ISSN No. 0976-5697

Volume 1, No. 4, Nov-Dec 2010

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

Managing Risk in E-commerce Security

Abhishek Khare
PG Department of Computer Science and Application
Shri Vaishnav Institute of Management, Indore,India
abhishekkhare@rocketmail.com

Abstract: Risk of E-commerce is an important basic research in the information security. It is one of the front subjects in IT field. In this Paper, risk factors and its possibility happening in E-commerce has been discerned, analyzed and evaluated. Most of risk factor in E-commerce is uncertain and fuzzy and its risk is difficult to be calculated. Technology has moved to Web Services programming, complicating the abstractness of project management. Standardization, as in the case of Web services is the need of the hour, viz. Simple Object Access Protocol-enabled applications, Hyper Text Transfer Protocol communicability etc, thus increasing the vulnerability of basic concepts in distribution designs. This again contributes to the risk factor in e-commerce projects. In this paper we have explored the major risks associated while doing business over internet and suggested remedies for the same. As businesses grow increasingly dependent upon Web applications, these complex entities grow more difficult to secure. Most companies equip their Web sites with firewalls, Secure Sockets Layer (SSL), and network and host security, but the majority of attacks are on applications themselves – and these technologies cannot prevent them.

Keywords: E-commerce,Audit, Transitory, Encryption,Firewall,Flooding, information security

I. INTRODAUCTION

This paper explains what you can do to help protect your organization, and it discusses an approach for improving your organization's Web application security.

What makes Web applications vulnerable?

In the Open System Interconnection (OSI) reference model, every message travels through seven network protocol layers. The application layer at the top includes HTTP and other protocols that transport messages with content, including HTML, XML, Simple Object Access Protocol (SOAP) and Web services.

This paper focuses on application attacks carried by HTTP—an approach that traditional firewalls do not effectively combat. Many hackers know how to make HTTP requests look benign at the network level, but the data within them is potentially harmful. HTTP-carried attacks can allow unrestricted access to databases, execute arbitrary system commands and even alter Web site content.

Typical Web application attacks

A Web application's specific vulnerabilities should dictate the technology you use to defend it. Figure shows many points within a system that might require protection. Often, it is best to employ generic countermeasure concepts first to help ensure that you choose the technology best suited to your needs rather than one that claims to counter the latest hacking technique.

To protect Web applications against attacks, enterprises should employ generic preventive approaches as well as targeted technologies.

Table: I shows common threats and preventive measures. However, specific threats to your application may be different.

Table : Common types of Web application attacks

Description	Common causes	Preventive measures
Impersonation		
Typing a different user's credentials or changing a cookie or parameter to impersonate a user or pretend that the cookie originates from a differ-ent server	Usingcommunications-based authentication to allow access to any user's data Using unencrypted credentials that can be captured and reused Storing credentials in cookies or parameters Using unproven authentication methods or authentication from the wrong trust domain Not permitting client software to authenticate the host	Use stringent authentication and protection for credential information using: Operating system (OS)-supplied frameworks Encrypted tokens such as session cookies Digital signatures

Tampering		
Changing or deleting a resource without authorization (e.g., defacing a Web site, altering data in transit)	Trusting data sources without validation Sanitizing input to prevent execution of unwanted code Running with escalated privileges Leaving sensitive data unencrypted	Use OS security to lock down files, directories and other resources Validate your data Hash and sign data in transit (by using SSL or IPsec, for example)
Repudiation		
Attempting to destroy, hide or alter evidence that an action occurred (e.g., deleting logs, impersonating a user to request changes)	Using a weak or missing authorization and authentication process Logging improperly Allowing sensitive information on unsecured communication channels	Use stringent authenti-cation, transaction logs and digital signatures Audit

Information disclosure		
Revealing personally identifiable information (PII) such as passwords and credit card data, plus information about the application source and/or its host machines	Allowing an authenti-cated user access to other users' data Allowing sensitive infor-mation on unsecured communication channels Selecting poor encryption algorithms and keys	Store PII on a session (transitory) rather than permanent basis Use hashing and encryption for sensitive data whenever possible Match user data to user authentication
Denial of service (DoS)		
Flooding—sending many messages or simultaneous requests to overwhelm a server Lockout—sending a surge of requests to force a slow ser-ver response by consuming resources or causing the appli-cation to restart	Placing too many applications on a single server or placing conflicting applications on the same server Neglecting to conduct comprehensive unit testing	Filter packets using a firewall Use a load balancer to control the number of requests from a single source Use asynchronous protocols to handle processing-intensive requests and error recovery
Elevation of privilege		
Exceeding normal access privileges to gain administrative rights or access to confidential files	Running Web server processes as "root" or "administrator" Using coding errors to allow buffer overflows and elevate application into a debug state	Use fewest-privileges context whenever possible Use type-safe languages and compiler options to prevent or control buffer overflows

Preventive measures can also be taken to ward off attacks that attempt to access sensitive information and overwhelm server resources.

Four strategic best practices for protecting Web applications

To address security-related issues as they pertain to Web applications, organizations can employ four broad, strategic best practices.

A. Increase security awareness

This encompasses training, communication and monitoring activities, preferably in cooperation with a consultant.

(a) Training

Provide annual security training for all application team members: developers, quality assurance professionals, analysts and managers. Describe current attacks and a recommended remediation process. Discuss the organization's current security practices. Require developers to attend training to master the framework's prebuilt security functions. Use vendor-supplied material to train users on commercial off-the-shelf (COTS) security tools, and include security training in the project plan.

With help from a third-party consultant, enterprises can employ training, communication and monitoring activities to improve security awareness.

(b) Communication

Collect security best practices from across all teams and lines of business in your organization. Distribute them in a brief document and make them easily accessible on an intranet. Get your IT security experts involved early and develop processes that include peer mentoring. Assign a liaison from the security team to every application team to help with application requirements and design.

(c) Monitoring

Ensure that managers stay aware of the security status of every application in production. Track security errors through your normal defect tracking and reporting infrastructures to give all parties visibility.

B. Categorize application risk and liability

Every organization has limited resources and must manage priorities. To help set security priorities, you can:

Define risk thresholds and specify when the security team will terminate application services.

Categorize applications by risk factors (e.g., Internet or intranet vs. extranet).

Generate periodic risk reports based on security scans that match issues to defined risk thresholds.

Maintain a database that can analyze and rank applications by risk, so you can inform teams of how their applications stack up against deployed

C. Set a zero-tolerance enforcement policy

An essential part of governing the development and delivery process, a well-defined security policy can reduce your risk of deploying vulnerable or noncompliant applications. During inception, determine which tests the application must pass before deployment, and inform all team members. Formally review requirements and design specifications for security issues during inception and elaboration—before coding begins. Allow security exceptions only during design and only with appropriate executive-level approval.

D. Integrate security testing throughout the development and delivery process

By integrating security testing throughout the delivery lifecycle, you can have significant positive effects on the design, development and testing of applications. You should base functional requirements on security tests your application must pass, making sure that your test framework

II. CONCLUSION

The Risk is a problem waiting to happen and the goal of risk management is to make this inherently risky process of applications development successful and consistent. A risk management approach is crucial to success in e-commerce; the need is for proactive risk mitigation both in planning and in development. In the development of a new technical system or project, there is a constant need to minimize uncertainty and errors, which accompany an unprecedented endeavor. Cyber security efforts focused on reporting and monitoring threats and vulnerabilities, education and security measures for "safe computing," research and development. The Internet, while opening the world's markets to virtually every business, has also broadened the risk of doing business.

III. REFERENCES

[1] Dr. Carl J. Case and Dr. Kimberly and S. Young, “Internet risk management: building a framework for research”, Published in the proceedings of the American Society of Business and Behavioral Sciences.
[2] Mark S. Frankel, and Sanyin Siang, “ Ethical and legal aspects of human subjects research on the internet”, Scientific Freedom, Responsibility and Law Program Directorate of Science and Policy Programs American Association for the Advancement of Science 1200 New York Avenue, November 1999
[3] http://www.methvenlaw.com/n_netbiz.htm
[4] http://www.engineeringtalk.com/news/wib/wib101.html
[5] http://www.entrepreneur.com/tradejournals/article/ 114168860.html
[6] http://www.cert.org/congressional_testimony/ PA_ecommerce_hearing_sep99.html
[7] http://www.bia.ca/articles/ldr-inherent-risks-foreign-country.htm
[8] http://www.cert.org/pres_comm/cert.rpcci.body. html#key.factors
[9] http://www.cert.org/pres_comm/cert.rpcci.body.html# assessment
[10] http://www.cert.org/pres_comm/cert.rpcci.body. html#implications

