VALIDATING SECURITY CONSTRAINTS FOR E-COMMERCE WEB APPLICATIONS

Arti Sood[1], Raninder Kaur Dhillon[2]

[1]Department of Computer Science & Engineering, Guru Nanak Dev Engineering College, Ludhiana, India

[2]Assistant Professor, Department of Information Technology, Guru Nanak Dev Engineering College, Ludhiana, India

ABSTRACT

Software engineering is the basis of development of any software. In the past few years, security of software is of high concern. Therefore detection of errors which can lead to insecurity is quite important. In this paper, efforts are made for detection of errors by designing an application for e-commerce websites which takes file as input, checks it and outputs its constraint status. This paper aims to analyze the security assets of e-commerce websites that are threatened and help in proper detection of errors by validating various security constraints as many a times developers may overlook the constraints unapplied.

Keywords: Component; Security Assets, Security Constraints, Software Security, Software Testing

I. INDRODUCTION

Software can have a huge impact in any aspect of society. Software systems play a vital role in our modern society. Software engineering is an engineering discipline that is concerned with all aspects of software production from requirement specification to maintaining the system after it has been launched into use. In general software development processes consists of a particular set of software development practices which are performed by the software development team in a predetermined order.

In today's world where we are advancing towards e-commerce and business is happening over web/internet, security is an area of major concern for all kinds of applications. As the technologies advance, there is a wide increase in the threat to the security of software. Security is concerned with every phase of software development, right from requirements specification to design, implementation, testing, and deployment of software. Thus security of software has become an essential part, which has to be taken care of during the process of a software development.

Whenever we start the development of software, along with its functional requirements there has always been a need to think seriously about the security requirements as well. Security is vital to any software and terminates only when only when the software retires. It is also important for the development team to successfully identify and resolve all the security issues, during the development of software itself conforming that the deliverable product is secure and highly reliable to the users who use the software.

II. TESTING

Testing is the process of evaluating a system or its component(s) with intent to find whether it satisfies the specified requirements or not. Testing is executing a system in order to find any gaps, errors, or missing requirements in contrary to the actual requirements. Software testing is a formal process carried out by a specialized testing team in which a software unit, several integrated units or an entire software package are examined by running the programs on a computer [1]. It is a very important part of software engineering to check the quality of the product in context to which it is engineered to operate.

It is used to validate and verify that a software product meets its specifications and technical requirements. Software testing can be performed at any time during the software development process. Testing is performed in context to detect software failures so that appropriate errors may be discovered and corrected. It is an investigation done to find out the actual quality of the system developed, to check whether the given system works as expected and meets the technical requirements [2]. Testing compares the present state & behavior of system to the user expectations, specifications, standards, and target audience expectations as every software has its own target audience.

Designing effective test cases is important but the strategy used to execute them is also as important. There are a number of questions that arise while a system is being tested. To name a few of them, Should the entire program be tested? , Or should tests be run on small cases of it? At what time the customer is involved? And so on. So, a strategy for execution of testing is given by the project manager, testers and engineers involved in the project. And after that too, there may be lot of cost involved in testing phase as compared to any other software development activity.

As testing is an activity which is planned earlier and conducted in some step-by-step systematic approach, there are templates of testing into which test cases are placed. A number of software testing strategies have following things in common:

- At different situations or times, the testing approach is different.
- It starts from the component or unit level moving towards the integration of the whole system.
- Debugging is a different term than testing. Debugging is usually at implementation phase and may be accommodated within testing.
- Testers, for large projects, and developers, for small projects conduct testing.

Strategy for software testing includes the low level as well as high level tests which ensure small segments of code correctly working and major system functions against user requirements.

III. IMPORTANCE OF TESTING

Software testing plays a vital role in the field of software engineering. Testing is critically important for any software as without testing, software is, no doubt, developed but is at threat to be caught by any error at any time also causing failures. Testing helps in delivering the software that meets user specifications, prevents unexpected results and also improves the maintenance part of the application.

According to one survey software errors costs about 80% of the software development costs of a project that are spent only on identifying and fixing errors [3]. It assures the quality of any software. On an average, there are about 30-85 errors per 100 lines of code while the number reduces to 0.5-3.0 errors per 1000 lines of code after testing is undertaken. This conveys that still 0.5 to 3 errors are there for one thousand lines of code which are left undetected before delivering the product. Therefore, testing is considered as important as implementation and other phases of software development.

IV. RELATED WORK

Security engineering with patterns is currently a very active area of research [4]. Security patterns -- an adaptation of Design Patterns to security -- capture experts' experience in order to solve recurrent security problems in a structured and reusable way [5]. Applying design pattern enables developers to reuse it to solve a specified design issue [5, 6]. Recently [7] proposed a UML based validation method for security constraints. This method uses Security Design Pattern (SDP) and Security Requirement Pattern (SRP) that a developer uses to validate the security constraints [6, 7, 8]. A security pattern is reusable as a security package and incorporates security information; permitting programming engineers to outline secure frameworks like a security expert. This method leads to the identification of assets, threats, and countermeasures during an early stage of development [6]. The appropriate application of the pattern and the existence of vulnerabilities, identified in the early stage of the design model had been validated.

V. PROPOSED METHOD OF VALIDATING FILES

For any software, testing is conducted in order to ensure that there is no error while the system runs. Securing the system is as well important for the testers developing software. To make sure the security constraints are applied to the software which is under development, validations are important. This paper proposes a validation method for validating the security constraints for e-commerce websites.

This is important because if the validations are left unapplied, it can lead to the insecurity of software which may lead to failure also. Secure software is important to assure the quality of the software. Therefore, there is need of designing a system which can validate a file so that it can be easily known whether a given specified security constraint is applied or not. The testing method used here is introduced as an application that validates a file for various security constraints for a given security asset (module). It is an automated testing method for jsp and sql files used for developing e-commerce websites.

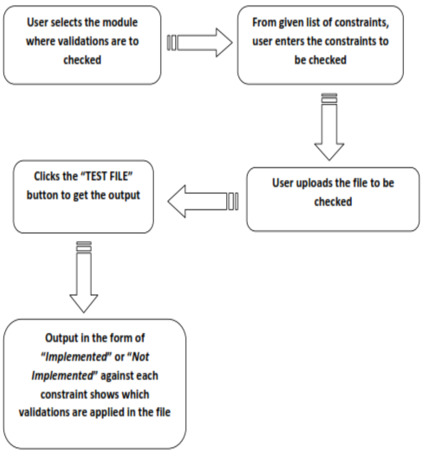

User selects the module where validations are to checked
From given list of constraints, user enters the constraints to be checked
User uploads the file to be checked
Clicks the "TEST FILE" button to get the output
Output in the form of "*Implemented*" or "*Not Implemented*" against each constraint shows which validations are applied in the file

Fig 1. Steps for validating a file

Figure 1 shows the various steps for validating various security constraints of a file. Before the steps were implemented, first all the security assets (functions) of e-commerce websites that can be threatened are analyzed in detail. These functions are used and implemented as different modules.

Step 1: In the first step, user selects the module where validations are to be applied.

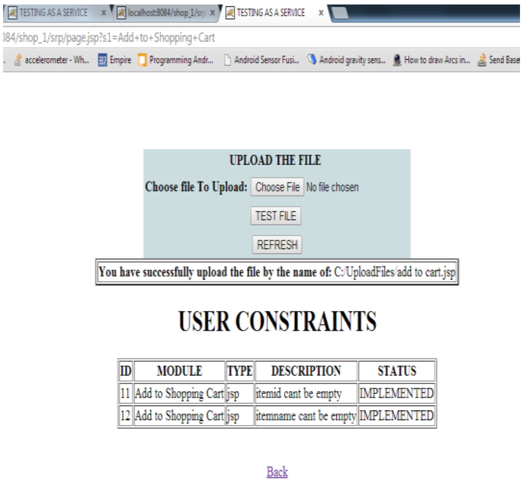
Step 2: In the next step, from the list of constraints, the user enters the constraints that need to be validated in that particular module.

Step 3: Finally the file (jsp or sql) is uploaded.

Step 4: After the file gets uploaded, it outputs the constraint status, showing if the security constraints are implemented or not.

VI. RESULTS AND DISCUSSION

This section presents the simulated results of validating the security constraints for web application testing. Figure 2 and Figure 3 below show some examples of the security assets that can be threatened. The security assets considered in this paper are add to cart and login modules which are selected by the user to validate various security constraints present in them.



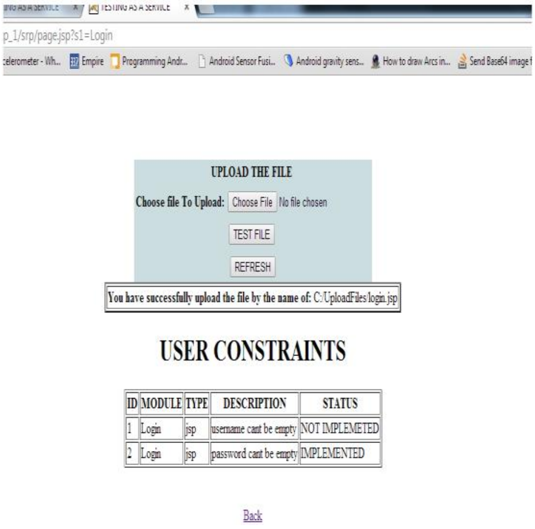
084/shop_1/srp/page.jsp?s1=Add+to+Shopping+Cart

Choose file To Upload:

You have successfully upload the file by the name of: C:/UploadFiles/add to cart.jsp

USER CONSTRAINTS

ID	MODULE	TYPE	DESCRIPTION	STATUS
11	Add to Shopping Cart	jsp	itemid cant be empty	IMPLEMENTED
12	Add to Shopping Cart	jsp	itemname cant be empty	IMPLEMENTED



p_1/srp/page.jsp?s1=Login

Choose file To Upload:

You have successfully upload the file by the name of: C:/UploadFiles/login.jsp

USER CONSTRAINTS

ID	MODULE	TYPE	DESCRIPTION	STATUS
1	Login	jsp	username cant be empty	NOT IMPLEMETED
2	Login	jsp	password cant be empty	IMPLEMENTED

Fig 2 Validating the Constraints of Add to Cart Module

Fig 3 Uploaded file to be tested and validating it

VII. CONCLUSION AND FUTURE WORK

Employing security patterns requires special training to the developers; it is a costly method as well. The previous methods for detection of errors are expensive and are available as tools. In this paper, efforts are done to detect the errors by validating security constraints as an application. It is also a less expensive method compared to other methods.

This application is specially designed for validating the security constraints during the development of software which also reduces the testing effort to a great extent. This application takes file as input, applies validations to it and outputs its constraint status. The status indicates if the security constraints are implemented or not. The proposed system also considers wide range of security assets (functions) in this context. The proposed system is

USE model independent as it doesn't depends on UML diagrams for generating security patterns. It checks the security constraints status for the web related files.

The scope of the proposed work is limited to validating the constraint status of jsp and sql files only. This work can be extended to implement testing as service so that multiple users can use this service.

REFERENCES

[1] Khaled M. Mustafa; Rafa E. Al-Qutaish and Mohammad I. Muhairat (2009), " Classification of Software Testing Tools Based on the Software Testing Methods", Second International Conference on Computer and Electrical Engineering, IEEE Computer Society Washington, DC, USA ©2009

[2] Chawla, Rshma and Dr. Mehta, Naveeta (2012), "Software Security Patterns In Security Engineering", IJRIM Volume 2, Issue 2 (February 2012) (ISSN 2231-4334).

[3] Khan, Ehmer M. and Khan, Farmeena (2014), "Importance of Software Testing in Software Development Life Cycle" IJCSI International Journal of Computer Science Issues, Vol. 11, Issue 2, No 2, March 2014

[4] Bouaziz, Rahma; Kallel, Slim and Coulette, Bernard (2013), "An engineering process for security patterns application in component based models", 2013 Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises.

[5] Huaxin, MU and Shuai, JIANG (2011), "Design Patterns in Software Development", Software Engineering and Service Science (ICSESS), IEEE 2011,Bejing.

[6] Moreira, Gabriel de Souza Pereira and Cardoso, Felipe Rafael Motta (2009), "Design patterns reuse for real time embedded software development", International Conference on Information Technology, New Generations, ITNG 2009, Las Vegas, Nevada, 27-29 April 2009

[7] Kobashi, Takanori; Yoshioka, Nobukazu and Okubo, Takao (2013), "Validating Security Design Pattern Applications Using Model Testing", 2013 International Conference on Availability, Reliability and Security. (ARES 2013), pp. 62-71, IEEE CPS, 2-6 Sep., Regensburg, Germany.

[8] Konrad, Sascha; Cheng, Betty H.C.; Campbell, Laura A. and Wassermann, Ronald (2003), "Using Security Patterns to Model and Analyze Security Requirements", In IEEE Workshop on Requirements for High Assurance Systems, Proc. of RE03 Workshop on Requirements for High-Assurance Systems (RHAS03), Monterey Bay, CA, Sept. 2003

[9] Petrova-Antonova, Dessislava; Ilieva, Sylvia and Stoyanova, Vera (2013), "TESSI: A Web Service Testing Tool", Research Challenges in Information Science (RCIS), IEEE Seventh International Conference, 2013.

[10] Taterh, Swapnesh; Yadav, K.P. and Sharma, S.K. (2012), "Threat Modeling and Security Pattern used in Design Phase of Secure Software Development life Cycle", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 4, April 2012 . ISSN: 2277 128X