International Journal of Innovation, Management and Technology, Vol. 2, No. 1, February, 2011
ISSN: 2010-0248

A Study of Various Approaches to Assess and Provide Web based Application Security

Dhanya Pramod

***Abstract*—World Wide Web has grown in leaps and bounds and provides a promising platform for hosting applications. The web applications are developed without being taking care the criticality of security aspects and thus prone to attacks. The various efforts made by researchers and open forums would help to develop secure web application development, deployment and maintenance. This paper brings forth various aspects and work done to incorporate security in web based applications. An overview of security assessment methods is also included.**

***Index Terms*—webapplications; security; vulnerability, attacks**

I. Introduction

Web applications are always available on the internet and accessible any time. The vulnerabilities present in the application can be exploited by an attacker to abuse the functionality of the application or leak the data associated with it. There are several well known attack techniques to penetrate the web applications. There have been tremendous research efforts in this field to identify and plug these security vulnerabilities. The best known efforts, methods and tools to deal with security issues have been discussed in this paper.

Web Application security Consortium (WASC) [28] gives an open platform for security professionals, academia, software developers, software vendors and system auditors to access the latest security issues and countermeasures. The web application security project that can be accessed online contributes to develop and promote industry standard terminology for describing the issues. The efforts of the consortium bring forth the attacks and create awareness of the same among various stake holder communities. It also classifies various types of attacks using a cooperative effort and publishes as Web Security Threat Classification.

The WASC also facilitates documentation of the details of each class of attack and give a name to each class of attack. It also provides a structured way to organize the classes of attack. WASC is the leading contributor of dissemination of security related issues and takes efforts to provide a consistent language to the same. According to the survey conducted by WASC following are the cause for critical web hacking incidents. Improper Output Handling, Insufficient Anti-Automation, Improper Input Handling, Insufficient Authentication, Application Misconfiguration, Insufficient Process Validation, Insufficient Authorization, Abuse of Functionality, Insufficient Password Recovery and Improper Filesystem Permissions

Open Web Application Security Project is a worldwide not-for--profit organization to reach various communities and makes the security issues more clear and understandable. It allows people to visualize security and to make necessary steps to take care of the issues. OWASP conducts security survey every year to identify the vulnerabilities that have caused attacks and the top ten vulnerabilities of 2010 are Injection ,Cross-Site Scripting ,XSS) ,Broken Authentication and Session Management ,Insecure Direct Object References ,Cross-Site Request ,forgery (CSRF) ,Security Misconfiguration ,Insecure ,cryptographic Storage ,Failure to Restrict URL Access ,Insufficient Transport Layer Protection and Invalidated Redirects and Forwards

If you look at the previous year's top 10 and current, it is observed that few have been added to the list and some have been removed. Security Misconfiguration was one of the critical loopholes in 2004, but not much exploited later on has now positioned well at 6th rank in 2010. Invalidated redirects and forwards has drawn the attention of attackers in this year. Injection attacks and cross site scripting continues to top the list while the exploits on insecure cryptographic storage and failure to restrict url access has been increased.

In addition to Web Application Security Consortium there exists lot many organizations and individuals who work towards security attack prevention. To name a few SPI Dynamics, Symantec, Security Forge, IBM, Microsoft, IEEE, ACM etc.

SPI Dynamics [15] has a series of whitepapers describing various steps in building secure web application by incorporating it into the development phases, various attacks and its aftereffects, web application security assessment etc. Symantec with its recent acquisitions provide end to end security solutions and has become a pioneer in the security products industry.

II. Approaches To Application Security

The research community worldwide has looked into security aspects of web based applications with utmost care and bring forth methods to incorporate security at various levels. The most accepted approach is to incorporate security in the software development cycle itself. Secure modeling and model driven development approaches are becoming popular in this context. The access to web application has been restricted by some of the approaches and input to the

Manuscript received November 23, 2010.

Dr.Dhanya Pramod is currently working as the associate professor at Symbiosis Centre for Information Technology, A constituent institute under Symbiosis International University.

application is being filtered by most of the approaches. The protection could be done at gateway level, operating system level or application level. As the existing web applications with vulnerabilities have been exploited by hackers, application scanning approaches are significant to assess the security loopholes. The various approaches have been classified and the significant work done in these categories have been discussed here.

A. Secure Modeling Approach

Jurjens [13] defines a general approach to security concepts modeling and focus on the importance of incorporating security concerns during software development life cycle. The approach can be extended to design security notations and models for application specific counter measures of attacks.

J Juurjens in his work "Principles of secure software design". [14] explores the needs of developing secure-critical systems as there are many well-known examples of security weaknesses exploited in practice. Here he proposes a systematic methodology to aid the difficult task of developing security-critical systems in an approach based on the notation of the Unified Modeling Language.

He has presented the extension UMLsec of UML that allows expressing security-relevant information within the diagrams in a system specification. UMLsec is defined in form of a UML profile using the standard UML extension mechanisms. In particular, the associated constraints give criteria to evaluate the security aspects of a system design, by referring to a formal semantics of a simplified fragment of UML. The contribution of this work can be adopted by code generation tools and solution architects to design and develop a secure system.

B. Database Access Level Filtering

Injection attacks are the most critical ones and sql injection and code injection are the most common among the exploits. As database is a part of each and every application the work done in handling the database security is being reviewed.

William Halfond [29] in his paper "Protecting web applications using positive tainting and syntax aware evaluation" proposes a tool to prevent SQL injection. This approach focuses on trusted data identification rather than malicious data. The method marks trusted data at character level and a security aware evaluation of query strings are done at database access level. In this way any possibilities of attack in the input string is prevented. This dynamic analysis tool uses a lightweight approach to implement the approach. In java based applications the JDBC library method invocations are monitored and database interaction points are identified. Syntax checking module is inserted at this point. No modification done at the application level.

This method has the potential to filter any kind of sql injection attack as it is based on sql syntax aware positive filtering. Any kind of attack that may pop up in future will not defeat the purpose of this approach.

C. Security Analysis Framework

Xiang Fu, Xui Lu [30] in their work "A static Analysis framework for detecting SQL injection vulnerabilities (SAFELI) proposes a static approach to sql injection identification. The work proposes a framework that uses compile time vulnerability detection. The approach uses a 2 step process that consists of white box analysis method to analyze byte code and a hybrid constraint solver to analyze the input string which can handle a combination of Boolean, int and string variables. The SAFELI framework is primarily developed for ASP.net applications. It has been designed in a modularized way with the following components.

MSIL instrumentor: Byte code is analyzed and every SQL statement submission hotspots are tagged to trigger the constraint solver.

Symbolic Execution Engine: When hotspots are reached a library of preset attack patterns is consulted and a hybrid constraint is generated. It is then sent to the solver

Library attack Patterns: Regular expressions are used to store attack patterns.

Constraint Solver: Matches the constraints and input to identify strings that satisfy the constraint.

Test Case Generator: Injects the values in to HTML fields and posts the web page back to server. Use heuristics to analyze response.

This work completely concentrates on Microsoft implementation and similar approach can be applied to other open source technologies.

The database vendors have taken efforts to filter the information that has been passed to the users because the error messages could be exploited for the data leakage. Also the use of compilers for sql execution has been recommended instead of interpreters to avoid the dynamic injection and formation of sql statements. Even though these guidelines are known and implemented, sql injection is still a major catch hole.

D. Operating System and Compiler level Protection

Buffer overflow attacks were prevalent earlier and the popularity has been reduced in recent years. The attack leads to the execution of malicious unwarranted code and create undesired situations on your system. The protection methods can be broadly classified into two, kernel level and compiler level. In both of these cases the memory area that can be accessed by a process is restricted to avoid execution of unwanted code.

"Comparison of available tools for buffer overflow" by John Wilander and Mariam Kamkar [10] discusses the buffer overflow concern. This paper gives a good illustration of the tools available for buffer overflow prevention. It contains an exhaustive analysis of buffer overflow attack scenarios and categorized them. The paper does not talk about java/.net based applications buffer overflow issues and is focused on c/c++ based applications. Stackguard, Propolice, Stackshield etc are some of the commonly used tools for buffer overflow prevention. The most appealing prevention method is the operating system level Data Execution Prevention (DEP) , a security feature included in modern Microsoft Windows operating systems that is intended to prevent an application or service from executing code from a non-executable memory region. This helps prevent certain exploits that store code via a buffer overflow.

E. Client Based Security

The majority of the attacks that are targeted at web applications are done through user inputs. Filtering of user inputs is inevitable in this context. The work "A proposal and implementation of Automatic detection/collection system for Cross site scripting vulnerability" by Omar Ismail and Masahi Etoh [20] discusses the cross site scripting vulnerability. Their work is based on the client side XSS detection and manipulation of request and response. As the filtering is done at client side the possibility of carrying the malicious input to server could be avoided. Response filtering facilitates the sanity of output.

F. Application Gateway Level Security

"Abstracting application level web security" by David Scott and Richard Sharp [7] suggests some ways in which the web applications can be protected. They have illustrated the difficulties that are inherent in adding security to these applications. They emphasize on security that is beyond the restrictions of technologies, server and database which is used for the development and deployment. They have presented a structuring technique that incorporates security policies. A specialized language called security policy description language was proposed to program an application level firewall (security gateway). The program written in SPDL is compiled and executed on firewall. Firewall dynamically analyzes this http request and response using SPDL to ensure security. The stress in this paper is given to defend form modification, sql attacks and cross site scripting. They gateway does client side form validation and stick to authenticated data passing using Message authentication codes; and thus common attacks are prevented.

An automatic Defense Mechanism for malicious injection attack by Jin-Cherng Lin and Jan-Min Chen [12] presents an application level security gateway to filter malicious input that lead to scripting attacks. This method is targeted at binary form of software while our approach is based on source code level compile time aspect weaving. This approach gives error page in case any malicious code is found.

G. Security Standards and Markup Languages

OASIS (Organization for the Advancement of Structured Information Standards) has already taken efforts to propose security interoperability standards. AVDL (Application Vulnerability Description Language) creates a uniform way of describing application security vulnerabilities using XML. Another standard SAML (Security Assertion Markup Language) deals with attributes and authorization of authenticated entities. XACML (Extensible Access Control Markup Language) describes way to denote access control using policies.

H. Web Application Scanning Approach

Yao-Wen Huang, Shih-Kun Huang, Tsung-Po Lin, Chung Hung Tsai in their paper "web application security assessment by fault injection and behavior monitoring" [30] discusses a platform for assessing web application security. In the mentioned paper they analyses the design of web application security assessment mechanisms in order to identify the attacks to which they are susceptible and the loopholes that have given a way for the attackers. An effort has been made to suggest methods of sql injection detection and cross site scripting. They have given stress to various testing techniques including dynamic analysis, black box testing, fault injection and behavior monitoring to assess web applications. A tool has been developed to test real world situations named web application vulnerability and error scanner (WAVES an open source project available at http://waves.sourceforge.net). Also a comparison of this tool with other tools is provided. WAVES is a feasible platform for accessing Web Application security.

Their proposed mechanism uses a crawler interface that incorporates and simulates web browser. A complete reverse engineering process to identify all data entry points that can be holes for attacks are identified in the first assessment phase. The syntax and semantics of an input field is extracted using an intelligent form parser. A self learning knowledge base based on the Topics model provides semantically correct values for a form field automatically. Behavior monitoring is done to detect malicious scripts. To eliminate false negatives the NRE (Negative Response Extraction) algorithm is used and allows "deep injection". An injection knowledge Manager is used to formulate an initial input pattern to retrieve a negative response page. Most possible injection patterns are then formulated using an automated form completion algorithm. After sending the injection WAVES analyses the resulting pages using the NRE algorithm. So many user generated events are provided as test cases to facilitate crawl thoroughness and results in more comprehensive behavior observations. It detects both known and unknown malicious scripts.

As an interface between testing techniques and Web applications, WAVES can be used to conduct a wide variety of vulnerability tests, including cookie poisoning, parameter tampering, hidden field manipulation, input buffer overflow, session hijacking, and insecure server configuration–all of which would otherwise be difficult and time-consuming tasks.

Stephan Kais, Engin Kirda in their paper "SecuBat-A web vulnerability scanner" [25] discusses that many web application security vulnerabilities result from generic input validation problems. Examples of such vulnerabilities are SQL Injection and Cross-Site Scripting (XSS). The main contribution of this paper is to show how easy it is for attackers to automatically discover and exploit application-level vulnerabilities in a large number of web applications. A detailed study of sql injection and cross site scripting is done and analyzed the possibilities. To this end, they presented SecuBat, a generic and modular web vulnerability scanner that analyzes web sites for exploitable SQL and XSS vulnerabilities. Using SecuBat identified a large number of potentially vulnerable web sites. Moreover, they selected one hundred of these web sites for further analysis and manually confirmed exploitable flaws in the identified web pages. The research findings revealed the lacking security in the web applications of well-known global companies, computer security organizations, and governmental and educational institutions. Such vulnerabilities, for example, could be used

to launch phishing attacks that are difficult to identify even by technically more sophisticated users. The SecuBat also has a crawling component to determine the doors of attack and attack plugins are used to detect them.

This paper is surely an asset as its source is available and increases the confidence of web site administrators and web developers to proactively audit the security of their applications.

Acunetix web vulnerability scanner is a commercial tool available to assess the vulnerability of applications.

I. Aspect Oriented Approach

Security is a cross cutting concern of web applications rather than a functionality. Aspect oriented methodology allows separation of concerns from functionality. Security models can be designed separately and then weaved to web applications using this approach.

Lidia [17] introduces an approach that can be used to weave multiple aspects in to the executable UML model. This work throws light on testing design models.

Using Aspect Oriented Techniques to support separation of concerns in Model Driven Development by Arnor Solberg [2] is an appreciable work that proposes a Model Driven Development framework that adopts Aspect Oriented techniques and discusses the PIM to PSM transformations. Here PIM contains various aspects and base model and transformations are done separately to map to PSM model. The integration of separation of concerns and base model is done at the PSM level. Aspect weaving is not discussed at PIM level.

J. Model Driven Web Application Development Approach

Model driven development approaches enables secure web application design and code generation. There are lots of significant works done in this area.

Nora Koch and Andreas Kraus in their paper "The Expressive power of UML-based Web Engineering" [18] proposes a methodology for web application development. It uses UML 2.0 as the base modeling language and defines stereotypes for modeling the domain specific aspects. The lightweight extension proposed by UWE is very simple. To represent web applications in a better way UWE defines navigation diagrams and presentation diagrams which seem to fulfill the purpose. It didn't use any additional notations for extensibility even though the work has proved the efficiency of UML to express the domain aspects. They have given a very general approach to web application development and security issues except access control have not been taken into consideration.

N.Koch in another paper "Classification of model transformation techniques used in UML-based web engineering" [19] illustrates the UWE transformation approach as a systematic, semiautomatic and tool supported methodology. In this approach method generation of classes is done automatically based on objects of the activity diagrams. But requirement to architecture model is done manually. Also a semiautomatic transformation is carried out from requirements to process and from process to navigation. UWE uses QVT (Query-View-Transformation) language and provide facility for XML metadata interchange and model transformations are defined at metamodel level.

N.Moreno, P.Fraternali and A.Vallecillo in their paper "WebML modeling in UML" [23] proposes a more flexible and elaborated web application modeling language. The expressive power of WebML is appreciable but it deviates from the pure UML notations. Hence transforming WebML model to standard MOF (Meta Object Facility) metamodel is not a very easy task. WebML proposes a UML2.0 profile in order to do the mapping. As the current research direction is towards model driven development and latest specifications UML2.0 and MOF 2.0 have evolved, the modeling language which uses the recent facilities is needed.

"Incorporating security behavior into business models using a model driven approach" [22] by Peter Linington and Pulitha Liyaanagama brings together MDE and aspect in an effort to incorporate security into business models. Here they have mentioned secure client server communication and authentication. Other security aspects are not been taken into consideration and yet to be covered. They have proposed a graphical model merging process where each concern is modeled separately and weaved together. The merging transformation is very complex. This approach also has some implementation issues as it sticks to basic Eclipse modeling framework which does not have an efficient representation of behavior.

Manuel Koch and Karl Pauls deals with access control concerns in their paper "Generation of role based access control requirements from UML diagrams". The approach is based on the requirement and analysis phase diagrams of UML and is called VBAC (View based Access Control).

In the work 'Model Driven Security for process oriented systems' by David Basin and Jurgen Doser[5] have shown how model driven paradigm can be adapted to introduce security. An effort has been made to integrate designing language and security language. As the work is an extension of UML based design model, transformations of design model to architecture model is feasible. Still the work does not address the issue of implementation platform. Also the security languages could be evolved from time to time and the models need to be redesigned. In their paper 'Model driven security from UML models to access control infrastructure' discusses the new approach to build secure systems using model driven security. SecureUML is used to specify the access control policies and provides semantics for verifying transformation.

III. Conclusion

Even though lots of secure software engineering practices exists, web applications still face severe attacks. Most of the methods follow signature based filtering than behavior based filtering which can be bypassed easily. Also attackers find ways to exploit the new technologies and countermeasures. The various approaches discussed in this paper deals with various levels of defense.The combination of various methods and approaches discussed above would aid in protecting them to certain extent.

ACKNOWLEDGMENT

I thank all those who have reviewed and made suggestions for improvement.

REFERENCES

[1] Ahsan Habib, Mohamed M. Hefeeda, and Bharat K. Bhargava, "Detecting Service Violations and DoS Attacks", National Science Foundation,pp1-13
[2] Arnor Solberg , "Using Aspect Oriented Techniques to support separation of concerns in Model Driven Development", IEEE 29th Annual international computer software and applications conference(COMPSAC'05)
[3] Application security-An essential part of your risk management program-IBM whitepaper 2005,pp1-2
[4] Bobbitt M.Bulletproof web security. Network security magazine. Techtarget storage media may 2002,pp 1-10
[5] David Basin, Jurgen Doser, "Model driven security for process oriented systems. SACMAT03 ACM conference Cotno, Italy
[6] David Larochelle and David Evans. "Statically Detecting Likely Buffer Overflow Vulnerabilities" In 2001 USENIX Security Symposium, Washington, D. C., August 2001
[7] David Scott, Sharp R Abstracting Application level web security. 11th international conference on WWW.
[8] Filippo Ricca, Massimiliano Di Penta, Marco Torchiano,, Paolo Tonella, Mariano Ceccato , The Role of Experience and Ability in Comprehension Tasks supported by UML Stereotypes , Softwae Engineering, 2007 , ICSE 2007 May 29 th International Conference, Pages 375-384.
[9] Gregor Kiczales, Erick Hilsdale, An overview of AspectJ
[10] John Wilander,Marian Kamkar.A comparison of publicly available tools for dynamic buffer overflows prevention. N/w and distributed System security symposium conference proceedings 2003, pp3-10
[11] Joshi J,W. Ghafoor, A Stafford.E. "Security models for web based applications" Communications of ACM Feb 2001
[12] Jin-Cherng Lin and Jan-Min Chen. "An automatic Defence Mechanism for malicious injection attack."
[13] J.Juerjens. UMLsec: Extending UML for Secure Systems Development. In Proc. Of 5th Int. Conf. on the Unified Modeling Language, Lect. Notes in Comp. Sci. 2460 pages 412-425, Springer, 2002.
[14] J Jurjens UMLSec: Extending UML for secure systems development
[15] Kevin Heineman, SPI Dynamics: "Complete web Application security :Phase I building web Application security into your development process. SPI Dynamics whitepaper 2002
[16] Leslie Lamport. Password authentication with insecure communication. Communications of the ACM,Nov 1981
[17] Lidia Fuentes, Pablo Sanchez , Designing and Weaving Aspect-Oriented Executable UML models, Journal Of Object Technology August 2007.
[18] Nora Koch and Andreas Kraus , The Expressive Power of UML-based Web Engineering, 2nd Int. Workshop on Web-oriented Software Technology(IWWOST02), Malaga, Spain, June 2002.
[19] N.Koch, Classification of model transformation techniques used in UML based web engineering, IET Softw,2007,pp 98-111
[20] Omar Ismail and Masahi Etoh, "A proposal and implementation of Automatic detection/collection system for Cross site scripting vulnerability" .
[21] P.Fraternali, P.C Lanzi Exploiting conceptual modeling for web application quality evaluation. 13th International conference on World wide Web May 2004
[22] Peter F. Linington and Pulitha Liyanagama. Incorporating Security Behavior into Business Models using a Model Driven Approach. 11th IEEE International Enterprise Distributed Object Computing Conference(2007)
[23] P. Fraternali, N. Moreno and A. Vallecillo. WebML modielling in UML, IET Software, 2007 pp 67-80
[24] Secure Software Development by Example. IEEE security & privacy, July 2005
[25] Stephan Kais, Engin Kirda "SecuBat-A web vulnerability scanner". 15th international conference World Wide Web May 2006,pp248-253
[26] Steven M Bellovin, Michael Merrit. "Encrypted key exchange:Password based protocols secure against dictionary attacks". IEEE symposium on security and privacy 1992
[27] Symantec whitepaper, Internet security threat report trends for july 05-06
[28] Web Application security consortium.Threat classification
[29] William G.J. Halfond, Alessandro Orso, WASP: Protecting Web applications Using Positive Tainting And Syntax-Aware Evaluation. IEEE Transactions On Software Engineering, Vol. 34, No. 1, January/February 2008, pp65-81
[30] Xiang Fu, Xin Lu, Boris Peltsverger, Shijun Chen , "A Static Analysis Framework For Detecting SQL Injection Vulnerabilities", 31st Annual International Computer Software and Applications Conference(COMPSAC 2007)
[31] Yao-wen Huang, Shih-kun Huang, Tsung-Po Lin, Chung-Hung Tsai
[32] Web application security assessment by fault injection and behaviour monitoring.12th International World Wide Web Conference-ACM Press,pp 149-156
[33] Yao-wen Huang, Fang Yu. Securing web application code by static analysis and runtime protection. 13th International conference on World wide web May 2004-ACM press,pp41-48

Dr.Dhanya Pramod is a senior member of IACSIT, Singapore. She has done her post graduation in Computer Science from Pondicherry central university and Ph.D in Computer Applications from Symbiosis International University. She has a strong academic foundation and was the First Rank holder of university both at undergraduate and post graduate level.
She has over 10 years of experience including industry, research, academics and administration. Her research interest is networks & application security and aspect oriented programming. She has published papers in refereed journals and several conferences of international repute.