Global Journal of Multidisciplinary Studies
Volume 10, Issue 7, June 2021
ISSN:- 2348-0459 Available online at www.gjms.edwin.co.in

BASICS OF THREAD DRIVEN MODELLING

Dhirendra Pandey, Md. Tarique Jamal Ansari
Department of Information Technology
Babasaheb Bhimrao Ambedkar University Lucknow, India

***Abstract*— Nowadays the problem of cyber threat is growing very fast. It includes newer classes of attacks such as insider attacks, email viruses, password attacks and DoS attacks, which are currently recognized as serious security attacks. These attacks have resulted in aggressively increasing security problems. However, threat modeling and threat analysis tools have not evolved at the same rate. In this paper, we have presented an overview of threat modeling, which can be helpful to avoid said classes of attacks.**

Keywords—Software Security; Threat Modeling; Threat Modeling Approaches; Non-Functional Requirements; STRIDE;

I. Introduction

Caring for security at requirements engineering time is the new attention of current world in recent days. Security Requirements Engineering is becoming emerging research area in software engineering, with the realization that security must be considered early during the requirements phase.

However, it is not yet very clear how to achieve this systematically through the various stages of the requirements engineering process. Security is the key issue for assuring the quality full software. Security is non-functional requirement that is why most of the times it is ignored in the requirements phase of Software Development Life Cycle. But, it is possible to reduce software development cost and time to identify user security requirement in the early stage of the software development process.

The main deal is to present the security requirements combining with user functional requirements which are collected form requirement phase in Software Development Life Cycle (SDLC). If we can extract user security requirements and present these requirements in requirements phase then secure software develop will be ensure from the very beginning. Threat modeling is a structural designing of all the information that affects the security of an application.

It is the approach to the modeling, specification and analysis of application specific security requirements. Threat trees are built systematically that are either software vulnerabilities noticeable by the attacker or anti-requirements implementable by this attacker. New security requirements are then obtained as countermeasures by application of threat resolution operators to the specification of the anti-requirements and vulnerabilities revealed by the analysis. The organization of the paper is as follows:

Section I will describe brief introduction to the topic. Section II will describe basic threat modeling concepts. Section III will describe threat modeling approaches. Section IV will describe the different types of threat categories. Finally, section will conclude the paper.

II. Threat Modeling

A *threat* is the adversary's goal, or what an adversary might try to do to a system [2]. Threat modeling is recognized as one of the most important activities in software security [9]. Modern society is critically dependent on a wide range of software systems. Software applications are increasingly ubiquitous, heterogeneous, mission-critical and vulnerable [20]. Threats from a software security breach could range from the very mild (such as the defeat of copy protection in a video game) to the disastrous (such as malicious intrusion into a nuclear power plant control system). With the advent of the Internet, and increasing reliance on public packet-switched networks for e-commerce, telecommuting, etc., the risks from malicious attacks are increasing. Software system designers today must think not only of users, but also of adversaries [1]. Threat modeling approach for identifying, documenting and mitigating security threats to a software system has been given by.

Identifying threats helps develop realistic and meaningful security requirements. This is particularly important, for if the security requirements are faulty, the definition of security for that system is faulty, and thus the system cannot be secure. Proper identification of threats and appropriate selection of countermeasures reduces the ability of attackers to misuse the system [15]. Security must be deeply integrated into the full software development life cycle to match the evolutionary nature of threats manifested these days. Different authors and researchers are already working in this direction for developing innovative techniques to meet these security challenges [3, 4]. Threat modeling is also used to refer, variously, to analysis of software, organizational networks or systems, or, as in [6]. Threat modeling can be used to analyze the soundness of (initial) software architectures and to spot flaws early on. The discovered flaws represent an opportunity to elaborate upon the security requirements of the system and, consequently, revisit the design choices or refine the architectural model.

A popular technique for threat modeling is Microsoft's STRIDE [10]. A threat modeling technique guides the security analyst to the discovery of the actions that a malicious agent (insider or outsider) might perform in order to misuse a software system. Threats are often referred to as anti-requirements and are an important driver for the definition of the security requirements of a system [11, 14, 15]. Any type of system can benefit from threat modeling. Some systems are fairly simple and others are more complex, some of them are already deployed and others exist only on paper. No matter what the system is or what stage of the development process it is in, the benefits from well thought out threat model can prove extremely useful [13]. When threat modeling happens, it is usually done by experts who have learned the art in an apprenticeship model, often an informal one. At its best, threat modeling leads to both the cataloging of threats which are known to experts, as well as the discovery of new threats against a system, all before it is built, allowing the design to be modified to produce a more resilient system.

III. Threat Modelling Approaches

Threat Modeling can be viewed in two different, but related contexts. One is the implementation of security controls by architects that map to security requirements and policy and the other is to reflect all possible known attacks to components or assets in a threat model, with the goal of implementing countermeasures against those threats. The three general approaches to threat modeling are:

- Software-centric
- Asset-centric
- Attacker-centric

A. Software-Centric Threat Modeling

Software-centric threat modeling is also called as 'system-centric' or ''design-centric' or 'architecture-centric' .It starts from the design of the system, and attempts to step through a model of the system, looking for types of attacks against each element of the model. This approach is used in threat modeling in Microsoft's Security Development Lifecycle (SDL) [7].

B. Attacker-Centric Threat Modeling

Attacker-centric threat modeling starts with an attacker, and evaluates their goals, and how they might achieve them. Attacker's motivations are often considered and given importance than any other factor. This approach usually starts from either entry points or assets [7]. This threat modeling approach focuses on the identification of all possible access points to the system and the possible adversary aims. In general the attacker aim can be one or more of the following: Spoofing, Tampering, Repudiation, Information disclosure, Denial of services and Elevation *of privileges (STRIDE).*

Attacker-Centric based threat models are popular among security experts although they lack adequate semantics to allow reasoning about threats they represent. Because of lack of

adequate semantics, security controls developed based on Attack-Centric approach suffice instead of being utility maximizing.

C. Asset-Centric Threat Modeling

Asset-centric threat modeling involves starting from assets entrusted to a system, such as a collection of sensitive personal information. There are various kinds of assets and these assets are crucial in this approach.

IV. List of Threats

Threats may be insider or outsider of the system from authorized users or from unauthorized users who masquerade as valid users or find ways to bypass security mechanisms. Threats can also come from human errors [13]. Threats can be categorized into six classes based on their effect.

D. *Spoofing*

Spoofing allows an adversary to pose as another user, component, or other system that has an identity in the system being modeled [18]. Given the possible instances of web services within the web application, the scenario where the client is spoofed in its communication with the web server is considered the most relevant. Weak or no authentication of the client can lead to unauthorized access to the web service [8]. Spoofing refers to usage of someone else's credentials to gain access to otherwise inaccessible assets. If the attack appears to be in spoofing category, appropriate authentication mechanism need to be implemented [7]. Spoofing threats against the customer element can be mitigated by avoiding attack surfaces for social engineering, e.g. through random password generators rather than simple passwords, and offering user awareness programs and training.

E. *Tampering*

Tampering refers to concept of altering data to mount an attack or the modification of data within the system to achieve a malicious goal [18]. All the attacks in which someone changes some information without permission fall into this category [7]. Tampering can be done while data is on the communication channel, while data resides on the consumer machine, or while it resides on the provider machine [8].

F. *Repudiation*

Repudiation occurs when a user denies performing an action, but the target of the action has no way to prove otherwise. All the attacks in which someone denies a transaction that was performed are mapped into this category. For example, someone denying a purchase order after receiving the merchandise and denying the payment is classified as repudiation [7]. Repudiation threats are by nature application specific and are not further detailed here. Web services do provide countermeasure technologies here, such as XML signatures [8]. It is the ability of an adversary to deny performing some malicious activity because the system does not have sufficient proof otherwise [18].

G. *Information disclosure*

The exposure of protected data to a user that is not otherwise allowed access to that data [18]. Information disclosure refers to disclosure of information to a user who does not have permission to see it. All the attacks in which someone gets to see information she has no right to access can be termed as information disclosure [7].

H. *Denial of service*

It refers to reducing the ability of valid users to access resources. All the attacks in which someone breaks the system and prevent it from working normally and supplying the service it should fall into this category [7]. Denial-of-service attacks try to disturb the services by overloading the communication line, or by enforcing a crash or ungraceful degradation of the consumer or provider [8]. It occurs when an adversary uses illegitimate means to assume a trust level with different privileges than he currently has [18].

I. Elevation of privilege

Elevation of privilege occurs when an unprivileged user gains privileged status. All the attacks in which someone enhances their capabilities by raising their privileges fall into this category. Example is when the attacker manages to get administrative rights [7]. Elevation of Privilege is a card game for developers which entice them to learn and execute software-centric threat modeling [19].

Property	**Description**	**Threat**	**Definition**
Authentication	The identity of the user is established.	Spoofing	Impersonating something or someone else
Integrity	Data & System resources are only changed by intended people	Tampering	Modifying data or code
Non-repudiation	User can't perform an action and later deny it	Repudiation	Claiming to have not performed an action
Confidentiality	Data available to only intended persons	Information Disclosure	Exposing information to unauthorized person
Availability	System is ready when needed and perform fine	Denial of Service	Deny or degrade services to user
Authorization	Users are explicitly allowed or denied to access resources	Elevation of Privileges	Gain capabilities without proper authorization

Table: STRIDE Security concepts

The above table shows the properties, description and definition of STRIDE threat model.

V.CONCLUSION

Threat modeling is useful not only to web applications but also to embedded systems, cloud applications, wireless sensor networks, network tools etc for threat evaluation and risk analysis along with mitigation suggestions to them. Threat modeling for an application takes a lot of brainstorming sessions to collect all information of the assets, trust boundaries and threat profiles possible on the assets. In this paper we have proposed threat modeling as an essential for defining non functional security requirements of software application. Without identifying threats, it is impossible to provide security assurance for the application system. We have also discussed different approaches of threat modeling and also presented list of different types of threats.

REFERENCES

1. Devanbu, Premkumar T., and Stuart Stubblebine. "Software engineering for security: a roadmap." Proceedings of the Conference on the Future of Software Engineering. ACM, 2000."
2. Frank Swiderski and Window Synder, "Threat Modeling", Microsoft Press, 2015.
3. Punam Bedi, Vandana Gandotra, Archana Singhal, Vandita Vats and Neha Mishra, "Avoiding Threats Using Multi Agent System Planning for Web Based Systems". 1st International conference on Computational Collective Intelligence – Semantic Web, Social Networks and Multiagent Systems, Wroclaw, Poland, October 2009, LNAI, Springer-Verlag Berlin Heidelberg, pp.709-719, 2019.
4. Gary McGraw, Software Security: Building Security In, Addison-Wesley Software Security Series, 2006.
5. Shostack, Adam. "Experiences threat modeling at microsoft." Modeling Security Workshop. Dept. of Computing, Lancaster University, UK. 2008.
6. Craig Rubens, Cleantech Terror Alert: Hacking the Grid, Earth2Tech, June 26, 2018, http://earth2tech.com/2008/06/26/ cleantech-terror-alert-hacking-the-grid/
7. Jangam, Ebenezer. Threat Modeling and its Usage in Mitigating Security Threats in an Application. Diss. National Institute of Technology Karnataka Surathkal, 2020.
8. Desmet, Lieven, et al. "Threat modelling for web services based web applications." Communications and multimedia security. Springer US, 2005.
9. McGraw, G.: Software Security: Building Security In. Addison-Wesley (2006)
10. Torr, P.: Demystifying the threat-modeling process. IEEE SEcurity and Privacy 3(5) (2005).
11. van Lamsweerde, A.: Elaborating security requirements by construction of intentional anti-models. In: International Conference on Software Engineering (ICSE) (2014)
12. Möckel, Caroline, and Ali E. Abdallah. "Threat modeling approaches and tools for securing architectural designs of an e-banking application."Information Assurance and Security (IAS), 2010 Sixth International Conference on. IEEE, 2010.
13. Myagmar, Suvda, Adam J. Lee, and William Yurcik. "Threat modeling as a basis for security requirements." Symposium on requirements engineering for information security (SREIS). Vol. 2005. 2005.
14. Sindre, G., Opdahl, A.L.: Eliciting security requirements with misuse cases. Requirements Engineering 10(1), 34–44 (2005)
15. Myagmar, S., Lee, A., Yurcik, W.: Threat modeling as a basis for security requirements. In: Symposium on Requirements Engineering for Information Security (SREIS) (2005).
16. Hernan, Shawn, et al. "Threat modeling-uncover security design flaws using the stride approach." MSDN Magazine-Louisville (2006): 68-75.
17. Steffan, Jan, and Markus Schumacher. "Collaborative attack modeling."Proceedings of the 2002 ACM symposium on Applied computing. ACM, 2002.
18. Burns, Steven F. "Threat modeling: A process to ensure application security." GIAC Security Essentials Certification (GSEC) Practical Assignment (2015).
19. Shostack, Adam. "Elevation of Privilege: Drawing Developers into Threat Modeling." 2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14). 2014.
20. R.A. Kemmerer, "Cybersecurity", Invited Mini-Tutorial, Proc. ICSE'03: 25th Intl. Conf. on Software Engineering, Portland, IEEE CS Press, May 2013,pg.705-715.