Integrating Risk assessment and Threat modeling within SDLC process

Maheshwari V
School of Information Technology
Vellore Institute of Technology, Vellore, India
maheshwari.v2014@vit.ac.in

Prasanna M
School of Information Technology
Vellore Institute of Technology, Vellore, India
Prasanna.m@vit.ac.in

***Abstract*— Risk assessment and threat modeling are conducted for different purpose. The integration of risk assessment and threat modeling process limit the risk of software-based system. Incorporating security in all phases of software development life cycle is a tedious task in many organizations. In design phase of SDLC, the 50 % software defects are identified and detected. Most of the security attacks are happen in application layer. This paper explains the combined use of risk assessment and threat model to understand the security risk of an application. We also discuss how the model may be identifying threats and how to frame threat prioritization for threat category. Finally, we recommend understanding of risk of detection and creating a fair environment to reduce the likelihood of committing criminal acts by attackers.**

Keywords—Risk-based testing; application layer; Software development life cycle (SDLC), Threat model.

I. INTRODUCTION

Security Risk is having a relevant impact on the development of today's software system. Building an application is not feasible without understanding of potential threats of targets in an application. Now-a-days, more web security is developed with new technologies as a result new attack techniques came into existence and increase a risk to an organization. Understanding the risk is the prerequisite step of analysis of threats.

Risk management is one of the most important activity in Software development life cycle(SDLC), commonly used to identify the threats of a software systems in early phases. By attacker and defensive perspective, the risk is analysed in all the phases of SDLC such as Training, Requirements, Design, Implementation, Verification, Release and Response.Threat and Vulnerability are a part of a risk, where risk is a guiding factor to support decisions in all the phases of test process[1].

Threat modeling is a structured approach to identify,quantify and address the security risk associateed with an application. Integrating threat model in the SDLC helps to increase the security at the very beginning stage of developing an application. Threat modeling process can be developed into three steps: Decompose an application, Determine rank and threats and Determine contermeasures and mitigation[2].The main objective of threat modeling is minimizing the risk and associated impacts.Still in many organization , security of software application is often addressed after implementation or deployment. More than 70% of security vulnerabilties existed at the application layer and not at the system or network layer.[3]

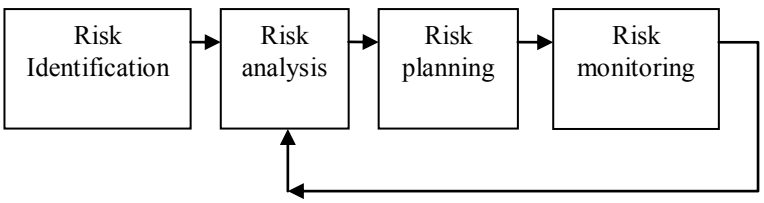
The rest of this paper is organized as follows .Section 2 describes background on risk management and Threat modeling. Threat model in secure development life cycle used for the representation of threats is described in Section 3.Techniques of threat model are discusses in Section 4. Threat approach and comparison are summarized in Section 5. Case studies explained in Section 6.Finally Section 7 draws conclusion.

II. BACKGROUND AND RELATED WORK ON RISK ASSESSMENT AND THREAT MODELING

A. Risk Management

The risk management process used by the software development organisation, as it was employed risks to be managed proactively from early phase of SDLC in order to discover threats and to significantly increase the security of the applications. The risk management process is often divided into four steps: Risk identification, Risk analysis, Risk planning and Risk monitoring. Most enterprise organization follows security throughout the software development life cycle.Security was a serious issue but it was not taken much consideration during the development stage.The risk value(R) is calculated for each risk by multiplying the probability (p) with the severity value (S), i.e. R= P*S. [4]

Fig 1: Risk Management process –SDLC



B.Threat Modeling

Threat model is one of most important activity in software security[5]. A security analyst discovers the actions that a malicious agent might perform in order to misuse a software system. Threats are often referred to as anti-requirements which provide insight into how a malicious user, attacker can abuse a system and established to determine what happens when this functionality goes away[6].The security weakness of the software system is identified by threat modeling technique called STRIDE.

III. THREAT MODEL IN SOFTWARE DEVELOPMENT LIFE CYCLE

The importance of integrating threat modeling during SDLC, to identify the security issues before the deployment of an application. The impact of a threat to target the vulnerabilities of the system and its severity of weakness is determined the likelihood, impact and need to be remediated [7]. The main advantages of embedding threat modeling in all phases of the SDLC are: Security requirements, secure design, security issue prioritization, secure release of applications after development, secure release of applications after an incident [8].From the business risk perspective, the risk mitigation decision is identified by threats, attacks and vulnerabilities. Mapping of threat to vulnerabilities and vulnerabilities to asset helps to mitigate the threats through security measures.

A. Threat Model in Secure development Life cycle

Firstly the threat model is built by defining its requirements, attributes, dependencies of data and trust boundaries related to an application. Then threats are identified from attacker and defensive perspective such as process, data flow, data store and interaction with user. And finally countermeasures of known attacks associated with the application list that enables the programmer to reduce the chances of attacks (Fig 2).

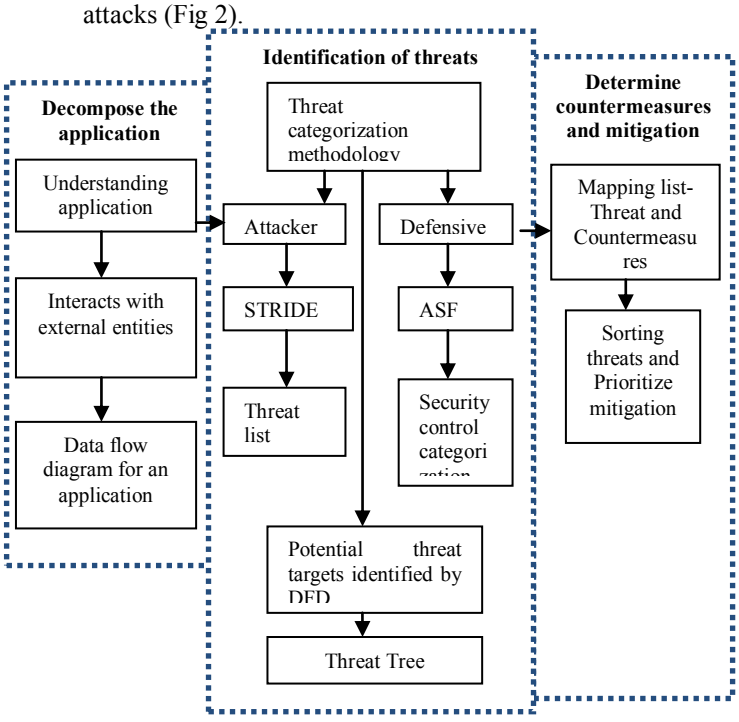


Fig 2: Represents Threat modeling process in SDLC

The type of threats for an each component is determined for an application in a form of threat tree, vulnerabilities and countermeasures. It is an iterative process to evaluate threat identification. The use of this model is to predict and prevent risk even before start of coding (Fig 3).

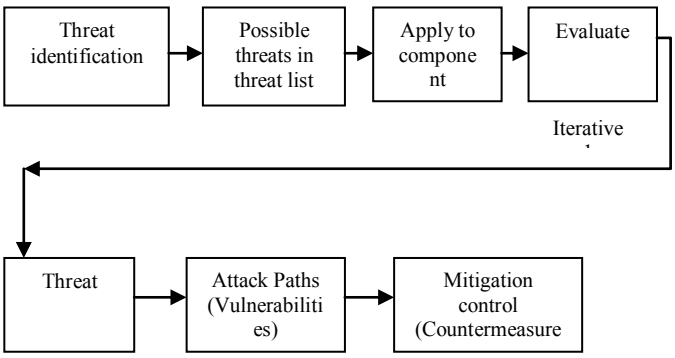


Fig 3: Iterative process of threats

IV. TECHNIQUES OF THREAT MODELING

Basically in software, the risk analysis methodologies are classified into two ways i.e. commercial and standard methods. Each and every method has some commonalities and distinct in behavior, feature etc. The weakness is normal for a system design framework, where software flaws are identified and conceivable risk is determined. The risk assessment validates the security requirements of the software system where the most common security properties are integrity, confidentiality, authentication, authorization and non-repudiation.

The security risk is identified by threat modeling approach at the beginning stage of design phase of an application. There are N number of methods to measure risk and determine the countermeasure and mitigation [9] [10]. Threat modeling techniques are discussed below:

A. STRIDE

The standard threat model applied in most of the Microsoft products which has stride technology that is abbreviated as S-Spoofing, T-Tampering, R-Repudiation, I-Information disclosure, D-Denial of Service, E-Elevation of privilege. The key factors are threat model information, external dependencies, entry points, assets, trust levels, data flow diagrams. STRIDE is based on data flow diagram where DFD elements are mapped to the threat categories. The checklist represents the threat where threats are resembled as tree-based structure to provide a depth of each threat.

Finally the threats are documented to find out the misuse cases in security requirements.

B.DREAD

Dread is a threat- risk ranking modeling tool developed by Microsoft. Based on risk factors the priority is given for the risk as high, low and medium. DREAD is abbreviated as D-Damage, R- Reproducibility, E- Exploitability, A- Affected Users and D- Discoverability.

C.Generic risk model

The formula for generic risk model is:
Risk = Likelihood× Impact. The probability or likelihood is characterized by the simplicity of abuse, which relies on the type of threats, system attributes and understands of risk.

V. Software Threat Approaches

The threat modeling has three phases i.e. Inception phase, Object identification phase, Reactive phase. In inception phase, the data flow diagram is sketched for an application, entry and exit points and assets are identified. In next phase, threat effects are found in use scenario and feature scenario. At last in reactive phase the ranking of threats is done through stride classification, dread classification, threat trees, mitigation and reporting. The threat approach [11] is distinguished in Table 1.

VI. Case Study

Problem definition: The life insurance system is explained using STRIDE model- based threat modeling technique. The DFD model is drawn by initial activities of the system. The information is represented for the case study through data flows, external entities, processing nodes, data storage and database components. A Cloud Broker, Cloud Customer and the Cloud Service Provider present customer feedback, points are awarded to service provider services by the broker and trustworthiness of the service provider is monitored Fig.4. The threat modeling of life insurance system has revealed various ways of potential attacks. They are specified by STRIDE threat modeling tool. Table 3 shows the threat generated report, where one threat involves multiple threat types.

The above scenario demonstrates the customer registration request to cloud provider where insurance process, data store and interaction within the user to identify the threats. This tool illustrate how protective measures is exist, lack of protection exist can be analyzed. Fig 4.1, 4.2

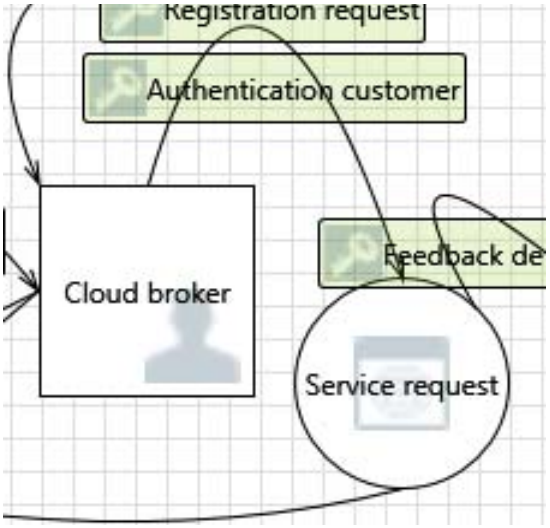


Fig 4.1 Authentication process

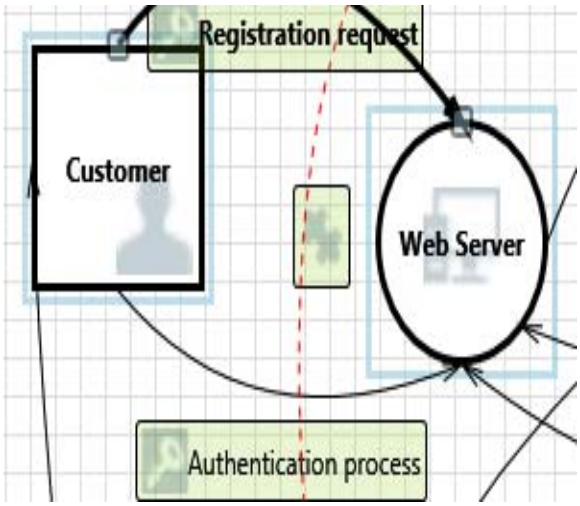


Fig 4.2 Authentication Customer

VII. Conclusion

This paper is provided to introduce an integration of risk management and threat analysis and modeling for applications. The commonalities of risk assessment and threat modeling are assessing risk, determine potential threats and mitigate threats. A real time scenario of life insurance system is modeled in Microsoft threat modeling tool to predict and prevent risk problems, even better at early stage of coding stage. The drawback of STRIDE is very hard to quantify the cost and effectiveness and also it doesn't generate the list of threats. In future, the prototype of Microsoft SDL has to improve with the specific use of threats to generate in DFD editor.

Table 1-Threat approach

Threat Approach	Goal	Representation	Tools	Uses
Software centric	Design of a threat model system.	Data flow diagram, Use case or component diagram.	Microsoft SDL, TAM (Threat Analysis Modeling)	Specified security controls are identified.
Asset centric	Identify asset	Attack tree , Attack graph	Trike, Amenza Securitree	Attackers reach an asset through attack paths. Paths are weighted and prioritized respectively.
Attacker centric	Specific goals of an attacker	Tree diagram	Attackers profile, Analyst	Detect or mitigate an attack.

A. Comparison of threat modelling methodologies

Table 2- Threat modeling technologies

Contents	STRIDE	PASTA	TRIKE
Primary goal	Understanding threats, threat classification and security properties of a threat model at the beginning stage of a design phase.	Threat management, enumeration and scoring are done for business objective, compliance requirements along with business impact analysis.	Threat model for security auditing process. It follows risk-based approach.
Security properties	Confidentiality, Integrity, Availability, Authentication, Authorization, Non-Repudiation	Combines threat modeling approaches i.e. attack-centric and asset-centric	Mitigation control

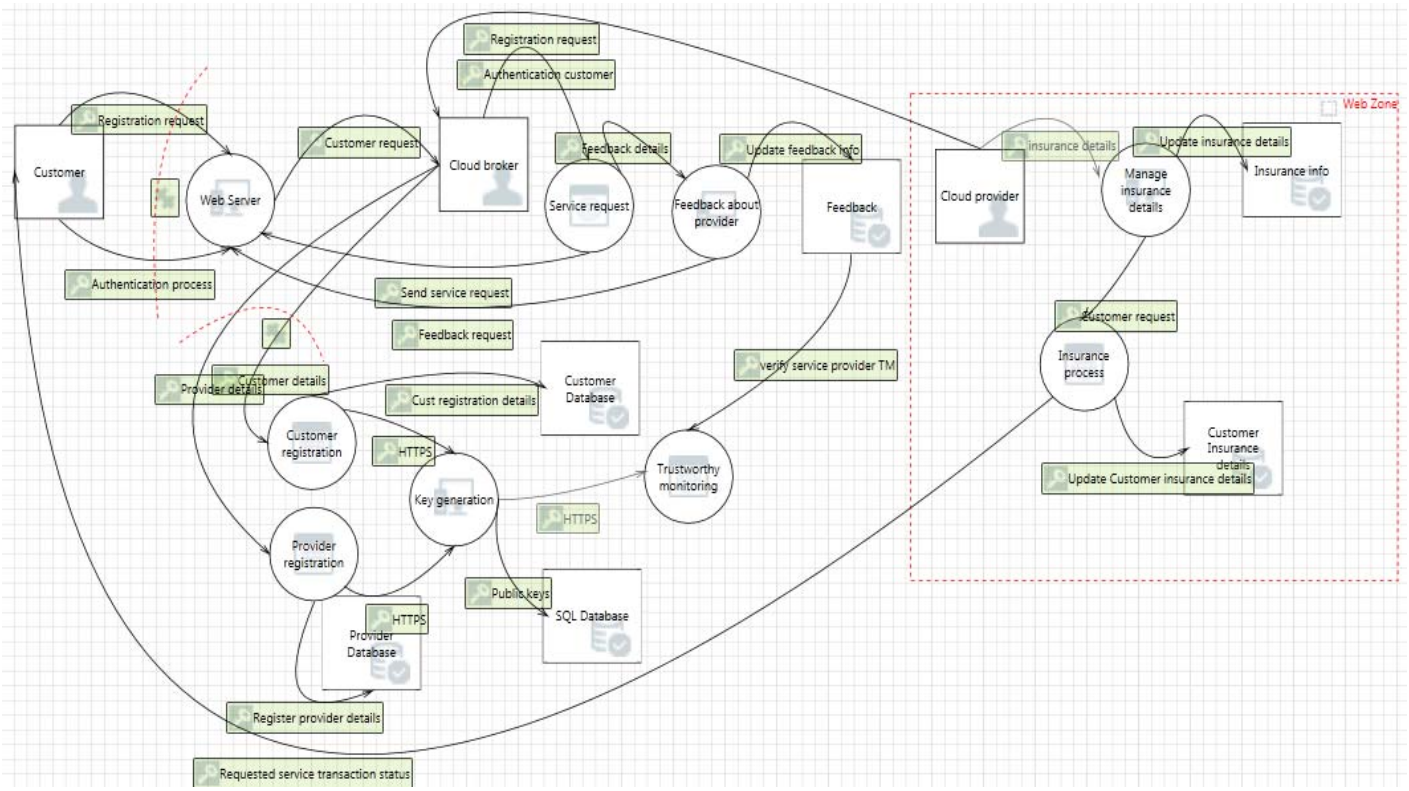


Fig 4. DFD diagram for Life Insurance system using Threat modeling tool –STRIDE

Table 3-Threat modeling report

Sr. No	Interaction	Threat Type	Category	Description	Priority
1	Authentication customer	Elevation Using Impersonation	Elevation Of Privilege	Service request may be able to impersonate the context of Cloud broker in order to gain additional privilege	High
2	Authentication customer	Spoofing the Human User External Entity	Spoofing	Cloud broker may be spoofed by an attacker and this may lead to unauthorized access to Service request. Consider using a standard authentication mechanism to identify the external entity.	High
3	Authentication process	Cross Site Scripting	Tampering	The web server 'Web Server' could be a subject to a cross-site scripting attack because it does not sanitize untrusted input	High
4	Authentication process	Spoofing the Customer External Entity	Spoofing	Customer may be spoofed by an attacker and this may lead to unauthorized access to Web Server. Consider using a standard authentication mechanism to identify the external entity	High
5	Authentication process	Elevation Using Impersonation	Elevation Of Privilege	Web Server may be able to impersonate the context of Customer in order to gain additional privilege	High
6	Authentication process	Potential Data Repudiation by Web Server	Repudiation	Web Server claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data	High

REFERENCES

[1] Farahmand, F., & Spafford, E. H. (2013). Understanding insiders: An analysis of risk-taking behavior. *Information systems frontiers*, *15*(1), 5-15.

[2]Scandariato, R., Wuyts, K., & Joosen, W. (2015). A descriptive study of Microsoft's threat modeling technique. *Requirements Engineering*, *20*(2), 163-180.

[3] Fong, E., & Okun, V. (2007, January). Web application scanners: definitions and functions. In *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on* (pp. 280b-280b). IEEE.

4]Lindholm, C., Notander, J. P., & Höst, M. (2014). A case study on software risk analysis and planning in medical device development. *Software Quality Journal*, *22*(3), 469-497.

[5] Toor P(2005) Demystifying the Threat Modeling Process. IEEE Security Priv 3(5):66- 70.

[6] C. Warren Axelord, The Need For Functional Security Testing, CrossTalk- Mar/Apr 2011.

[7] Howard M, Lipner S(2006) The Security Development Lifecycle. Microsoft Press.RedMond.

[9] Shoastack A (2008) Experiences Threat Modeling at Microsoft In:Workshop on Modeling Security.

[10] Dhillon, D. (2011). Developer-driven threat modeling: Lessons learned in the trenches. *IEEE Security & Privacy*, (4), 41-47.

[11].https://www.owasp.org/index.php/application_threat_modeling]