Evaluation of web application security risks and secure design patterns

Conference Paper · January 2011

DOI: 10.1145/1947940.1948057 · Source: DBLP

2 authors:

Asish Dalai
National Institute of Technology Rourkela

Sanjay Kumar Jena
National Institute of Technology Rourkela

Evaluation of Web Application Security Risks and Secure Design Patterns

Asish Kumar Dalai and Sanjay Kumar Jena
National Institute of Technology Rourkela, Odisha, India
dalai.asish@gmail.com, skjena@nitrkl.ac.in

ABSTRACT

The application of security in web application is of profound importance due to the extended use of web for business. Most of the attacks, are either because the developers are not considering security as a concern or due to the security flaws in designing and developing the applications. The enforcement of security in the software development life cycle of the application may reduce the high cost and efforts associated with implementing security at a later stage. For this purpose, various attempts has been made to define some security patterns keeping the attacks in mind. The developers now can use these patterns but sometimes it is difficult to choose a pattern from the large list, which may or may not suit the context. This paper is based on analyzing the existing security patterns. Here web application vulnerabilities has been classified and pairing is done between each vulnerability and a suitable pattern.

Categories and Subject Descriptors

D.2.11 [**Software Engineering**]: Software Architecture—*Patterns*; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*Authentication, Unauthorized access.*

General Terms

Security, Design.

Keywords

Web Application Security Risks, Design Patterns, Security Patterns

1. INTRODUCTION

Developers have chosen the web as the prime choice for designing and deploying the applications due to its cross platform compatibility. Web application have undergone several advancements with latest technologies: Ajax and web services. The attacks to the web applications are also increasing at a greater rate along with advancement in technology. Hence, the traditional approach of software development life cycle may not suit well where security in the web application comes into picture. Writing secure code and/or testing the code to make the application attack proof is a challenging task. Using context specific security design patterns for web application may reduce the chances of attack. Attempts have been made to define some security patterns which can be used by the developers. Choosing the appropriate one for the context is difficult.

In this paper, the related works in security patterns has been studied and then the most common vulnerabilities in the web are categorized with an attempt to provide a suitable security pattern for each vulnerability. This then leads to a number of problems and indications for future research directions for web application security patterns. This paper is structured as follows: Section 2 describes the origin and growth of design patterns. Section 3 contains the security patterns representing the related works. In Section 4 the most common web vulnerabilities have been classified and suitable patterns are associated with them. Finally the concluding remarks are given in the last section.

2. DESIGN PATTERNS

Design patterns are reusable solution to commonly occurring problems in design. Design pattern in the field of software development, provides experts' knowledge and experience in form of a design template.These templates are implemented in software development life cycle to avoid the recurrence of specific issues in software applications.

The idea of design pattern was actually started with civil architecture for modeling buildings and towns. Later it is used in software engineering in the design of the software applications. The experience and knowledge of the developers during the course of development has been captured and modeled as a solution to specific problem named as a design pattern. The developers in future can use these patterns which will reduce their task for developing applications.The same idea of design pattern when defined to solve the security problems in the software applications called as security design patterns. The use of these security patterns then resolves the security issues in the applications.

The origin and journey of the design patterns from building architecture to the field of software engineering and then to the area of security are summarized as below:

ICCCS'11 February 12-14, 2011, Rourkela, Odisha, India

Table 1: Pattern Template [4]

Element	Description
Name	Name of the pattern
Also Known As	Alias or Other names of the pattern if any are known
Example	A real-world example demonstrating the existence of the problem and the need for the pattern.
Motivation	A description of situations in which the pattern may apply.
Applicability	A general description of the characteristics a program must have for the pattern to be useful in the design or implementation of the program.
Structure	A textual or graphical description of the relationship between the various participants in the pattern.
Participants	The entities involved in the pattern.
Consequences	The benefits the pattern provides and any potential liabilities.
Implementation	Guidelines for implementing the pattern.
Sample Code	Code providing an example of how to implement the pattern.
Example Resolved	An example of how the real-world example problem described in the Example section may be resolved through the use of the secure design pattern.
Known Uses	Examples of the use of the pattern, taken from existing systems.

- 1977-79 - Architect Christopher Alexander introduced the concept of design patterns with respect to design of buildings and towns.
- 1987 - Beck and Cunningham experimented with applying patterns to programming and presented their work at OOPSLA.
- 1994/95 - The "Gang of Four" (Erich Gamma, Richard Helm, Ralph Johnson, and John M. Vlissides) published a book containing a large number of design-level patterns aimed at object oriented programming languages [7].
- 1997 - Yoder and Baraclow published a paper outlining several security patterns [19].

A pattern can be characterized as *a solution to a problem that raised within a specific context* [17]. A pattern not only describes the solution but also explains the context and problem for which it is used. Thus, it is defined as {problem/context/solution} triplet. Patterns can be illustrated using a class diagram or a role model.Gamma et al. proposed a template [7] which includes 1. name of the pattern, 2. intent/motivation, 3. applicability structure and 4. related patterns. Later in 2006, Buschmann et al. have proposed an extension to this template [17].A standard template to represent the design pattern as proposed by Dougherty et al. [4] is shown in Table 1. Following a standard template facilitates understanding the important concepts and allowing the developers with security background to easily apply the pattern in their own context.

3. SECURITY PATTERNS

Security design pattern implements the experts' knowledge and experience in the form of proven solutions to recurring security problems. Generally security is disregarded due to lack of security aspect in the life cycle. Only the threat analysis in the viewpoint of an attacker reveals the vulnerabilities in the application and identifying the threats in the later stage requires a great deal of effort. So the presence of an efficient security design pattern enables to bridge the gap between developer and security experts thereby reducing the vulnerabilities. Security patterns try to provide constructive assistance in the form of worked solutions and the guidance to apply them properly.A significant amount of research has already been performed in the field of security patterns. This section provides some of the major contributions to this field and presents a brief description of each piece of work in Table 2.

Developers can also follow catalogue consisting set of design and implementation guidelines emphasizing programmers viewpoint for writing secure programs. These guidelines have been pragmatically collected from actual programming experiences.

Researchers have designed set of patterns to meet security requirements of the application, but the growing risks in the web and the new threats has put a challenge and has given a new dimension to research in security patterns. In the next section a classification of the common vulnerabilities has been made and suitable patterns are associated with each vulnerability.

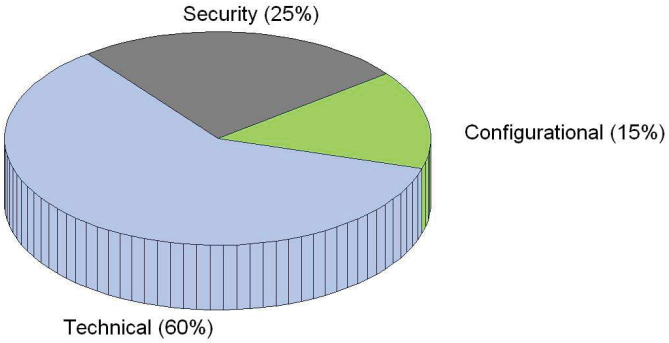

Figure 1: Breakdown of Vulnerabilities in Web Application

4. WEB APPLICATION SECURITY RISKS

The web server and the browser perform so many tasks for rendering the page on the user's screen. With the advent of web 2.0, users got a rich experience but at the same time it has opened many doors for the hackers to seep through and gain unauthorized access to user and business sensitive

Table 2: Related Works on Security Patterns

Author	Year	Name of the pattern	Remarks
Yoder and Barcalow [19]	1997	Architectural patterns for enabling application security	A group of 7 Patterns provides security framework for building application.
Braga et al. [2]	1998	Cryptographic Design Pattern	Consisting a group of 9 patterns provides four fundamental objectives of cryptography and a pattern language for cryptographic software.
Brown et al. [3]	1999	The Authenticator Pattern	Performs authentication of a requesting process before deciding access to distributed objects .
Sasha Romanosky [15]	2001	Security Design patterns Part 1	A set of 8 patterns providing architectural and procedural guidelines.
Fernandez and Pan [5]	2001	A Pattern Language for Security Model	The authorization, Role Bases Access Control, Multilevel Security and file authorization patterns are cataloged.
Kienzle and Elder [11]	2002	Security Patterns for Web Application Development	A group of 29 patterns categorized as structural and procedural pattern.
J Juren [10]	2004	secure data transfer pattern	For Transferring data securely
Blakely and Heath [1]	2004	Open Group Guide to Security Patterns	Contains architectural level patterns and design-level patterns focusing on system availability and the protection of privileged resources.
Halkidis et al. [9]	2004	Choked Point System pattern	That state can be recovered and restored to a known valid state in case a component fails.
T. Priebe et al. [14]	2004	A pattern System for access control	Provides access control.
Steel et al.[18]	2005	Secure logger pattern and Secure pipe pattern	Application events and data must be securely logged and avoids man in the middle attack.
M Hafiz [8]	2005	secure pre-forking pattern	Task can be forked securely and efficiently in a multitasking environment.
Morrison and Fernandez [12]	2006	The Credentials Pattern	Provides authentication and authorization.
Schumacher et al. [17]	2006	Single Access Point Pattern	System with external access should have a single access point.
Schumacher et al. [17]	2006	Firewall patterns and Accounting patterns	To mitigate the attacks and satisfy confidentiality of services also ensures the availability.
Sasha Romanosky [16]	2006	Privacy Patterns for online Interaction	Provides secure online transactions.
E B Fernandez et al. [6]	2007	Secure pattern for VOIP Network	Guarantee the integrity of calls.
Daniela et al.	2009	An Access Control Pattern	Based on qualifications to grand access to physic resources

data. Security experts has focused on the most common vulnerabilities. Open Web Application security Risk (OWASP) has published "OWASP Top 10" [13] containing 10 most security risks. Many attempts have been made to list the vulnerabilities and their countermeasures. Here most common vulnerabilities has been taken and classified in to three major categories:

1. Technical Vulnerabilities
2. Configurational Vulnerabilities
3. Security Vulnerabilities

The figure 1 shows their impact on the world wide web.

4.1 Technical Vulnerabilities

Vulnerabilities due to technical flaws in the code comes under this category. These vulnerabilities causes most serious attacks like: Injection attacks, XSS and CSRF attacks etc. Based on the attack caused by the vulnerability and intent of the patterns a suitable pattern is associated with each vulnerability as given in Table 3 .

Table 3: Patterns for Technical Vulnerabilities

Technical Vulnerabilities	Security Pattern
Content Injection	Input Validation
cross site scripting flaw	Input Validation and Authentication
cross site request forgery	Secure GetPost
buffer over flow	Input Validation
insecure direct object reference	Access Control Pattern

4.2 Configurational Vulnerabilities

The vulnerabilities due to architectural flaws in the system are categorized as configurational vulnerability. Some time the security goals are not documented properly so there remains some server configuration flaws and leads to various risks. Table 4 contains the configurationl vulnerabilities and one suitable pattern for each of them.

Table 4: Patterns for Configurational Vulnerabilities

Configurational vulnerabilities	Security Pattern
encryption flaw	Cryptographic Design Pattern
broken authentication and session management	Authenticted session
path traversal, directory browsing	Secure Directory
server misconfiguration and predictable pages	document the server configuration
Information Leakage	Encrypted storage
Insufficient Authentication	Authenticator
Disclosing error details	Limited View
code decompilation, source code discloser	code obfuscation
unvalidated redirect and forwards	directed session
failure to restrict url access	Limited View

4.3 Security Vulnerabilities

Sometimes the flaws in transport layer or network layer or in the other layers of the network causes the application to fail. Man in the middle, Denial of service are the examples of such serious attacks. Vulnerability of this category are due to the flaws not in the application but in the system. Table 5 contains the list of such vulnerabilities and suitable patterns for each.

Table 5: Patterns for Security Vulnerabilities

Security vulnerabilities	Security Pattern
Denial of Service	Secure Logger
DNS cache Poisoning	Cryptographic Patterns
Path manipulation	Pathname Canonicalization
man in the middle attack	Secure pipe pattern

5. CONCLUSION

In this paper the related works in web application security risks has been analyzed. A standard template [4] has been provided for the pattern designers to follow. The most common vulnerabilities has been classified into three broad categories. A pairing between each vulnerability and a suitable pattern is done which may help the web application developer. The future research will address the patterns for critical issues like denial of service and cryptography.

Acknowledgment: The authors are indebted to Information Security Education and Awareness (ISEA) Project, MCIT, Department of Information Technology, Govt. of India for sponsoring this research and development activity.

6. REFERENCES

[1] B. Blakely and C. Heath. Security design pattern,tech report g031. OpenGroup, 2004.

[2] A. M. Braga, C. M. F. Rubira, and R. Dahab. Tropyc: A pattern language for cryptographic software. PLoP, 1998.

[3] F. L. Brown and E. B. Fernandez. The authenticator pattern. PLop, 1999.

[4] C. Dougherty, K. Sayre, R. C. Seacord, D. Svoboda, and K. Togashi. *Secure Design Patterns*. Software Engineering Institute, 2009.

[5] E. B. Fernandez and R. Pan. A pattern language for security models. PLop, 2001.

[6] E. B. Fernandez, J. C. Pelae, and M. M. Larrondopetrie. Security pattern for voice over ip network. In *International multi conference on Computing in Global Information Technology(ICCGI'07)*, 2007.

[7] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object Oriented Software*. Addison-Wesley, 1995.

[8] M. Hafiz. Secure pre-forking- a pattern for performance and secuirty. PLoP, 2005.

[9] S. T. Halkidis, A. Chatzigeorgiou, and G. Stephanides. A quantitative evaluation of security patterns. International Conference on Information and Communication Security(ICICS), 2004.

[10] J. Jurens. Secure system development with uml. *Springer*, 2004.

[11] D. M. Kienzle and M. C. Elder. Final technical report: Security pattern for web application development, 2002. `http://www.scrypt.net/celer/securitypatterns/final`.

[12] P. Morrision and E. B. Fernandez. Securing the broken pattern. In *11th European Conference on Pattern Language of Programs (EuroPLoP)*. PLop, 2006.

[13] OWASP. Owasp top 10 application security risks-2010. `http://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project`.

[14] T. priebe, E. Fernandez, J. I. Mehlau, and G. Pernull. A parten system for access control. 18th Annual IFIP WG 11.3 Working Conference on Data and Application Security, 2004.

[15] Romanosky. Security design patterns. technical report. http://www.cgisecurity.com/lib/securityDesignPatterns.pdf, 2001.

[16] S. Romanosky, A. Acquisti, J. Hong, L. F. Carnor, and B. Friedman. Privacy pattern for online interaction. In *PLop 2006 Conference*. PLop, 2006.

[17] M. Schumacher, E. B. Fernandez, D. Hybertson, and F. Buschmann. *Security Patterns: Integrating Security And System Engineering*. John Wiley and Sons Inc, 2006.

[18] C. Steel, R. Nagappan, and R. Lai. *Best Practice and Strategy for J2EE, Web Service and Identity Management*. Prentice Hall, 2005.

[19] J. Yoder and J. Barcalow. Architectural patterns for enabling application security. In *International Conference on Pattern Language of Programs*. PLoP, 1997.