Secure Web Based System Development

Oluleke Bamodu
Faculty of Computing, Engineering and Technology
Staffordshire University
Stoke-on-Trent, United Kingdom
indomitablejnr@engineer.com

Benson Otafu
School of Computer Science
University of Hertfordshire
Hatfield, United Kingdom
besomn@ojooo.com

Prof. Liwei Tian
Science and Technology Research Center
Shenyang University
Shenyang, China
tianliwei@163.com

***Abstract*— Web based systems which are applications developed for use in networked environments require very high level of in built security. This paper is an attempt to clearly lay out and explain the necessary skills, comparatively analyze several important lifecycles and propose threat modeling techniques for the development of secure web based system.**

Keywords- web based system; software development; threat modeling; lifecycle; threat mitigation; essential skills

I. INTRODUCTION

Software development is the design, maintenance and development of software applications mostly through programming. Software development is a multi-seriate process, often comprising of series of development stages such as problem identification, and solution-requirement analysis, conceptual planning and designing, implementation, testing, and debugging and deployment which is known as the development life cycle [1].

Web based systems or applications are software developed for use in a networked environment such as the internet. Since the introduction of the World Wide Web (WWW), more and more software have been moved from the usual stand alone computer system to the massively interconnected network. Information and data which were traditionally saved on personal or local hard drive are now being moved into databases which could be accessed from anywhere through a connection to the database. Although this has brought about great convenience, it also comes along with immense risks and as such necessitates pragmatic steps towards building higher level of security into such systems.

II. ESSENTIAL SKILLS FOR DEVELOPING SECURE WEB BASED SYSTEM

Developing software requires several skills, including, programming (writing and reading codes), use of library and framework, knowledge of software development life cycle (SDLC) [2]. To develop a web based system for retrieving sensitive data in a database, further skills such as threat modeling, analysis and mitigation, secure coding, testing and documenting with experience of making security fixes, cryptography and encryption, network security as well as database management are needed.

TABLE I. ESSENTIAL SKILLS

Threat modeling, analysis and mitigation	One of the downside of putting sensitive information on the internet is it being prone to attacks. Attacks do come in different ways and from different groups of people. An attack could come from motivated hackers who are all out to get access to the sensitive information for what it is worth, or it could originate from script kiddie which are not system experts and also aren't interested in the information, but break into system with pre-written codes just to cause mischief. To defend against such threats, skills in threat modeling, analysis and prevention are essential. Threat modeling provides a model which could be used in identifying security design faults, potential threats and corresponding necessary conditions for such threats and providing necessary solutions. Threat analysis on the other hand analyses the existing and new solutions to determine how effective they would be against possible attacks. The analysis helps in determining which vulnerabilities to eliminate for optimized results in mitigating the threats [3]. Hacking experience could be a part of this skill, this helps in thinking like the hackers so as to effectively defend against them.
Secure coding, testing and documenting with experience of making security fixes	Software development is a complex task, and a significant amount of it involves coding. Vulnerabilities found in software sometimes are introduced directly through the coding. eg buffer overrun, and usually prove to be fatal. To build a secure system, secure coding, testing and documenting skills are essential. Experience in fixing vulnerability and security glitches is also essential as fixes would need to be made as unknown vulnerability begin to show up during the life

	cycle of the system. It is worth noting here that the mentioned experience should not be only theory or book based, but should include practical experience on a real system [4].
Cryptography and encryption	Cryptography which is "the practice and study of techniques for secure communication in the presence of third parties" [5] and encryption which "is the process of obscuring information to make it unreadable without special knowledge" [6] are an essential skill for developing a secure web based system, especially one that deals with sensitive information. To preserve the confidentiality and integrity of the information stored in the database or when retrieving from the database, encryption is needed which necessitate this skill.
Network security	For a web based system, retrieving information from the database would mostly have to be done online, this calls for network security skills, such as firewall, intrusion detection system, TCP/IP, proxy and network hardware configuration to handle authentication and authorization of access to the data.
Database management	Development of a web based system that retrieve information from a database cannot be completed without relevant skill in database management. To make the system secure, database retrieval, architecture, storage, optimization and SQL skill become essential.

III. Comparison of van Wyk life- cycle with Boehm Lifecycle for Secure Web Based System Development

Software development life cycle is an important concept in software engineering; it introduces elements of engineering into software development. Many models and methodologies have been put forward based on SDLC, such as the Boehm life cycle, waterfall model, van Wyk model, V-model, iterative, agile, and extreme programming methodologies.

In the development of a secure web based system, van Wyk life cycle model is compared with the Boehm life cycle or spiral model as it is often referred to.

TABLE II. Comparison of van Wyk and Boehm lifecycle

Van Wyk Lifecycle	***Boehm Lifecycle***
➢ Van Wyk lifecycle is based on waterfall and iterative design methodologies which are matured system models.	➢ Boehm lifecycle is based on waterfall and iterative design methodologies which are matured system models.
➢ Continuous code, design and requirement testing, and refinement of key products or addition of new components to the system as more information or further requirements become available is possible which is highly required for a secure web based system. This short coming of pure waterfall model is address by the iterative method incorporated into the system.	➢ Continuous code, design and requirement testing, and refinement of key products or addition of new components to the system as more information or further requirements become available is possible which is highly required for a secure web based system. This short coming of pure waterfall model is address by the iterative method incorporated into the system [7].
➢ Risk analysis is built into the architecture which links business impact with technical security.	➢ Building of prototypes and simulation can be achieved easily with the model and can help with minimizing cost and risk [8].
➢ Supports multiple builds and orderly transitioning to a new build, which is very useful for secure web based system as continuous updates and security patches have to be applied as vulnerabilities in the web based system become known.	➢ Supports multiple builds and orderly transitioning to a new build, which is very useful for secure web based system as continuous updates and security patches have to be applied as vulnerabilities in the web based system become known [9, 10].
➢ Effective for risk management.	➢ Effective for risk management.
➢ Relatively new process with unknown size capacity.	➢ Could be used for large and complex project development, but quickly become ineffective when cost surpasses 2 billion dollars [9].
➢ Shorter development cycle, considered as a security life cycle model [8].	➢ Prototype development time cycle for original Boehm lifecycle model is known to be two year which would make it unsuitable for fast moving developments required in a secure web based system development [11, 12].

IV. STRIDE AND DREAD TO SECURE WEB BASED SYSTEM DEVELOPMENT

STRIDE and DREAD are components of the Microsoft defined Security Development Lifecycle (SDL). They fall under threat modeling, which is in the design phase of the 7 security practice groups made up of:

Training – Requirement – Design – Implementation – Verification – Release – Response [13]

STRIDE and DREAD are used in the definition and analysis of the possible security treats to a system, and could be applied to a web based system.

STRIDE, is an acronym which stands for:

Spoofing
Tempering
Repudiation
Information Disclosure
Denial of Service
Elevation of Privilege

Spoofing is the process of obtaining privileged information by illegally posing as someone else. Threat of spoofing in a web based system could be reduced by use of authentication. Encrypted password could be used to accomplish this.

Tampering involves unauthorized modification of stored data in a database or alteration of information passed over an open network. To prevent or guard against tempering, integrity of the information stored and passed over the network must be check and confirmed at all time. Information passed over the open network could be encrypted; a backup system or database could also be developed for the web based system to maintained integrity.

Repudiation is the denial of having received a service by a user where a service has actually been provided. To prevent this, a web based system can require verification or receipt evidence when a service is completed or use a form of tracking of user operation within the system.

Information Disclosure is letting out information to unauthorized users, or authorized users being able to read or access information without the required privilege to access such files. Confidentiality must be maintained, while data types, encryptions and privileges must be effectively designed and implemented to prevent information disclosure.

Denial of Service (DoS) usually is prevention or denial of services to valid users by flooding a system with massive load of requests beyond the systems processing capability. To prevent DoS, availability could be maintained through alternative backup systems.

Elevation of Privilege, grants an unprivileged person privileges enough to compromise or damage a system entirely to an unprivileged person. This can be minimized by requesting authorization [14].

To apply STRIDE to web based system development, threats can be listed out and categorized into 3 categories as in: network threats, host threats, and application threats.

Network threats include: DoS directed at web services, IP spoofing and tempering as a result of faulty firewall configurations.

Host threats include: lack of defined trust boundaries and vulnerability exploration through un-patched servers.

Application threats include: lack of or missing sensitive data encryption, buffer overruns and SQL injections.

After categorization, an "Attack Tree" should be made for use in determining threats and vulnerability as well as for drawing out mitigation options.

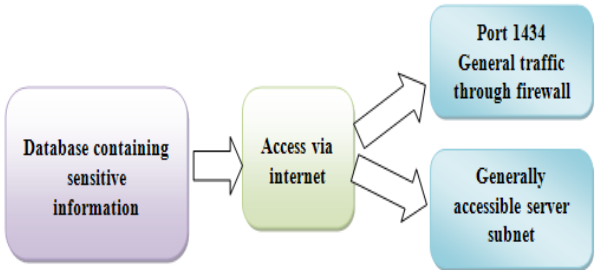

Figure 1. Attack Tree.

After the threats have been categorized and the attack tree modeled, the next step is to apply the DREAD [3].

The acronym DREAD stands for:

Damage Potential
Reproducibility
Exploitability
Affected Users
Discoverability

DREAD is used to gauge the danger or risk of a particular vulnerability usually in tabular form.

A range of 0 – 10 is used for rating the risk, with 10 being most critical and 0 least critical.

TABLE III. DAMAGE POTENTIAL TABLE

Extent of damage when a threat occurs.	10	The whole database is compromised or destroyed.
	5	Little information compromised or leaked.
	0	None.

TABLE IV. REPRODUCIBILITY TABLE

Whether or not the threat can be reproduced.	10	Threat can be reproduced every time without authentication.
	5	Threat only can be reproduced at a particular time and probably by an authorized user.
	0	The threat is very hard to reproduce even with authorized access and knowledge of the security glitches.

TABLE V. EXPLOITABILITY TABLE

Tools and skills needed for performing the threat..	10	Just basic tools and little programming knowledge and time.
	5	Considerable programming knowledge and malware tools.
	0	Advanced programming skills, advanced tools and considerable amount of time.

TABLE VI. AFFECTED USERS TABLE

Number of users affected by the threat.	10	All users.
	5	Some users.
	0	None.

TABLE VII. DISCOVERABILITY TABLE

Ease of discovering the threat.	10	Common fault available in other systems and can easily be search for online.
	5	Needs monitoring of the network and some guessing to discover the vulnerability.
	0	Almost impossible to guess and access to the source code will be needed.

After the DREAD table has been made, the risk is calculated as below.

Risk = (damage + reproducibility + exploitability + affected users + discoverability) / 5

Table VIII below shows an example of the dread table for two possible threat descriptions.

TABLE VIII. DREAD TABLE FOR THREAT DESCRIPTION

Description	*D*	*R*	*E*	*A*	*D*	*Risk*	*Rating*
Attacker obtains sensitive information from database through internet with blocked firewall port	0	0	0	10	10	4	Low
Attacker obtains sensitive information from database through internet with open firewall port.	5	10	5	10	10	8	High

Mitigation methods are to be applied to moderately critical to severely critical vulnerability, such as for description two in Table VIII above to reduce the threat and risk. A new DREAD table should also be made after the mitigation, to verify that the selected mitigation did resolve the earlier posed threats.

ACKNOWLEDGMENT

Special thanks to Joseph Spring for pointing out corrections to be made in the earlier version of this paper.

REFERENCES

[1] Wikipedia: Software Development. Available at http://en.wikipedia.org/wiki/Software_development

[2] B. Vandegriend, "The core skills all software developers need" 2009. Available at http://www.basilv.com/psd/blog/2009/the-core-skills-all-software-developers-need

[3] T. Olzak, "A practical approach to threat modeling," March 2006. Available at http://http://www.adventuresinsecurity.com

[4] M. Howard, and D. LeBlanc, Writing Secure Code, 2nd ed., Microsoft Press: Washington, 2003,pp.3-124

[5] Wikipedia: Cryptography. Available at http://en.wikipedia.org/wiki/Cryptography

[6] K. Zotos, and A. Litke, "Cryptography and Encryption". Available at http://arxiv.org/ftp/math/papers/0510/0510057.pdf

[7] Wikipedia: Waterfall Model. Available at http://en.wikipedia.org/wiki/Waterfall_model

[8] A Comparison of Three Life Cycle Models. Available at http://homepages.feis.herts.ac.uk/~comqjs1/A_Comparison_of_Three_LCMs.pdf

[9] Wikipedia: Spiral Model. Available at http://en.wikipedia.org/wiki/Spiral_model

[10] C. Larman, and V. R. Basili, "Iterative and Incremental Development: A Brief History," IEEE Computer (IEEE Computer Society) , Vol 36(6),pp.47–56, Jun. 2003, doi:10.1109/MC.2003.1204375.

[11] B. W. Boehm, "A spiral model of software development and enhancement," IEEE Computer , vol.21, no.5, pp.61-72, May 1988 doi: 10.1109/2.59

[12] H. Peter, "Interpreting the Spiral Model of Software – Intensive System Development - A ULCM Approach,". Available at http://sse.stevens.edu/fileadmin/cser/2006/papers/120-Hantos-Spiral%20Model%20ULCM.pdf

[13] Microsoft Security Development Lifecycle. Available at http://www.microsoft.com/security/sdl/default.aspx

[14] MSDN Magazine: Threat Modeling. Available at http://msdn.microsoft.com/en-us/magazine/cc163519.aspx