Web Development Evolution: The Assimilation of Web Engineering Security

William Bradley Glisson
Department of Computing Science, The University of Glasgow, Scotland
glisson@dcs.gla.ac.uk

Professor Ray Welland
Department of Computing Science, The University of Glasgow, Scotland
ray@dcs.gla.ac.uk

Abstract

In today's e-commerce environment, information is an incredibly valuable asset. Surveys indicate that companies are suffering staggering financial losses due to web security issues. Analyzing the underlying causes of these security breaches shows that a significant proportion of them are caused by straightforward design errors in systems and not by failures in security mechanisms. There is significant research into security mechanisms but there is little research into the integration of these into software design processes, even those processes specifically designed for Web Engineering. Security should be designed into the application development process upfront through an independent flexible methodology that contains customizable components.

1. Introduction

Current web applications face major security problems because security design is not integrated into the Web Engineering Development Process. This security integration into the software design process should take place through the implementation of an independent flexible Web Engineering Security (WES) methodology. The purpose of the methodology is to integrate with an organization's existing development process while providing the necessary structure to create and implement secure applications.

Failure to fully integrate the security design into the application design process creates an environment that is conducive to cyber security breaches. Exploitation of these breaches translates into staggering corporate financial losses. The 2004 CSI/FBI Computer Crime and Security Survey and the PricewaterhouseCoopers information security breaches survey estimates losses from internet security breaches to be in the millions of dollars within the last year.[1, 2]

Security has become a major requirement in the web application development environment. The Organization for Internet Safety (OIS) publishes Guidelines for Security Vulnerabilities Reporting and Response. In this document they define a security vulnerability as "a flaw within a software system that can cause it to work contrary to its documented design and could be exploited to cause the system to violate its documented security policy".[3] Hence, any flaws in the system design or application coding can potentially lead to security vulnerabilities. Common security problems include un-validated parameters, cross-site scripting, buffer overflows, command injection flaws, error-handling problems, insecure use of cryptography, and broken Access Controls.[4, 5]

It can no longer be assumed that security will be addressed in the acquisition of the functional or non-functional requirements. These surveys indicate that there are fundamental security problems with the methodologies being used in real world web application development. Security should be designed into the development process upfront and revisited throughout the entire development process.[6]

This brings up the next logical question – what is a secure application? "A secure application is well designed, well managed, well reviewed" from an application development standpoint, and "well maintained" throughout the product life cycle.[7]

So how do we encourage security of the application through a methodology? This paper proposes a solution to the current security development issue through the use of the Web Engineering Security (WES) methodology development process.

2. Web Engineering Security Methodology

The WES methodology, as shown in figure 1, starts with a Project Development Risk Assessment. This initial phase examines the security risk associated with the implementation of a project. The Application Security Requirements phase examines the requirements from the customer perspective within the frame work of organization compatibility. Security Design / Coding examines the architecture, the solution design and the coding practices that are implemented to solve the issue.

0-7695-2471-0/05 $20.00 © 2005 IEEE

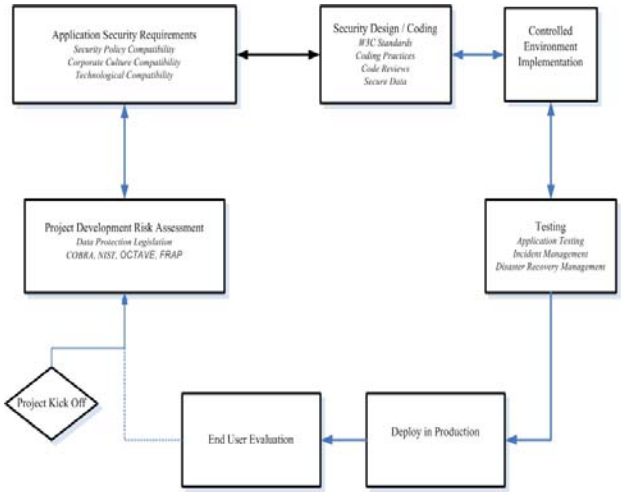
Application Security Requirements
Security Policy Compatibility
Corporate Culture Compatibility
Technological Compatibility
Security Design / Coding
W3C Standards
Coding Practices
Code Reviews
Secure Data
Controlled Environment Implementation
Project Development Risk Assessment
Data Protection Legislation
COBRA, NIST, OCTAVE, FRAP
Testing
Application Testing
Incident Management
Disaster Recovery Management
Project Kick Off
End User Evaluation
Deploy in Production

Figure 1. WES Methodology

A Controlled Environment Implementation can be as simple as implementing and running the application on a desktop to test the security controls within the application. It can also be as complex as implementing the application on a separate server that mirrors production. The point here is to release the code in a secure environment that simulates the production environment for compatibility testing before the application is made available to the general public.

Testing is critical to the success of any application. This hypothesis holds true in the area of security as well. Testing not only includes the examination of code but incident management and disaster recovery. Deployment of the application in a production environment should only take place after it has successfully completed testing.

End User Evaluation involves both security maintenance and communicating with the user to determine the success of the application's security. This can range from informal communication, to surveys, and structured interviews with the end user. Security has to find a balance between usability and providing a secure environment. If users are circumventing security to do their jobs or make life easier then these reasons need to be investigated and resolved in some manner. Security maintenance has to do with discovering vulnerabilities after a production release. As new technologies emerge from the view point of development and maintenance, new vulnerabilities will be created and uncovered and these issues will have to be addressed to maintain application security. [8, 9]

WES was designed to complement software development through customer communications, short development cycles, and practical security solutions to business problems.[10] WES attempts to achieve this by stressing core principles while providing a general outline with customizable sub-components. The core principles behind the development of WES include good communication, employee education, and cultural support. Good communication is a critical component of the methodology, as it is needed to assure solution compatibility within the development team and within the organization. Communication with the end user is needed to acquire the appropriate application requirements. Employees need to be educated on the importance of security and the potential impact on the organization.

This education should include developing awareness of various types of technical attacks and social engineering attacks.[11] Security needs to be viewed in the application development process as "everybody's problem".[12] Separating security responsibilities from the development process has the potential to send the signal to the development group that security is someone else's problem. Security for the purposes of this discussion is defined in terms of confidentiality, integrity and availability.

Confidentiality ensures that information availability is limited to the appropriate individuals.[8] Integrity concerns data integrity meaning that information can only be modified by appropriate individuals.[8] Availability means that information is accessible by appropriate individuals.[8, 13] Cultural support for security should embrace confidentiality, integrity and availability throughout the management structure. The WES methodology aims to provide a roadmap for web application development that will help guide organizations to a more secure system. The goal is to help developers create applications that are secure by design.

The general categories are not set in stone but are strongly recommended. The items within the categories will need to be tailored to the specific needs of the individual organization and their current policies and procedures. The methodology is designed to complement an organization's current methodology, while providing guidance to the development process from a security perspective. Project Development Risk Assessment, Application Security Requirements, Security Design / Coding and Testing, from an organizational integration perspective, warrant a more in-depth discussion.

3. Project Development Risk Assessment

The purpose of the risk assessment is to identify any risk associated with the development of the proposed application functionality. This would include examining appropriate data protection legislation that might apply to your organization's application. There are several tools and suggested practices available in the market for conducting risk analysis. These tools include COBRA, the Facilitated Risk Analysis Process (FRAP) and the

Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE).[14, 15] The National Institute of Standards and Technology (NIST) has recommendations for conducting company wide risk analysis on their web site.[16] OCTAVE is an in-depth organization wide risk analysis approach developed at Carnegie Mellon.[17]

If an organization wide risk analysis is conducted periodically, then the information in the analysis can be used as a starting point for the application risk analysis. The reverse is also true. Information from the individual application analysis can be used as an initial guide to organizational analysis. The risk assessment piece of the methodology can be customized to work in conjunction with an organization's existing risk analysis processes. The basic idea is to detail critical functions, determine the necessary service levels in doing so, identify possible threats and outline their motivating factors, estimate the probability of an attack, estimate the probability of a successful attack, and outline the cost of providing protection.[18] [19] [8]

Application threats can cover a wide range of possibilities including: human errors in coding, user errors, external attack, fraudulent individuals, technical sabotage, acts of God, and disgruntled employees; all of which should be accounted for in the risk assessment.[18]

Once the risk assessment has taken place, the specific application security requirements need to be determined through in-depth conversations with the end user and evaluation of organizational compatibility. Organizational compatibility determines how well security requirements fit into the frame work of an organization. The general areas that make up this category include security policy compatibility, corporate culture compatibility and technical compatibility.

4. Application Security Requirements

The Security policy encompasses all business interactions providing overall guidance to protecting resources.[6] This includes acceptable computing practices, all interactions with the network, internet, messaging, and business specific applications or services.[18] Companies may need to meet security policy standards requirements like the ones put out by the International Standards Organization (ISO).[20] In the context of web development, the main area of concentration, with regards to the security policy, would be application compatibility within the corporation. However, all areas would need to be addressed to ensure overall compatibility. The security policy should be a living document and updated as new architectures and applications are developed.[21] If a security policy does not exist at project inception, then the organization may need to investigate the validity of creating the appropriate document.

Corporate culture needs to be examined from several different perspectives that include managerial acceptance of the importance of security, the threat of social engineering, employee perception of security and security habits, and technological acceptance of cultural norms. Managerial acceptance and habits are critical to the success of security within an organization. Large organizations, looking to strengthen security in their corporate cultures, need to have the highest possible ranking champion promoting the change. In small organizations the change should be introduced by the owner. If management takes security seriously and encourages a secure environment through their actions, then the odds of this having a positive trickle down effect to employees within the organization are good. If the management does not care about security, why should the employee?

In general, the area of technological compatibility has to do with an organization's existing applications, software compatibility, legacy systems and the acquisition of new software and technology.[22] When considering the technical compatibility of a system, it is necessary to consider the existing employee skill set within the company. To implement a technical solution, does the necessary skill set exist internally, can it be acquired easily through employee training or will it require the company to acquire the necessary skills though outsourcing? To answer these questions an in-depth analysis will need to be conducted and compared with the solutions requirements. Technological compatibility, from a security standpoint, needs to examine the application to see if it is compatible with existing security solutions already in production. An example would be a new application that is not compatible with the company's existing single sign-on solution. If a solution requires new technologies, the organization should rate the security capabilities of the new technologies and determine if they meet the company's security standards out of the box. If they do not, can they be brought up to speed and at what cost?

This does not mean that these are the only areas that can contribute to this category or that they all have to be present within this section to ensure compatibility. There are environments that may choose not to implement a security policy or to investigate corporate culture due to the size of the company. For instance, a large financial institution will probably have all three categories (security policy compatibility, corporate culture compatibility and technical compatibility) documented to some extent. However, a small family run business, like a local restaurant, probably will not have a security policy and the culture in that business will be implicit. However, more than likely, they will have technical compatibility issues that they will need to address.

5. Security Design / Coding

Once the application security requirements have been determined, the next issue that needs to be addressed is security design. The design of the application needs to consider the overall architecture, the application design, and good design principles.

The architecture needs to fit into the existing organizational environment. There are several issues that need to be addressed within the realm of architecture. Some of those issues include the application layers[23] and maintainability [12], information compatibility, how strongly typed the language needs to be [9], approach to privileges from the application and the user's standpoint[8], and security in-depth.[8]

The design of the application needs to address the type of language that will be used[9], the ease of use[8], authorization techniques, and the use of encryption algorithms. The design will need to examine the code from a common attacks standpoint and implement the appropriate controls to ensure secure data. It should be noted that a professional code management system needs to be used by the development team to ensure accountability, within the team, and provide a means of roll back.[7]

Once the design has been chosen, the solution is coded. During coding, the developer should be cognizant of the World Wide Web Consortium (W3C) coding standards and pursue secure coding practices.[24] The trick, when designing a secure solution, is to balance the need for a secure application with the need for a particular functionality.

A good design principle is to create a simple solution that solves specific problems and fits into the applications global architecture. The design will depend on the level of security the customer is willing to accept from a risk and/or cost standpoint.

6. Testing

Developers need to examine their code independently and the program as a complete entity in order to determine possible misuse from a functional standpoint. That is, programs only do exactly what they are designed to accomplish. "Vulnerabilities can stem from the rapidly evolving use of software, in which programs meant for a limited purpose are applied in ways not anticipated by their developers."[25] A primary example is an e-mail server that is used to propagate a virus or used in a denial of service attack.

Testing is critical to the success of any application. Testing should cover application testing, incident management and disaster recovery plans. Application testing includes validation errors, program behavior testing, and code analysis. This will involve implementing appropriate programs to test static and runtime code, penetration, and application scanning. Automation, where possible, of the testing process will help provide stability. Testing should also involve executing scripts from both the developer and the end users to test the application. An important part of the testing phase should be to decide appropriate action plans for different incidents. When there is an issue, what are the procedures that need to be implemented to resolve the situation? This should also include amending the disaster recovery plan where appropriate. If the organization does not have a disaster recovery plan then maybe they should investigate the creation of a plan**.** The disaster recovery plan on the organizational level should be a living document. The disaster recovery plan for the application should be flexible enough to allow for the addition of a new functionality. Once the plan either has been created or amended then it should be tested.

7. Relevant Work

Several organizations have recognized the importance of security in the development life cycle. Their solutions range from process plug-ins, to modified system development life cycles, to the application of security patterns.

Secure Software has attempted to address this problem recently with the introduction of the Comprehensive Lightweight Application Security Process (CLASP) as a standalone process and a plug-in to RUP.[26] Microsoft has also attempted to address the security issue through their security development lifecycle which is discussed in "Trustworthy Computing Security Development Lifecycle".[9] Another proposed solution has been to apply security patterns through the use of a secure software lifecycle as discussed in "A methodology for secure software design".[27].

Ellis and Speed propose a process for developing a security project in their book.[18] Their solution treats the security aspect of a project as a project in itself.

8. Conclusion

Technical solutions alone will not solve current security issues in the global web environment. Security deficiencies in web enabled business environments are forcing organizations to acknowledge security as a primary business objective. In doing so, organizations will be forced to address application security from a development perspective. Security is an imperfect art form that is an increasingly moving target in today's society. The most effective way to handle security, in the application design, is to incorporate security upfront into the development methodology.

This can be accomplished through the use of an independent flexible web engineering security methodology with customizable components. The tools and processes used by individual organizations can be incorporated and customized to meet specific needs. The methodology has to be built on the principles of good communication, employee education, and cultural support to effectively mitigate future security issues.

Future research will include a closer examination of the business case for implementing the WES methodology. It will also include an in-depth analysis and integration into both traditional and agile development methodologies. The goal is to test, through proof of concept, the WES methodology and evolve WES, where necessary, into a methodology that is compatible with existing Web application development processes so that they are complementary and, yet, flexible enough to be tailored to individual companies. The research will require close collaboration with industry to test the methodology in the 'real world'.

References

[1] Gordon, Lawrence A, Martin P. Loeb, William Lucyshyn, and Robert Richardson, *2004 CSI/FBI Computer Crime Security Survey*. c2004, Computer Security Institute: Computer Security Institute. p. 2-18. http://www.gocsi.com/.

[2] PricewaterhouseCoopers, *The Information Security Breaches Survey 2004*. c2004, PricewaterhouseCoopers. p. 1-36.http://www.pwc.com/images/gx/eng/about/svcs/grms/2004Technical_Report.pdf.

[3] Organization for Internet Safety, Guidelines for Security Vulnerability Reporting and Response. http://www.oisafety.org/guidelines/Guidelines%20for%20Security%20Vulnerability%20Reporting%20and%20Response%20V2.0.pdf

[4] Mimoso, Michael S., Top Web application security problems identified SearchSecurity.com. April 12, 2005. http://searchsecurity.techtarget.com/originalContent/0,289142,sid14_gci873823,00.html?NewsEL=9.25

[5] Berinato, Scott, *The Bugs Stop Here*, in *CIO*. c2003. http://www.cio.com/archive/051503/bugs.html.

[6] Premkumar T. Devanbu, Stuart Stubblebine. *Software engineering for security: a roadmap*. in *International Conference on Software Engineering Proceedings of the Conference on The Future of Software Engineering*. 2000. Limerick, Ireland: ACM Press New York, NY, USA. http://portal.acm.org/citation.cfm?id=336559.

[7] Foster, James C., Five hidden tactics for secure programming. 25/03. http://event.on24.com/eventRegistration/EventLobbyServlet?target=lobby.jsp&playerwidth=950&playerheight=680&totalwidth=800&align=left&eventid=8449&sessionid=1&partnerref=SitePost&key=21A0DCD96C5F113F45DDD69439B96F46&eventuserid=3328607

[8] Pfleeger, Charles P. Pfleeger and Shari Lawrence, *Security in Computing*. Third Edition ed. c2003, Upper Saddle River, NJ: Prentice Hall.

[9] Lipner, Steve. *The Trustworthy Computing Security Development Lifecycle*. in *2004 Annual Computer Security Applications Conference*. 2004. Tucson, Arizona: Annual Computer Security Applications Conference. http://www.acsac.org/2004/papers/Lipner.pdf.

[10] Beck, Kent, Manifesto for Agile Software Development, The Agile Alliance,. 08/04/2005. http://www.agilealliance.org/

[11] Mitnick, Kevin., *The art of deception : controlling the human element of security / Kevin D. Mitnick & William L. Simon.* c2002., Indianapolis, Ind.: Wiley. 352.

[12] Graff, Mark G, and Kenneth R. van Wyk, *Secure Coding Principles & Practices*, ed. D. Russell. c2003, Sebastopol, CA: O'Reilly & Associates Inc. 1-183.

[13] Galicia e-Commerce Leveraging Centre, Web Engineering Methodology and Development Manual. 06/05/2005. http://www.e-negociogalicia.com/proxecto/documentacion/Web_Engineering_Methodology_and_Development_Manual.pdf

[14] C&A Systems Security Limited, COBRA. 25/04/2005. http://www.riskworld.net/

[15] Educause, Risk Analysis of Critical Areas and Processes. 25/04/2005. http://www.educause.edu/content.asp?page_id=1251&bhcp=1

[16] Peltier, Thomas. *Effective Risk Analysis*. in *23rd National Information Systems Security Conference*. c2000. Baltimore, Maryland: National Information Systems Security Confrence. http://csrc.nist.gov/nissc/2000/proceedings/papers/304.pdf.

[17] Alberts, Christopher, *Introduction to the OCTAVE® Approach*, J.S. Audrey Dorofee, Carol Woody, Editor. c2003, Carnegie Mellon Software Engineering Institute: Carnegie Mellon Software Engineering Institute. p. 1-37. http://www.cert.org/octave/approach_intro.pdf.

[18] Ellis, Juanita , and Timothy Speed, *The internet security guidebook : from planning to deployment / Juanita Ellis, Tim Speed ; developmental editor Ed Carrasco.* c2001., San Diego: Academic Press. 1-320.

[19] Phaltankar, Kaustubh M., *Practical Guide for Implementing Secure Intranets and Extranets*. c2000, Boston: Artech House, Inc. 121-182.

[20] ISO, International Organization for Standards. 06/05/2005. http://www.iso.org/iso/en/ISOOnline.frontpage

[21] Symantec, Importance of Corporate Security Policy Defining corporate security policies, basing them on industry standards, measuring compliance, and outsourced services are keys to successful policy management. 07/15/2005. http://securityresponse.symantec.com/avcenter/security/Content/security.articles/corp.security.policy.html

[22] Boman, Magnus, *Conceptual Modelling*. c1997, London: Prentice Hall. 269.

[23] Fernandez, E.B. *Coordination of security levels for Internet architectures*. in *Procs. 10th Intl. Workshop on Database and Expert Systems Applications*. c1999: Procs. 10th Intl. Workshop on Database and Expert Systems Applications. http://polaris.cse.fau.edu/~ed/Coordinationsecurity4.pdf.

[24] W3C, W3C World Wide Web Consortium. 24/04/2005. http://www.w3.org/

[25] Williams, Phil, Timothy Shimeall , and Casey Dunlevy, *Intelligence Analysis for Internet Security*, T.S. Casey Dunlevy, Editor. c2002, Routledge, part of the Taylor & Francis Group: London • New York • Oslo • Philadelphia • Singapore • Stockholm. p. 1-38. http://taylorandfrancis.metapress.com/link.asp?id=8epqf772b2jbpby3.

[26] Viega, Jon, Security in the software development lifecycle. 07/04/2005. http://www-128.ibm.com/developerworks/rational/library/content/RationalEdge/oct04/viega/

[27] Fernandez, E.B. *A methodology for secure software design*. in *Procs. of the 2004 Intl. Symposium on Web Services and Applications (ISWS'04)*. c2004. Las Vegas, NV. http://polaris.cse.fau.edu/~ed/EFLVSecSysDes1.pdf.