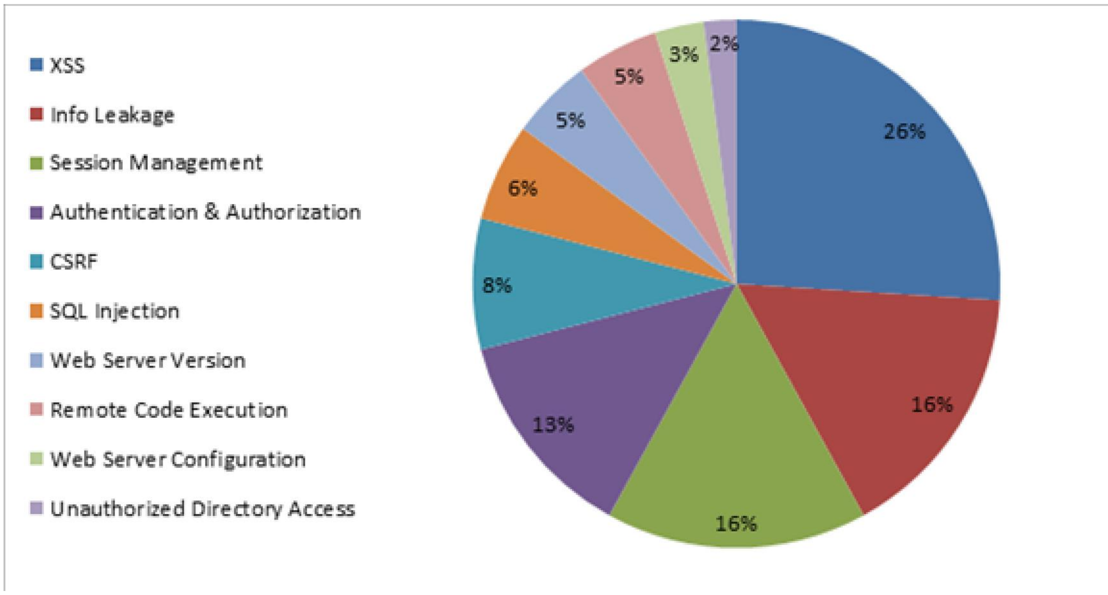

Figure 1 - Identify vulnerabilities in web applications

As shown in the figure, the highest vulnerability to inject malicious code has been. The continued absence of 2 and 3 errors, session management, and authentication and authorization access each 16%, 13%, CSRF (8%), inject SQL (6%), Web Server Edition (5%), remote code execution (5%), web server configuration (3%) and unauthorized access to directory (2%) of the total have been affected.

It summarizes the main objectives of the research were as follows:

1- Develop a conceptual model of secure electronic payment web-based information

2 - Development of web-based secure electronic payments

3- Identify the security requirements in electronic payment systems based on Web

4- Identifying vulnerable and addressing vulnerabilities in the electronic payment system

5- Evaluation of the security of electronic payment systems based on standard

6- Provide secure electronic payment method on the lifecycle of Iran Post Company

7- Provide a unified approach to the integration of electronic commerce in the Islamic Republic of Iran Post Company

One of the important features of Web applications, follow their client architecture - service transition. This means that Web application developers are able to achieve server-side and client-side processing of various technologies used. One of the important points of client-side processing is the extent of their reliance on the information on the server.

Among the features of these programs can include:

• Web-based application data for all users to share.

• User web applications are characterized by a large group of users of different ages and conditions of use web application.

• Web-based applications to get information from the forms they use.

• Web application servers are operating normally.

• It can become a portal.

• It is independent of the client operating system

Create a robust interface for using Flash, JavaScript for designing pages.

The structure of web applications:

Web apps are logically composed of three layers:

1. Persistence layer: data and uses the data stored on the user changes.

2. Layer processing: processing of information does the application usually is the most important task.

3. Layer Interface: User interface display applications.

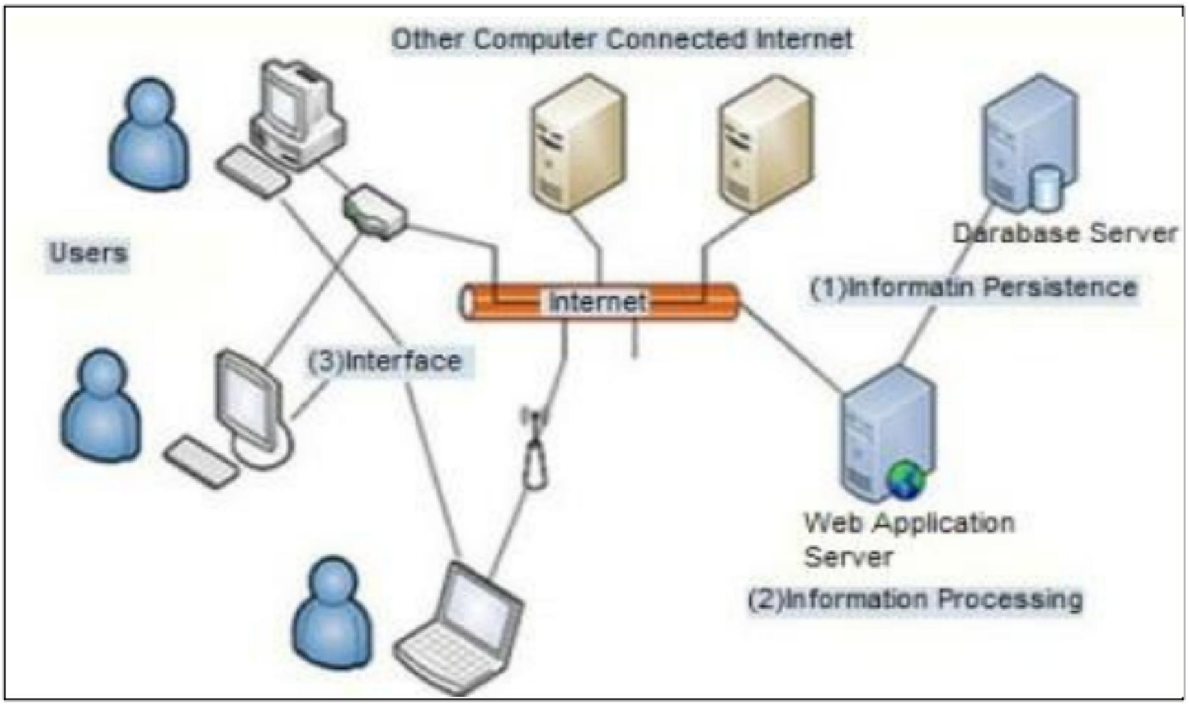
Other Computer Connected Internet
Users
Internet
Darabase Server
(1)Informatin Persistence
(3)Interface
Web Application
Server
(2)Information Processing

Figure 2 - Architecture of Web Apps

These layers are where the Web application is different. Figure 2 shows the architecture of web applications [10].

Materials and Methods:

The research described in this thesis based on research methods and methodologies and standards in the field of security evaluation system of the web. Finally, this paper aims at the maximum specification approved safety standard for evaluating the security of a web application is in four levels, segmentation as the conceptual basis of activities put it and then all the activities and security controls and requirements needed from the beginning of the production cycle, the software provides a systematic and organized. As e-commerce is still nascent and Iran Post Company operates as an island, and you pay is not the same; it's needed to be addressed to enable the integration of safety and security with international standards in terms of which the matter will be discussed in this article.

Intrusion and attack on the client side

Of client-side attacks, abusive or enjoying the opportunity to focus on the users website. When a user of a website is, trust between the two sides in terms of technology and also psychologically occur websites that a user expects to see the content of the valid. The user also expects that the use of a website it will not be attacked. An attacker may exploit different ways for users to use.

Content spoofing

This method of attack is to trick the user into believing that the content of the Website is lawful, and not from an external source [22].

Inject malicious code

One of the most common client-side attacks that XSS attacks. Briefly inject malicious or XSS, is vulnerability in the web applications through which an attacker is able to inject client-side script code in a user's browser to a web page and run the program. Inject and execute malicious script occurs when the application information of the user (or other unreliable sources) takes as input without that information to validate the output as a web page displays. Scripting language code, usually JavaScript, but other client-side scripting languages like Action Script, VBScript, and can also be used.

Review and user access control

An important security requirement in applications, that users access the data and functions to be controlled. In any application, there are different classifications for users; For example, common categories are: anonymous users, the typical user authentication and system administrators. Moreover, different situations, different users are allowed to access different data sets. For example, users should be able to read your mail program and can not read other people's letters. In most applications, web access management with a combination of three mechanisms is the following:

- Authentication
- Meeting Management

• Access Control

These mechanisms are interdependent and safety program depends on the strength of the weakest link in the chain. Failure of a component may enable an attacker unrestricted access to data and functionality found.

Ten jeopardize the security of web applications

The main objective of the ten risks is important and necessary training associated with the most significant vulnerabilities in web applications in most organizations, architects, designers and software developers to place. Ten major risk tackling vulnerabilities that offers a very good start for secure software is considered. Vulnerabilities in order of importance according to Table 1 are presented in 2013 [31].

Table 1: Ten major stake in web application security OWASP 2013

A1	Injection	Inject danger like (OS, SQL and LDAP) occur when untrusted data as part of a command or query is sent to an interpreter. Given by the invaders, the interpreter into executing unintended commands or accessing unauthorized data makes.
A2	Abaroken Authentication and Session management	Most of the applications that are part of the communication is related to authentication and session management, to be implemented properly. This allows attackers to attack passwords, tokens and use that identity to others.
A3	Inject malicious code	The danger occurs when the application is untrusted and non-secure data and applies it to a web browser without proper validation Blackberry. XSS allows attackers to execute script in the victim's browser which can result in the theft of user communication sessions, Hacking a website or redirect the user to a malicious Web site.
A4	Insecure Direct Object References	The danger occurs when the programmer, such as a file, directory or database references. Without access control or other protection, attackers can manipulate these references to access unauthorized data use.
A5	Security Misconfiguration	For security, require careful configuration of security for applications, frameworks, web servers, application servers, and we are. It is necessary to properly define all these settings, run and maintain as many default settings are not secure enough.
A6	Sensitive Data Exposure	Many Web applications do not properly protect sensitive data, such as credit cards, tax ID and authentication credentials, an attacker may be weakness of the protection for ID theft, identity fraud, credit card or other their crimes. Require additional protection such as encryption of sensitive data storage and transmission requirements.
A7	Missing Function Level Access Control	The majority of web application user interface access to levels seen just before the function check, but requires the access control checks each functional server side to be done.
A8	Cross Site Request Forgery(CSRF)	Request forgery attacks, forcing the browser to send a forged HTTP request, including the victim's session cookie and other identifying information to web applications are vulnerable. This risk, which allows remote attackers to force the victim's browser to send a request to the vulnerable application, thinks are legitimate requests received by the victim.
A9	Using Components With Know Vulnerabilities	Components, such as libraries, frameworks and other software patches, always run with full access. If vulnerability is exploited components, such an attack would lead to loss of important data or takeover server.
A10	Invalidated Redirects and Forwards	Web-based applications, regularly directs users to other web pages and other data to determine safe use metal plates. Without proper validation, attackers can redirect victims to malicious web pages or the pages are illegal.

Levels of application security review

The first level of the Automatic

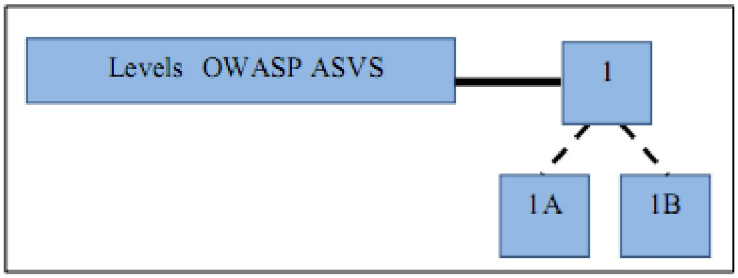

Figure 3 - Levels 1A and 1B in OWASP ASVS

At level 1, the components of the application is shown in Figure 4, may be individual or group of source files, libraries, or executable files are defined at this level, path or paths end-user requests that the application is being used, do not require identification or documentation.

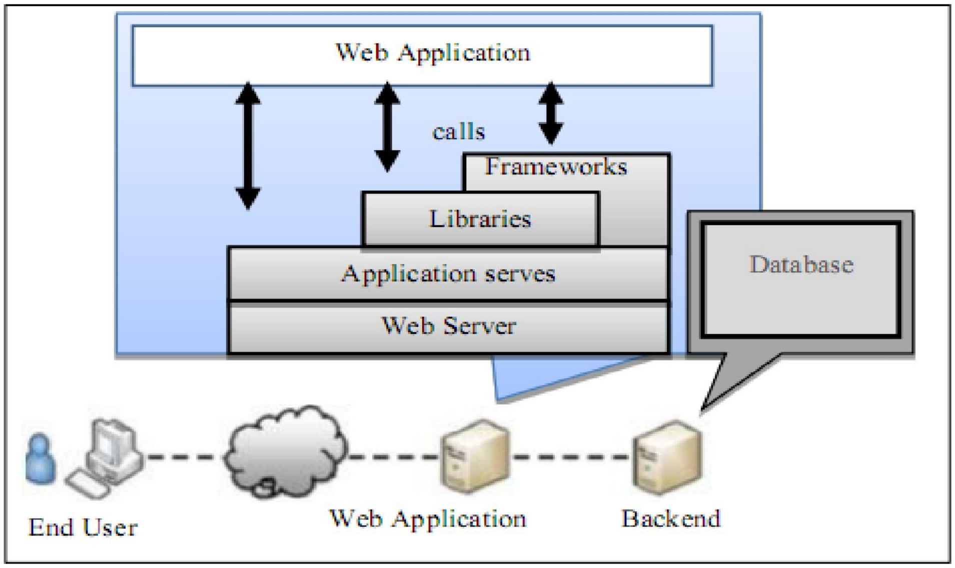

Figure 4- Architecture safe level of OWASP ASVS

The second level check Manual

The second level is composed of two components manufacturer:

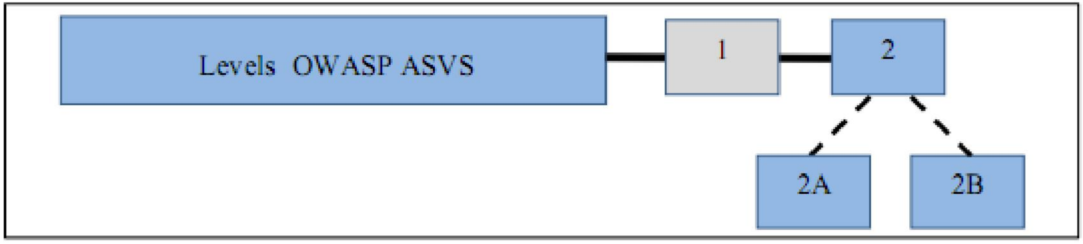

Figure 5- Levels 2A and 2B in OWASP ASVS

Second-level application components may be individual or group of source files, libraries, or executable files that define a high-level architecture can be an organizer.

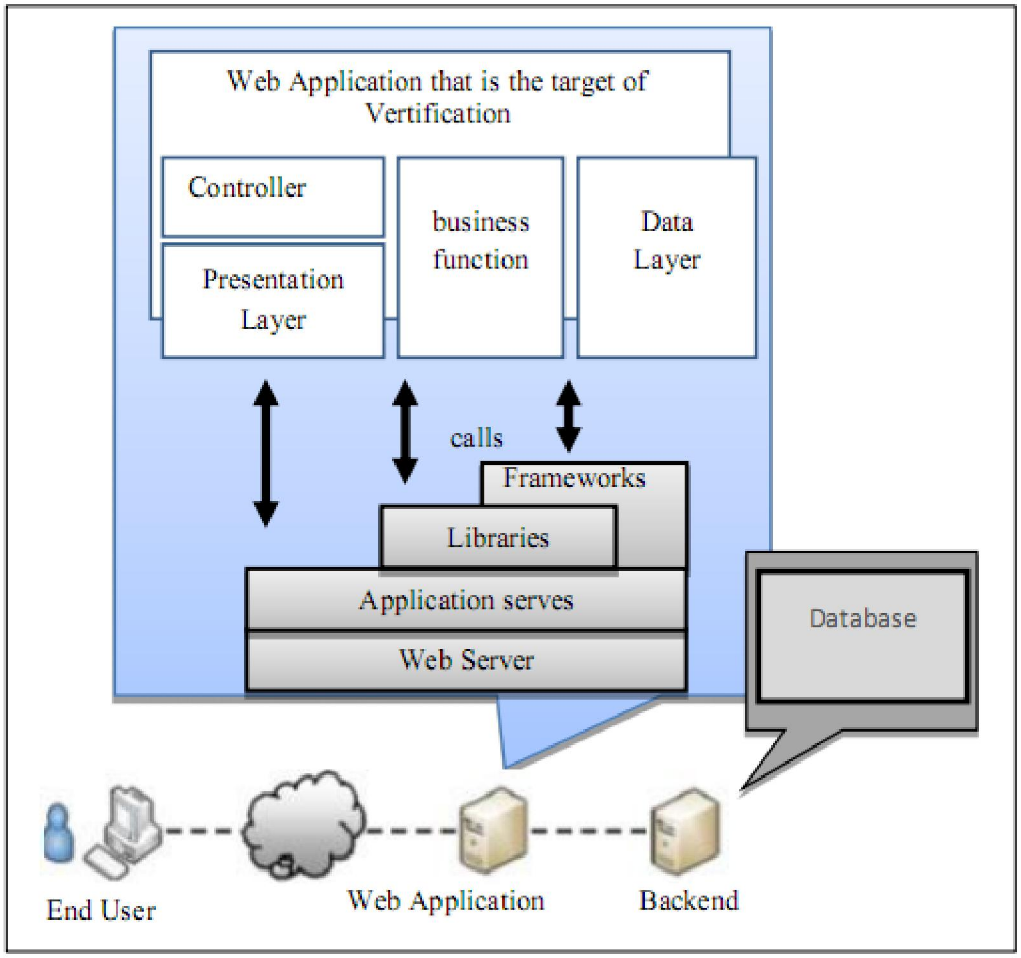

Figure 6- Example of the third-level secure architecture OWASP ASVS

As shown in Figure 6, the second path or paths that end-user requests that the application is taken to be documented, but it is not necessary to test all the way.

The third level of review

The third level is usually suitable for applications that manage their critical business transactions.

As shown in Figure 7, the third level is broken into several components:

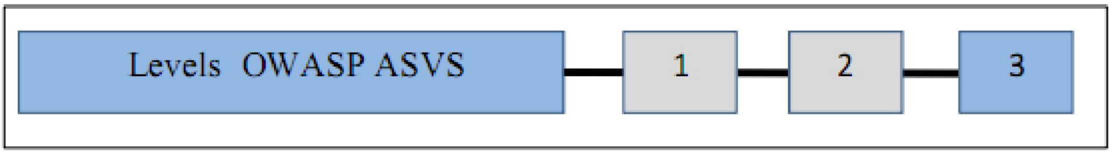

Figure 7: Level Three OWASP ASVS

As shown in Figure 8 is a high-level view of an end user requests a path or paths will cross, it should be documented. Also, all potential routes of a high-level view of the application must be tested.

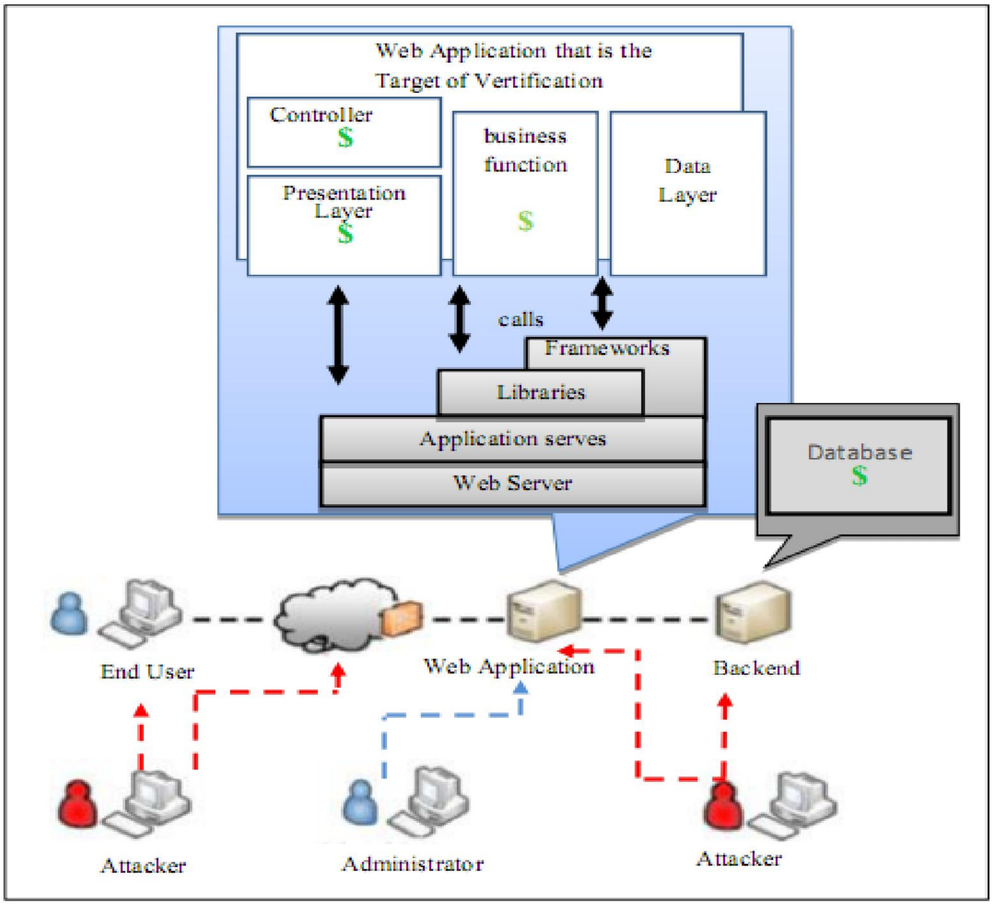

Figure 8- Example of third-level secure architecture OWASP ASVS

The fourth level of internal review

At this level, security threats are carried out by attackers and range checks to the third level expands to include all the code used by the application. As shown in Figure 9, the fourth component is not broken by its manufacturer.

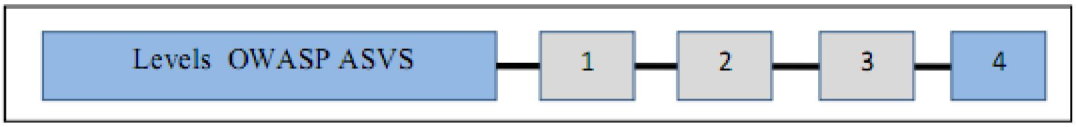

Figure 9- Level IV in OWASP ASVS

In Table 2, the range of security requirements for each of these areas, requirements that must be considered is specified.

Table 2: security requirements of the ASVS

Background	Security requirements evaluation
V1-safe architecture	V1.1-verify that all components in the application (including source files, libraries, executables,) are defined. V1.2-verify that all components that are part of the application, but the application to work properly it rely on are defined. V1.3-examine whether a high-level architecture is defined for the application (if the application developer has to do it. Review can provide evidence of a high-level design). V1.4-verify that all components of the application are defined in terms of security or commercial functions are provided.

	V1.5-check that all the components are defined in terms of security or commercial functions, but are not part of the application on the application to work properly it will be provided. V1.6-check whether the information provided is threat modeling.
V2-authentication	V2.1-check all the pages and resources that are required to authenticate, except those that are specifically intended to be public. V2.2-verify password fields, enter a password it will not show. V2.3-check that you took the maximum number of authentication attempts, the account is locked for a period of trial and error attacks (brute force) to prevent. V2.4-check that all controls are applied on the server side authentication. V2.5-check that all authentication controls (including libraries that call external authentication servers do) have a centralized implementation. V2.6-check that all authentication controls as safe (appropriate messages will be removed) fail. V2.7-check the authentication credentials are strong enough to resist common attacks and threats in the environment to resist. V2.8-check that all account management functions, at least in the initial attacks against the authentication mechanisms that are resistant. V2.9-examine the mechanism by which users can help, they change their own credentials and act against attacks that are related to the mechanism of initial authentication, insulate. V2.10-check authentication again before running any special software to perform critical operations. V2.11-check that the configuration for a long time, the authentication credentials is expired. V2.12-check that all decisions relating to record authentication. V2.13-checked the IDs and passwords a user ID with a unique number (internal) stored before storage, it should be done. V2.14-check that all authentication credentials for external access to the application, encrypted and stored in a protected location (not in the source code). V2.15-verify source implementation or use of the validation controls is not affected by any malicious code.
V3-Management Meeting	A V3.1-review meeting to manage and control the application of the default implementation of the framework is strongly recommended. V3.2-check whether the user authentication, the session is invalid. V3.3-check whether the session after a specified period of inactivity will expire. V3.4-check whether the session after a configurable maximum time the director will expire regardless of the activity (a complete collapse). V3.5-check all the pages that you access, authentication will need to have the exit link. V3.6-check whether the session ID will never be revealed, especially in the URL or in error messages. For this reason, you should check that URL rewriting session cookies are not supported. V3.7-check when entering the conference ID is changed. V3.8-check whether the session ID to authenticate again to change it. V3.9-check whether the session ID is changed or erased when you exit. V3.10-verify that only the session ID that is generated by the framework, the application will be considered valid. V3.11-review meeting, the authentication token is long enough to be random and threatening attacks to resist. V3.12-examine whether the cookie that contains the session ID tokens or are the amplitude and the direction can be initialized with a limited amount for that site. V3.13-verify source implementation or management control session is not affected by any malicious code.
V4 - Access control	V4.1- Verify that only specific users that they can have access to protected functions. V4.2- Verify that only specific users that they can have access to the URL.

	V4.3- Verify that only specific users that they can have access to their data files. V4.4-check that refers to objects are preserved and are available to any authorized user, only objects. V4.5-checked the directory search is not possible unless you purposely wanted. V4.6-examine whether users can only have access to the services for which they have permission. V4.7-verify that users can only access data they are authorized. V4.8-verify access control with fail-safe manner. V4.9-examine the rules that control access to the display layer is created by the server to run. V4.10-checked the account and character data and policy information, which they can use to control access, should not be altered except by those who have special permission. V4.11-verify that all access controls are performed on the server side. V4.12-examine whether a central mechanism to access protected resources exist. V4.13-examine the restrictions on entry and access to the application are created by businesses (such as daily transaction limits the duties of) should not be ruled out. V4.14-verify that all access control decisions are recorded. All pages should be recorded as a failure. V4.15-verify source implementation or use of access controls are not affected by any malicious code.
V5-validated input	V5.1-check that there is no buffer overflow or security controls to prevent buffer overflow. V5.2-verify an authentication model is defined and applied to all inputs. V5.3-check the input validation failure, causing it to reject or delete the entry. V5.4-check the character set such as UTF-8 for all input sources to be determined. V5.5-check that input validation is done on server side. V5.6-examine the application of a validation control unit for each type of data that is accepted to use. V5.7-verify that all input validation failures are recorded. V5.8-verify that all input data for conventional commentators have Downstream, the validation should be a priority. V5.9-check the input validation controls are not affected by any malicious code.
V6-coded output	V6.1-verify all data that is unreliable output, HTML are (contains elements of HTML, the attributes of HTML, data values JavaScript, CSS and block attributes URL) correctly apply the text to be encoded. V6.2-check that all controls are implemented server side coding. V6.3-check that all controls and coding, to encode all the characters unrecognized for certain interpreters, it is safe and appropriate. V6.4-verify that all data are unreliable output for SQL interpreters, the interface parameter to use prepared statements or they are coded correctly. V6.5-verify that all unreliable data those are output to XML, the parameter interfaces or are coded correctly. V6.6-verify that all unreliable data that are used in LDAP queries are correctly coded. V6.7-verify that all data are unreliable in the operating system command parameters are properly encoded. V6.8-verify that all data unreliable for any interpreter or other commentators that the above mentioned are the right encoding. V6.9-examine the output for any application that was encrypted by a security control unit is to the output type for the destination specified. V6.10- Check the source implementation or use of controlled output encoding not affected by any malicious code.
V7-encryption	V7.1-verify that all cryptographic functions are implemented on the server side. V7.2-verify that all cryptographic functions for fail-safe. V7.3-examine whether the secret keys are protected against unauthorized access (an original set of keys, a credit is applied to programs that are stored on disk and

	protected access to the configuration information is secured.) V7.4-verify that the passwords are hashed and stored when they are made. V7.5-check whether the encryption module failed to register. V7.6-evaluate the random numbers, random file name, GUID random and random fields, which are caused by methimazole encrypted, can not guess. V7.7- Check that all encryption modules are validated against FIPS140-2 or equivalent standards. V7.8-verify that all modules have been approved in accordance with the security policy of their published work. V7.9-check whether there is a clear policy on how to manage encryption keys and correctly review the policy to be run. V7.10-check code that supports using encryption modules are not affected by any malicious code.
V8-control and error logs	V8.1-examine the application, sensitive information such as session ID or personal information that helps with the attackers, as the output does not display error messages or stack. V8.2-check whether all the errors on the server, the server can be controlled. V8.3-check that all control registers are implemented server side. V8.4-verify control logic errors in the security controls to prevent access by default. V8.5-check that all events that contain the data are unreliable, as recorded in the software code, not run. V8.6-check the security event log files against unauthorized access and modification, are protected. V8.7-check whether a record is central to the implementation of the application it uses. V8.8-examine the application of sensitive data related to specific programs that may help the attacker does not register (such as session ID) V8.9- Review of the record, there is a tool that allows the analyst to record events based on specified search criteria. V8.10-examine the implementation or use of controlled registration code error, not affected by any malicious code.
V9-protect data	V9.1-review form that contains sensitive information, you can also enable client-side watermarking auto complete feature to is not present. V9.2- Check the list of critical data that the application process has to be defined and explicit policy for how to access and encrypt them there to run correctly. V9.3-verify that all sensitive data are sent to the server in the HTTP message body (do not use the URL parameters for sending sensitive data.) V9.4- Check that all sensitive data watermarking or temporary copies that are sent to the client are protected against unauthorized access to or after the authorized user to access, delete or are invalid. V9.5- Check that all sensitive data watermarking or temporary copies that are sent to the server to protect against unauthorized access to or after the user is allowed access to clean or be disabled. V9.6-check whether there is a way to critical data application upon completion of maintenance time required to remove.
V10-Security Community	V10.1-check whether a path can be any valid CA secure transport layer (TSL) server certificate is created and the server certificate is valid. V10.2-examine a failed relationship TSL will shift to an unsecure connection. V10.3-check all connections to be used for the TSL containing sensitive authentication data or functions. V10.4-check whether the line should be recorded TSL communication failure. V10.5- Check the valid path to be built and certifications for all customers using the correct link and falsify information are investigated. V10.6-evaluate how to communicate with external services that contain sensitive authentication data or functions to be performed. V10.7-evaluate the relationship with external services that involve sensitive data or

	functions of the minimum requirements an ID to use to work to be set up application. V10.8-examine the application of the standard TLS implementation uses. V10.9-verify for all communication, especially as UTF-8 character coding are defined.
V11-secure HTTP	V11.1-examine that transmission may contain invalid data. V11.2-evaluation of the program will only apply to a certain set of methods in HTTP, such as GET and POST accept. V11.3-examining the response HTTP, a header content type (content type) that specifies a character set is safe to include. V11.4-checked the flag HTTP Only for all cookies that do not require access from JavaScript can be used. V11.5-examining the secure flag for cookies that contain sensitive data, including session cookies, is used. V11.6-examining the HTTP headers in the request and the response are only ASCII printable characters. V11.7- Evaluation of the program is a powerful random token as part of all links and forms that are associated with interactions of or access to sensitive data to produce and review the application of the token for the current user in the processing of the requests to the appropriate review.
V12-security configuration	V12.1-check that all security-related settings are stored in places where access without authentication in the safe. V12.2-check that all access to the application if the application is not able to achieve its security configuration information and break. V12.3-check that all changes to the security settings that are managed by software security incidents are reported. V12.4-check the Save Settings to store the output is human-readable to facilitate audit.

References:

1. Moghadasi, A.; Challenges and strategies for implementation of e-commerce. 1. 2007; 2 (6) :22-56
2. Parveen, Sharma "Advance Technique for Online Payment Security in E-Commerce : “Double Verification” " International Journal on Computer Science and Engineering (IJCSE)Jun 2013
3. Al-hamami, Alaa Hussein, Fadi Ali Oqla Najadat and Mohammed Saad Abdul Wahhab, “WebApplication Security of Money Transfer Systems,” Journal of Emerging Trends in Computing and Information Sciences,“‘‘ March 2012.
4. E.Kazanavicius, V.Kazanavicius and A. Venckaukas, “Security Web Application by EmbeddedFirewall,” 2012.
5. Hu, Xiangyi, Guifen Zhao and Guanning Xu, ”Security Scheme for Online Banking Based on Secret Key Encryption,” Second International Workshop on Knowledge Discovery and Data Mining. IEEE, 2009.
6. G.Swapna and R.Pavani.Srivatsav, “Securing Web Applications By Analyzing The Logs Of The Database Server Or Web Server,” International Journal of Engineering Research and Applications (IJERA), pp.432-435, November- December 2012.
7. Chauhan, “Protecting Port 80 : Techniques for Eliminating Web Application Vulnerabilitie,” 2004.
8. Miettinen and Jarkko, “Security Aspects In Modern Web Application,” n.d, 2009.
9. L.Lorek, “New erip off maneuver: Swapping Price tage,” [Online]. Available http://www.zdenet.com/intweek/stories/news/04164269233700.html, March 2001.
10. W3C, [Online]. Available: http://www.w3c.org/.
11. OASIS, [Online]. Available: http://www.oasis-open.org/.
12. K.Tang, S.Chen, J.Zic and B.Yan, “A Performance Evaluation of Web Services Security,” in Enterprise Distributed Object Computing Conference(EDOC'06), Hong Kong, IEEE, Vol.10th, pp.67-74
13. Basic Security Profile Version 1.0, [Online]. Available: http://www.ws-i.org/BasicSecurityProfile- 1.0-2007-03-30.html.
14. ISO/IEC, “Information technology -- Security tecniques-Information security risk management”, ISO/IEC FIDIS 27005, 2008.
15. 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.

16. Glossary, [Online]. Available: http://www.enisa.europa.eu/activities/risk-management/current- risk/risk-management-inventory/glossary#G52.html.
17. Shirvani, R.; "offering a software lifecycle to identify security holes," MA thesis, University. North Tehran, 2012
18. Miettinen and Jarkko, "Security Aspects In Modern Web Application," n.d, 2009.
19. Curphey, Mark, Scambray, Joel and Olson and Erik, "Improving Web Application Security:Threats and Countermeasures," pp. 24-41, 2003.
20. Victor D. Sawma and Robert L. Probert, "E-Commerce Authentication - An Effective Countermeasures Design Model".
21. Labs,S, "A New Spoof: all frames-based sites are vulnerable," 2007.
22. Donaldson and M.E, " Inside the Buffer OverFlow Attack: Mechanism, Method and," 2005.
23. Thuemmel and A, "Analysis of format string bugs," 2006.
24. Faust and S, "LDAP Injection: Are Your Web Application Vulnerable?," [Online]. Available: http://www.securityfocus.com/bid/4278.html, 2005.
25. H. Shahriar and M.zulkernine, "MUSIC Mutation-based SQL Injection Vulnerability Checking," Proceeding of The Eighth International Conference on Quality Software, Vol.00th, pp.77-86, 2008.
26. n.d). Retrieved from cgisecurity: http://www.cgisecurity.net/papers header-based-exploitation.txt.
27. Katkar Anjali S.and Kulkarni Raj B, "Web Vulnerability Detection and Security," International Journal of Soft Computing and Engineering (IJSCE), September. 2012.
28. Pauli and Dr. Josh, "The Web Application Hackers Handbook," 2nd ed., pp. 17-37, 2011
29. OWASP Testing Guide v3.0, [Online]. Available: http://www.owasp.org, 2008.
30. The Ten Most Critical Web Application Security Risks, [Online]. Available: http://www.owasp.org, 2013.
31. OWASP Application Security Verification Standard, released version, published by Open Web Application Security Project, June. 2009.
32. System and software engineering-Software life cycle processes, ISO/IEC 12207, 2008.

5/23/2014