RESEARCHERID
THOMSON REUTERS

[ICRTCET-2018]

ISSN 2348 – 8034
Impact Factor- 5.070

GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

THREAT MODELING FOR COMMERCIAL WEBSITES

Rashmi.N[*1], Dr. Sheba Selvam[2], Prathap.R[3] & Sushmitha.M[4]

[*1,3&4]Department of CSE, RRCE, Bangalore, India

[2]Associate Professor, Department of CSE, RRCE, Bangalore, India

ABSTRACT

Security is the major issue in the world of web. Proper implementation of the testing procedure plays a vital role in every web application. The critical complications have to be resolved accurately in day-to-day scenarios due to the large amount of data stored in the web applications and also due to the increased transactions on websites.

The threat modeling has an association between its security and usability. As the technology and threats are evolving in recent decades, the threat model certainly requires further elaboration of risks to manage attacks in productive approach.

The proposed model deals with the security of the commercial web application with OWASP (Open Web Application Security Project) standards. This technique recognises most critical risks/threats in the system. Therefore for this purpose, we utilize a powerful web scanner tool called Burp Suite. This imparts unified platform for the secured web application testing and also distinguishes the known and unknown vulnerabilities within the web applications.

Keywords: *Threat modeling; OWASP Standards; Security Testing; Burp Suite.*

I. INTRODUCTION

The structured methodology of threat modeling to security is to address top threats having greater impact to the application. It removes faults of security threats and prioritize remediation efforts by allowing system security staff to communicate. It decomposes the Web application in identification and rate threats that are great impact your system. This paper approaches to the prominent way in dealing with security of the commercial Web Applications. The existing system does not support for various Web Applications. To resolve such circumstances, this proposed approach use tool based approach with OWASP standards (Open Web Applications Security Project). This is an organization that provides security based criterions for testing approach. These OWASP standards raise consciousness on the application security to point the most critical risks facing organizations.

Threats represent vulnerabilities to the security of any effects. Threats might be malicious, or may be accidental, an insider, an outsider, so on. A software choice can solely result in many threats. Although there is no vulnerability, threats might occur. The main aim is to method the network security and finding their respective objectives and vulnerabilities. Further steps in solving for the vulnerabilities have been undertaken. The result is to determine ± instance of the effort to keep a system secure.

Threat modeling acts as a dynamic source for igniting new inventions in the field of testing. The major key to threat modeling lies in securing the system more dynamically. The factors that the threat modeling depends are- application added, delete or edit, upgrade and user requirements evolve. Threat modeling mainly corresponds to the creation of the security profiles, identification of the potential vulnerabilities, prioritization of these potential vulnerabilities and finally the documentation of the adverse events and the actions for every test cases according to the process flow.

In simpler, threat modeling is analyzing security of an application. The process of developing the threat model for commercial web application contains the essential factors: Asset- System resource that contains the data in the database, Threat- Any malicious intruder that might damage these assets, Vulnerability- Fragility in some aspects or

ISSN 2348 – 8034
Impact Factor- 5.070

features of the system that leads for possible threat, Attack- Any action, by user or system, that maltreat the asset, Counter measures- It addresses threats in the system and dilutes the respective risks in the system

II. LITERATURE REVIEW

The Threat modeling has a major scope in modern researches and technologies¶. The services in the cloud computing are very tensile. The recent threat assessment has been the greatest challenge to be resolved yet. Most of the approaches endeavor the solutions for destructive vulnerabilities. The major standards for the security testing include OWASP standards that act as the initiation to render proper threat detection.

One major cause for the destructive vulnerability is the potential click jacking. The analysis of the attacks on the click jacking methods with the defense scheme for androids is very intensive [10]. A defensive scheme approach describes the threat that cause android platform. These have an action on the minor impact against the system. Click jacking has an procedure to be implemented- click safe, a client-based tool to provide the security and reliability against these attacks [8].

Threats may also result in security misconfiguration. Thus this cause the greater lose for user data. Based on logic-programming approach, analysis of the vulnerabilities in recent advanced smart phones has been undertaken [3]. The tools that audits automated security with concern to the configuration settings towards server side with web applications are developed [2]. They have the server packages for MySQL, PHP, and Apache over three operating system platform.

The security challenges provide consistency, rigid nature for testers for future threat testing possibilities. This may have protection configurations and constrain-satisfaction problems

This also has the approach towards VPN/IPSec preservation. Most of the android platform notices the major threat occurring strategies and can also be solved with Multiple Criteria Decision Analysis (MCDA) [9]. A case study also distinguishes the user defined configurations around 561+ android devices. They have the direct approximation to the security risk level.

Cross domain scripting attacks also have the similar scenario with higher complexities. They have an intensive research to be manipulated in various fields of technology. They also utilize the domain names gain information that is relevant to attacker from website [1]. Hence attack-free application is used without threat possibilities. Threats occurring in cross platform also have the effect on the compiler [6]. They are developed by concentrating on the end users.

For both the developers and the tool users, the benchmark plays a significant part. Thus DomXssMicro, a micro benchmark is been designed [4]. In this, 175 test cases are illustrated. These should have proper encoding of the trusted data and executed framework respectively [5]. The generator has better test coverage than industry best practices. The other threat that might occur in security testing relying on OWASP standards is SQL Injection [11]. Most common in penetration testing approach is this issue.

Here the above researches concentrate on the potential click jacking, system misconfiguration and cross domain scripting as the major approach among the OWASP Top Ten standards. We have also implemented for the same vulnerabilities with the most powerful web scanning tool- Burp Suite. Thus this tool not only focuses on these threats but also solve for other vulnerabilities in the security testing process.

III. PROBLEM DEFNITION

The project has a problem definition of solving threats for commercial web applications. The threat model that is developed for threats for a system is more flexible for any web applications. The development stage of any

application contains the code for implementation. Thus intermediate code has to be developed for running the code efficiently. As a result, the list of threats is outputted. This list of threats has to be processed repeatedly until the threats are completely erased. This proposed system provides models for various threats in web application. It helps in easy implementation for threat modeling concepts to solve vulnerabilities. For instantaneous applications, it can be used. This approach is the gateway for the vulnerabilities that might occur in the processing of any applications of websites. The main concern of this methodology is to solve threats that cause inconvenience for the end users. Thus the interface that is developed should be more flexible for usage and implementation .

IV. OBJECTIVE OF PROPOSED SYSTEM

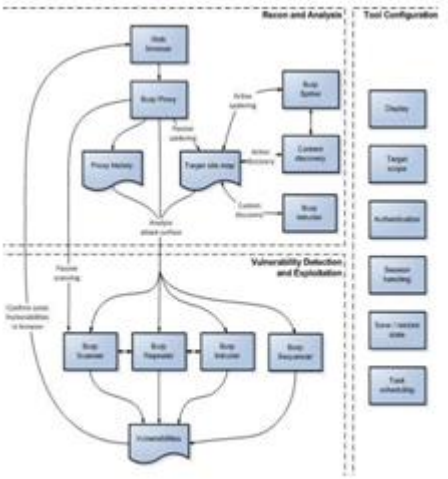

Fig1. Burp testing workflow diagram

The Fig1, specifies both the architecture and the flow diagram for burp suite tool functioning.

Here the main components in the burp are ± Burp Target, Burp Proxy, Burp Spider, Burp Scanner, Burp Intruder, burp Repeater, Burp Sequencer, Burp Decoder, Burp Comparer, Burp Extender, Burp Options and Burp Alerts.

- The Burp target contains the Target Site Map- with detailed information about the target applications and Proxy history- as the control to Burp Proxy tab.
- The Burp Proxy helps in intercepting, viewing and modifying all requests and the responses between the browser and the web server.
- The Burp spider helps in automatic dragging (crawling) of the web application. This can be used at any instance of time during testing process. It generates application content and functionality.
- Burp Scanner helps in automatically discovering security threats. It supports for the penetration testers and major component in workflow.
- Burp Intruder acts as an automated tool for customized attacks against web applications. Any task that arises during the testing process is manipulated by intruder.
- Burp Repeater is a simple tool for manually manipulation of the individual HTTP requests, and it also analyzes the responses of the web application. The request to Repeater can be sent from anywhere within Burp, modify the request and issue it again and again.

- Burp Sequencer is the tool that analyzes the standard of sample data items.
- Burp Decoder is used to transform the encoded data into raw form or vise versa. It depicts many encoding techniques.
- Burp Comparer is the tool for comparison between any two items of data.

- Burp Extender helps in burp extensions or any user code or the third party code
- Burp functions and Burp options helps in additional information for the testing process and for various tools respectively.

V. RESULTS AND DISCUSSION

The outcome of the security testing process is being discussed here. This process is simplified into four main modules namely- Configuration of the web browser, Setting burp intercept proxy, Web application analysis and Scanning vulnerabilities.

(1) Configuration of the web browser

Fig1. Configuration of the web browser

The Fig1. explains the processes that take place in the Mozilla browser. The main intension of the process is detailed in above figure. Set settings ³options´ in Mozilla browser. Set your browser to use burp as a proxy. Set the target URL in burp. Set manual proxy configuration. First step of security testing is to make sure the browser that is to be configured is installed properly. Here we have to set proper browser to run the web application. After the web browser is installed, we need to install the tool called ³burp suite´. When the software tool and the browser is ready, then we need to configure between both in order to do build secured connection between both of them. Thus the configuration settings to achieve this are as shown in the above figure. Manual setting configuration is should be done as the above settings. Thus after clicking OK, the configuration is set manually.

(2) Setting Burp Intercept proxy.

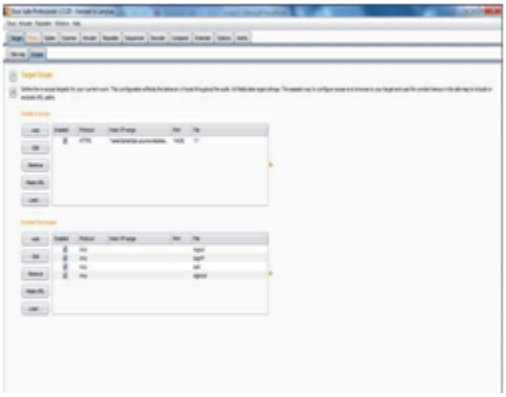

Fig2. Settings for the Burp proxy for browser to interact

The Fig2. explains the burp intercepting methodology in binding the web application browser and the Burp suite. Set Burp settings for Target Tab. Copy the URL that has to be tested to the clipboard and paste URL in include in scope target. Proxy set to the Intercept off. After setting with the previous settings page, the setting for the burp should be done. This can be done using the above method. The specific URL is to be specified in order to set the connection to

that website with burp. After every step of the this process, then the actual configuration is set completely and this connection is secured for the browser [3]Mozilla Firefox´. This is very crucial step for testing process to be implemented.

(3) Web Application Analysis

Fig3. Running web application in the web browser

The Fig3. describes about the web application that the project is concerned. This also explains the running the web application to which the testing should be done. Enter the credentials appropriately for e-mail and password fields. Make sure the application running in the Mozilla Firefox is configured properly. The web application to be tested has to be is logged in in the browser. Then the essentialities of the specific fields have to be given as a valid inputs. This has an effect on the Target site map area of the burp interface. The main effect of any control that are passed is directly induced in the main frame of the desired output location of the burp along with the request and response codes respectively.

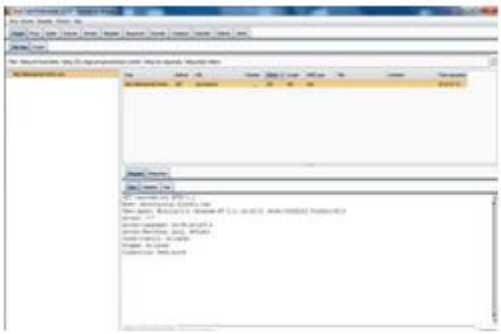

Fig.4 Respective parallel process that are encountered when browser and Burp are linked together

The above the Fig4 explains about the parallel process that encountered when the browser and Burp are linked together. The above screen specifies the backend web application that is running on the Mozilla browser. Every web application that is processed is scanned and contains the request and response codes respectively. The respective figure shows the entire request and the response codes for any application that is running in the web browser portal as the backend process. This is the level where the particular web application interacts with the burp directly through secured connection. Every website that is so executed in the Mozilla is listed in the Target site map of the proxy tab. And the necessary codes with respect to that website are given in separate tab portal of the proxy component tab.

(4) Scanning Vulnerabilities.

Fig5. Report of vulnerabilities that are found while scanning for a web application

The above Fig5 details about the threat list or vulnerabilities list that is discovered during the testing of the web application. The reports are generated for every threat in the list separately and individually. The further enhancement is to solve for these threats are discussed in later stages of the project implementation. The report thus generated contains the brief history and information about the specific threat that is so found. Thus this is the stage that the report for specific threat or the vulnerability is generated individually. The solution for these is to be found in the later stages for the complete implementation of the burp suite pro version for the web application [3]selectsmartqa.azurewebsites.com´.

VI. CONCLUSION

In the evolution of latest technologies, the innovations in field of web application security has a massive improvements. The threats that are found during the security testing process may include basic standards to be followed and is governed by the OWASP standards as significant protocols. The OWASP Top Ten has critical impact that may usually occur during the testing process. This paper discusses based on the three major threats- Potential Click jacking, Security Misconfiguration and Cross Domain Scripting.

Thus, the existing system doesn¶t solve for these threats as their implementation is only for limited number of threats to be scanned. They doesn¶t possess proper approach to the testing methodology with specific threat model and was very insecure. In our proposed procedure, large number of the vulnerabilities can be scanned and solved for a single web application using burp suite tool. Minute dissimilarities can also be clearly found and solved. .

REFERENCES

1. *al Azmi, Sulaiman, and Ahmad Raza Khan. "A comprehensive research on Xss scripting attacks on different domains and their verticals." Computer Science and Network Technology (ICCSNT), 2015 4th International Conference on. Vol. 1. IEEE, 2015.*
2. *Eshete, Birhanu, Adolfo Villafiorita, and Komminist Weldemariam. "Early detection of security misconfiguration vulnerabilities in web applications." Availability, Reliability and Security (ARES), 2011 Sixth International Conference on. IEEE, 2011.*
3. *Han, Zhihui, et al. "Systematic analysis and detection of misconfiguration vulnerabilities in android smartphones." Trust, Security and Privacy in Computing and Communications (TrustCom), 2014 IEEE 13th International Conference on. IEEE, 2014.*
4. *Hashizume, Keiko, et al. "An analysis of security issues for cloud computing." Journal of internet services and applications4.1 (2013): 5.*
5. *Mohammadi, Mahmoud, Bill Chu, and Heather Richter Lipford. "Detecting Cross-Site Scripting Vulnerabilities through Automated Unit Testing." Software Quality, Reliability and Security (QRS), 2017 IEEE International Conference on. IEEE, 2017.*
6. *Mulla, Fatima, Savitha Nair, and Aditi Chhabria. "Cross Platform C Compiler." Computing Communication Control and automation (ICCUBEA), 2016 International Conference on. IEEE, 2016.*
7. *Pan, Jinkun, and Xiaoguang Mao. "DomXssMicro: A Micro Benchmark for Evaluating DOM-based Cross-Site Scripting Detection." Trustcom/BigDataSE/I SPA, 2016 IEEE. IEEE, 2016.*

8. *Shamsi, Jawwad A., et al. "Clicksafe: Providing security against clickjacking attacks." High-Assurance Systems Engineering (HASE), 2014 IEEE 15th International Symposium on. IEEE, 2014.*
9. *Vecchiato,Daniel, Marco Viera, and Eliane Martns. ³Riak assessment of user-defined security configurations for Android devices. ³ Software Reliability Engineering (ISSRE), 2016 IEEE 27th International Symposium on. IEEE,2016*
10. *Wu, Longfei, et al. ³Analysis of clickjacking attacks and an effective defences scheme for android devices.´ Communications and Network Security (CNS),2016 IEEE Conference on. IEEE, 2016.*
11. *Xiao, Zeli, et al. "An approach for SQL injection detection based on behavior and response analysis." Communication Software and Networks (ICCSN), 2017 IEEE 9th International Conference on. IEEE, 2017.*