



Technical White Paper

Towards Developing Secure Web Applications

Sathish C V | **Amit Bhalla**

Practice Manager | Consultant
Sonata Software Limited

Sonata Software Limited
www.sonata-software.com

STATEMENT OF CONFIDENTIALITY

Information included in this document, in its entirety, is considered both confidential and proprietary to Sonata Software and may not be copied or disclosed to any other party without its prior written consent.

Abstract

This White Paper discusses how to develop secure Web applications and their importance. It explains the threats and vulnerabilities associated with Web application security, and ways to avert and overcome them.

The paper also gives details of some security models for Web applications and dwells into the best practices for development of secure Web applications.

About the Author(s)

Sathish CV is a Practice Manager at Sonata Software Ltd. With Sonata for over 7 years, Sathish has handled various projects of different complexity levels.

Amit Bhalla is a Consultant with Sonata Software Ltd. In this capacity, Amit has handled various secure Web-enabled projects.

Table of Contents

- 1. Overview1
- 2. Introduction1
- 3. Factors Affecting Web Application Security2
- 4. Threats and Vulnerabilities.....5
- 5. Measures6
- 6. Services offered by Sonata for Web Application Security9
- 7. Sonata’s Value Proposition9
- 8. Conclusion9

1. Overview

This White Paper discusses the various ways to ensure security of Web applications. It also highlights the threats and vulnerabilities involved in developing secure Web applications, and the measures to protect Web applications from unauthorized access.

This White Paper also elucidates Web application security models, authentication and authorization solutions for Web applications, the best practices to develop secure Web applications and the importance of security in moving Web applications to the Software as a Service (SaaS) model.

2. Introduction

The security of any Web application is of paramount importance because of the high value and business criticality of the data shared through it.

As per a report released by PGP -- a global enterprise security company -- in February 2009, the cost of data breach for companies had risen to \$202 per lost record in 2008.

Each Web application is unique in that it has its own sets of vulnerabilities and threats. A compromised Web application not only makes an everlasting dent in the reputation of an organization but also causes loss of credibility of the business model followed.

According to a Gartner survey of 5,000 U.S. adults published in November 2006, in 2006 alone, retailers lost almost \$2 billion because of consumer security fears, with about one-half of those losses (\$913 million) coming from people who avoided sites that seemed to be less secure and the rest (about \$1 billion) came from consumers who were too afraid to conduct e-commerce business at all.

Following is the conclusion of a few analyses of the status of Web application security:

- According to Symantec Internet Security Threat Report dated January-June 2005, 73 percent of Web application vulnerabilities could be exploited easily
- The CSI / FBI Computer Crime Survey, 2005, reported a 90-percent surge in Web attacks in 2005
- Research firm Gartner estimates that by 2010, 80 percent of all companies would have suffered an application security incident

Some Web Hacking Incidents

- In November 2008, a banking major's data breach hit 1.5 million people
- In January 2009, a public sector organization was found serving remote malware through iFrame attack
- An antivirus giant's site was breached using SQL injection, exposing sensitive data. The vulnerability was uncovered in February 2008

- In April 2008 "SQL by Design" leaked thousands of Social Security Numbers (SSNs) on an American state government site
- A leading e-payment service provider suffered Internet payment hacker attack in November 2008
- In March 2009, China-based hackers hacked numerous politically sensitive and high-value computer systems across the world

3. Factors Affecting Web Application Security

The factors affecting the security of Web applications can be classified into three categories:

1. Infrastructure-related factors
2. Implementation-related factors
3. Other factors

3.1. Infrastructure-related factors affecting security

IT infrastructure is the backbone of a Web application. Properly configured and maintained infrastructure is indispensable for a secure Web application. However, certain challenges are involved in maintaining secure infrastructure. Some of these are:

- **Network**

Insecure network with improperly configured firewalls is one of the major threats to the security of a Web application. It not only makes the communication insecure but also exposes vulnerabilities of the underlying system, which can be misused by a hacker to target other independent applications / systems.

- **Application Server**

An application server is vital for hosting a Web application. An improperly configured application server becomes an easy target for hackers, who can hack its exposed / unexposed vulnerabilities.

- **Database**

Database is a crucial component of any information-intensive Web application and is not immune to threats and vulnerabilities. If database accounts and privileges are not properly configured, there is always an opportunity for hackers to exploit the same.

For SaaS applications, database security and integrity become even more critical as the data is hosted at application provider's servers and may be accessed by multiple applications.

Around 62 percent of enterprises responding to a Gartner study released in April 2009 said that they worried about the security of data they sent to destinations outside their firewalls.

Therefore, assiduousness is a prerequisite for prevention of hacking and misuse of data.

- **Client**

Client-side vulnerabilities of a Web application may be caused by the underlying security loopholes of the Web browser. Improper implementation of SSL / encryption for sensitive data may also give rise to client-side vulnerabilities of a Web application.

- **Operating System**

The operating system used for hosting Web applications might have security loopholes because of their improperly configured security and privacy policies, which do not mandate a user to change the default password or change the password after a specified period of time.

With a multitude of diverse technological products for maintaining IT infrastructure -- each with its own security issues -- the synergy of these products becomes a challenging task for implementing a pre-defined security policy for Web applications.

Security of Web applications is also of prime importance while moving them to the SaaS model. As these applications reside with the SaaS vendor, they may expose the business processes and lead to security issues.

Web-based or offsite hosted applications need to adhere to the security, privacy and Internet use policy requirements of the business house. Data backup is also an important consideration for such applications. In SaaS model, data backup is typically offloaded to the SaaS provider, but the business house should take control of its own data.

Organizations must ensure that SaaS usage in their environment(s) is consistent with the policies and controls developed by them for traditional on-premise applications.

3.2. Implementation-related factors influencing security

A vulnerable application may be the outcome of not proposing and following the pre-set guidelines for developing secure Web applications.

In June 2007, the Gartner Group stated that over 70% of cyber attacks occur at the application layer.

Implementation-related factors which may affect the security of Web applications are:

- **Programming Language**

The level of security of a Web application also depends on the programming language chosen for its development. Therefore, it needs to be identified whether the selected

programming language supports the following factors, which aid in developing a secure Web application:

- Secure programming
- Multi-threading
- Extensibility & enhancements
- Timely updates and upgrades
- Protection of sensitive data in case of serialization and de-serialization
- Platform independence
- Multiple concurrent users
- Proper error handling
- Secure data transmission and interoperability

■ **Application Architecture & Development**

Application architecture defines the interdependence between different modules, layers and environments of an application. Various architectural factors that may influence the security of a Web application are:

- Coupling and cohesion
- Communication between different layers as Business layer and Data Access layer
- Incorporation of architectural guidelines
- Usage of security assessment tools for identifying loopholes
- Detection and protection of valuable and sensitive data as applications are built in multi-layer architecture

3.3. Other factors impacting security

Below are the factors which should be considered to determine the intensity of a Web application's security policy:

■ **Usage**

- Internet
- Intranet
- VPN

■ **Information**

- Sensitive
- Non-sensitive

■ **Hosting**

- In-house

- Third-party
- Cloud
- **Content**
 - Banking
 - Financial
 - Literature
 - News

4. Threats and Vulnerabilities

If the factors ensuring the security of a Web application are not implemented accurately, then it may become insecure and prone to various types of attacks.

The most common threats and vulnerabilities of a Web application that are exploited by attackers are:

SQL Injection	<ul style="list-style-type: none"> ■ Exploits the security vulnerability occurring in DB access layer ■ Input is not properly validated
XSS (Cross Site Scripting)	<ul style="list-style-type: none"> ■ XSS takes advantage of dynamically generated Web pages ■ A script activates when it is read by an unsuspecting user's browser
Predictable Resource Location	<ul style="list-style-type: none"> ■ Educated guesses or brute-force search is used for unauthorized Web application contents ■ Replace or update URL with some most commonly used words as /admin, /manager, /logs, /backup, etc.
Malicious File Execution	<ul style="list-style-type: none"> ■ Affected applications are those that fail to prohibit or control execution of uploaded files. Those sites are also affected which allow upload / download of files ■ XML documents submitted by an attacker might have a hostile DTD
Improper Error Handling	<ul style="list-style-type: none"> ■ Detailed internal error messages such as stack traces, database dumps, and error codes are displayed to the user (hacker) ■ These messages reveal implementation details that should never be revealed
In-Flight Data Changes	<ul style="list-style-type: none"> ■ Usually done by ISPs or some malware, add-ons or virus in the client machine
Google Hacking	<ul style="list-style-type: none"> ■ Use of Google's advance search features to search for particular vulnerability in Web applications

Following are the main security concerns for a Web application in SaaS environment:

- Protecting data in transport between the service provider(s) and service consumer.
- Storage of corporate data outside the company.
- Identity and access management, such as integrating the service provider's identity system with that of the consumer.

5. Measures

As says an old adage, 'prevention is better than cure.' So it is always better to build a Web application such that it automatically protects itself from known and unknown flaws, threats and vulnerabilities.

The following techniques help guard the application resources:

- Authentication
- Authorization
- Security Model
- Security at different levels
- Best practices



Authentication Solutions

Authentication is a prerequisite for the security of a Web application. Properly implemented authentication prevents unauthorized access to a Web application and guards sensitive data which are high in value as well as demand for a company's competitors or hackers.

The following authentication solutions can be implemented in a Web application to provide secure access to it:

- EV SSL (Extended Validation – Secure Sockets Layer)
- SSO (Single Sign On)
- Digital Signatures / Identity Manager
- Wireless Authentication using RADIUS (Remote Authentication Dial In User Service), EAP (Extensible Authentication Protocol), LDAP (Light-weight Directory Access Protocol), etc.

- Usage of secure devices, cards, PIN, etc.
- Biometric Systems

Authorization Implementations

Authorization is the process of granting access rights of resources to users and user roles based on the users' identities and permission defined in the application's security policy or as already configured by the administrator. Authorization can be provided using one or a combination of the following authorization solutions:

- Principle of Least Privilege
- JAAS (Java Authentication and Authorization Service)
- Role-based Authorization
- Time-based Authorization
- Re-authorization for high value transactions
- Authorization Matrix
- Consumption-based

Security Models

There are three major attributes for ensuring the security of Web applications:

- Confidentiality
- Integrity
- Availability

Security policies for Web applications are framed around these attributes and a security model specifies how to implement the guidelines laid down in a security policy. A Web application can use a single or a combination of various Security Models. Security Models may be either:

- Role-based: Access is granted on the basis of the user's role
- Transaction-based: Access is granted on the basis of the transaction performed

A modified and less stringent version of the Information Security Models listed below can also be defined and implemented on the basis of requirements:

- Clark-Wilson Model: Prevents unauthorized access to the system by authorized users.
- Non-interference Model.

Implementing Security at Different Levels

In a follow-up to the discussion on various factors affecting the security of Web applications, a summary of various ways to implement security at different levels is given below:

Network Level	<ul style="list-style-type: none"> ■ Firewalls ■ Intrusion Detection Systems
Application Level	<ul style="list-style-type: none"> ■ Developer trainings ■ Tools for scanning proper testing
Application Server Level	<ul style="list-style-type: none"> ■ Policies ■ Better configuration ■ SSL deployment
Database Level	<ul style="list-style-type: none"> ■ Defining roles and privileges ■ Database Principle of Least Privilege ■ In a multi-tenant application, separate roles and privileges for users of each application instance
Maintenance	<ul style="list-style-type: none"> ■ Monitoring the application ■ Applying patches

Best Practices for Development of Secure Web Applications

Ensuring security for a Web application is the responsibility of every member involved in application development. Some of the best practices that must be followed across the application development life cycle are explained below:

- Planning
 - Put a security policy in place
 - Minimize the attack surface area
 - Define scope for each tenant application in the case of SaaS applications
- Implementation
 - Keep the whole team updated about the new threats and vulnerabilities
 - Use parameterized statements and Secure Socket Layer (SSL)
 - Implement the principle of Least Privilege
 - Encrypt sensitive data in the database
 - Always validate any information from the client strongly
 - In multi-tenant applications, all application instances should be independent and ensure data integrity
- Maintenance
 - Do not allow unlimited trials of wrong user name and password
 - Deploy an Intrusion Detection System

- Log all the major events of the application
- Perform database restoration without affecting other applications in the case of multi-tenant applications

Services offered by Sonata for Web Application Security

Sonata offers the following services to ensure secure and sophisticated Web applications:

- Vulnerability Assessment
- Security Testing
- Firewalls & Intrusion Detection Systems
- Updates for Threats and Vulnerabilities
- Maintenance and Support
- Hosting the Application in Secure Environment

Sonata's Value Proposition

Having provided security solutions for Web applications to customers from different domains, Sonata has gained a wealth of experience in the space for Web application security. The factors mentioned below give Sonata an upper hand for providing solutions and services for Web application security:

- Availability of highly skilled resources in the Security domain.
- Dedicated teams, always eager to learn new technologies and implement cutting-edge solutions.
- Expertise in multiple hosting environments like Cloud, Third-Party or In-House.
- (Open Web Application Security Project) OWASP-compliant security framework:
 - 500+ test cases for OWASP
 - Covers the top 10 OWASP security threats like cross-site scripting, SQL
 - Complete SDLC approach for end-to-end OWASP
 - Tool-based testing approach (WebScarab) to identify vulnerabilities.

Conclusion

To develop secure Web applications, one must understand different security threats and counter them with the relevant measures effectively. The security model, implementation of security at different levels such as network level, etc., and following the best practices for Web application security are critical for the development of secure Web applications.

For more information, contact info@sonata-software.com

Click here to know more about [Web Application Services](#)