Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 7, July 2015, pg.174 – 179

RESEARCH ARTICLE

Threat Modeling Methodology and Tools

Pooja Lahoti[1], Pragya Shukla[2]
[1]Department of Information Technology with specialization in Information Security, Institute of Engineering and Technology, Devi Ahilya Vishwavidyalaya (DAVV) University, India
[2]Department of Information Technology with specialization in Information Security, The Institute of Engineering and Technology, Devi Ahilya Vishwavidyalaya (DAVV) University, India
[1] email: plahoti1389@gmail.com [2] email: pragyashukla_iet@yahoo.co.in

Abstract— ***Now-a-days, Security is a subject of concern. Providing Security to an application after development is costlier and time consuming, instead Microsoft introduced one method- Threat Modeling method to provide security analysis before the actual development of the application prior in design phase. Microsoft Security Development Life-cycle includes one practice as Threat Modeling to identify vulnerabilities, determine risks from the threats, and establish appropriate mitigation methods before the actual development to ensure the security at design level, so as to reduce threats to the system by 50 percent. In this document we will look in to the Threat Modeling Methodology, its concepts and facts. We examine the different tools introduced and there pros and cons to the system as well as how they effect the security .***

Keywords— ***SDLC, Security, Threat Modeling, Web Applications, Threat Model***

I. INTRODUCTION

Microsoft SDL Threat Modeling: A process used to implement application security in the design process of SDL in a relatively formal way. It is the process of designing a security specification to understand the security threats to a system, determine risks from those threats, and establishes appropriate mitigation and then eventually testing that specification. The Security Development Lifecycle (SDL) is a software development process addresses best practices to develop secure software and reducing development cost. [7], [9], [16].

In 1999, Microsoft has had documented Threat Modeling methodologies. After the valuable inclusion of Threat Modeling in SDL by Microsoft many companies have been using the practice as a security basis to find design flaws during development. [8], [10]

Threat Modeling looks at a system from an attacker's perspective to predict security attacks and then securing data so that attacker will not be able to attack a system, or generally just finding out the vulnerabilities. [6].

It involves the identification of entry point's formal and informal, privilege boundary definition and threat visualizations. Privilege boundary mapping is the assignment of access rights to system objects and threat visualization is a formal presentation of threats using techniques like attack trees, security pattern description, and attack nets. [11]

Threat Model provides a structure to design phase Threat Model is a security based analysis that helps people determine the highest level security risks posed to the product and how attacks can manifest themselves. Threat Modeling uses Data Flow Diagrams (DFD) with security annotations. [12]

General three approaches to Threat Modeling exist and these are: Asset-centric, Attacker-centric or Software-Centric, System-Centric or Threat –Centric

II. Literature Survey

- *Experiences Threat Modeling at Microsoft [1]*: The paper covers experiences of threat modeling products and services at Microsoft. Describes the current threat modeling methodology used in the Security Development Lifecycle.

- *Checking Threat Modeling Data Flow Diagrams for Implementation Conformance and Security [2]:* The Paper covers basics of Threat Modeling and its DFD and also designed an analysis to assist DFD designers validate their initial DFDs and detect common security design flaws in them.

- *Threat Modeling: as a Basis for Security Requirements [3]:* The paper covers how threat modeling can be used as foundations for the specification of security requirements.

- *Threat Modeling Revisited: Improving Expressiveness of Attack [4]:* This paper reviews the existing techniques and proposes threat net; a technique based on information and causality theory concepts which offer improved expressiveness and semantics of threat models.

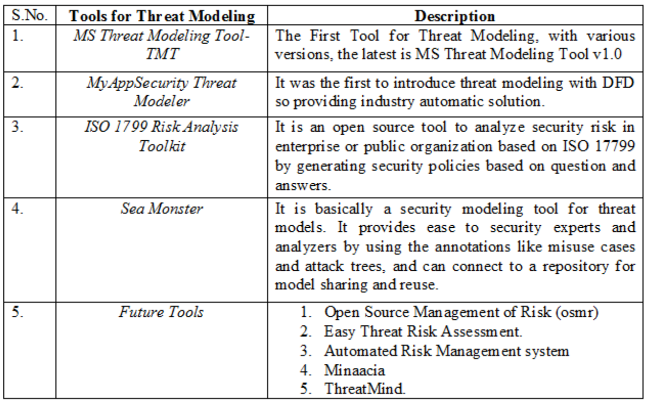
- *Threat Modeling: A process to ensure Application Security [5]:* This paper discusses the importance of implementing application security at design time. It also outlined and defined the process of creating Threat model.

III. Comparison between Tools

There are many tools introduced in the life of Threat Modeling. The very first tool is introduced by Microsoft well known as *Threat Modeling tool TMT* which was upgraded by Microsoft and recently introduced in 2014 with DFD. MyAppSecurity

Table I

Threat Modeling Tools



S.No.	**Tools for Threat Modeling**	**Description**
1.	*MS Threat Modeling Tool-TMT*	The First Tool for Threat Modeling, with various versions, the latest is MS Threat Modeling Tool v1.0
2.	*MyAppSecurity Threat Modeler*	It was the first to introduce threat modeling with DFD so providing industry automatic solution.
3.	*ISO 1799 Risk Analysis Toolkit*	It is an open source tool to analyze security risk in enterprise or public organization based on ISO 17799 by generating security policies based on question and answers.
4.	*Sea Monster*	It is basically a security modeling tool for threat models. It provides ease to security experts and analyzers by using the annotations like misuse cases and attack trees, and can connect to a repository for model sharing and reuse.
5.	*Future Tools*	1. Open Source Management of Risk (osmr) 2. Easy Threat Risk Assessment. 3. Automated Risk Management system 4. Minaacia 5. ThreatMind.

Here we are comparing between the two most popular tools in IT industries viz. Microsoft Threat Modeling Tool (TMT) and MyAppSecurity Threat Modeler.

SDL Threat Modeling Tool [14], [15], [16]: SDL Threat Modelling Tool developed by Microsoft is based on the STRIDE model. Data Flow Diagram of the system under consideration is developed. This Data Flow Diagram is then passed as input to these tools. These threat modelling tools then generates reports describing all the possible threats to the system. In addition to this threat report, relevant mitigation measures are also identified.
Microsoft introduced the first Threat Modeling tool *SDL Threat Modeling Tool –TMT,* which isn't designed for security experts but for making threat modeling easier for all developers by providing guidance and creating and analyzing threat models.
The SDL TMT enables any software architect or developer to:

- Discuss about the security design of their system.
- To determine potential security issues using well defined steps by analyzing the designs.
- To mitigate security issues by suggesting some measures to be taken.

The following comparison is intended to provide the analysis between Microsoft TMT and MyAppSecurity Threat Modeler for security professional to have a decisive ability to choose between according to the application requirement.

MyAppSecurity Threat Modeler [13]: Threat Modeler, the industry's first automated Threat Modeling solution, is now available via a subscription service.

Assessment Criteria
The criteria used for comparing the tools were collaboration, report, functionality, and other features of both products.

Comparison of MS-TMT with MyAppSecurity ThreatModeler [13]:

- *Design:* Microsoft Threat Modeling Tool builds a threat model based on the components like database services, ports, protocols and web services, etc.

  MyAppSecurity's Threat Modeler, provides a threat modeling framework that encompasses a high level component based design, combined with a software-centric approach.

- *Reports:* MS-TMT as well as MyAppSecurity Threat Modeler generates reports that list the identified threats for the application and their current status like solved, unsolved or not possible to mitigate.
  Threat Modeler generates complete out-of-the-box reports as well as creating custom reports to meet the unique requirement of the associates.

- *Threat library:* MS-TMT and Threat Modeler have built-in Threat Library i.e., pre-defined repository of common threats based on industry standards and best practices. CAPEC, WASC-TC, OWASP Top 10 are well-known industry sources to provide the threat list are located in a single place in the Threat Library of both the tools.

  MS-TMT and MyAppSecurity Threat Modeler provides customizable threat libraries, so we can add specific threats in the threat library according to the industry or organization practice and mitigate it the threats at the first place.

- *Threat Management:* Both provide dashboard for Threat Management that provides current status of identified threats.

- *Components:* To customize components like Data elements, Widgets, Protocol, etc. according to enterprise application architecture, MS-TMT have limited customization.

- *Updates:* The Threat Modeler provides Quarterly Threat Library Updates whereas MS-TMT has limited frequency or does not have any Threat Library updates with the latest threat data.

- *Web-based:* Threat Modeler users can access the tool via a web browser so to prioritize and manage risk across all threat-Modeled applications whereas MS-TMT users can access the tool after downloading into PC.

- *Scalability:* MyAppSecurity introduced the tool to provide Enterprise Level Scalability. Threat Modeler has an ability to build and maintain 100s or even 1000s of enterprise-wide applications that reside on different infrastructure stacks over MS-TMT.

- *Collaboration:* Multiple stakeholders can access the Threat Modeler tool and make changes at the same time, and hence provides Real-time Collaboration.

- *Access Control:* MyAppSecurity Threat Modeler assigns role based access and permissions.

- *Integration:* Threat Modeler provides bi-directional integration with other tools, APIs and add-ons.

- *Output:* MS-TMT provides limited specific guidelines for different stakeholders whereas Threat Modeler builds a comprehensive threat profile (Threat Map) of the system which can be either used to understand the system or generate actionable output

– High Value Targets

– Data Flow

– Threats to individual Components

– Risk

– Attack Trees

– List of Mitigation Steps (Abuse Cases)

– Security Assessment Checklist

– List of Total vs. Open Vulnerabilities

- *Re-usability and Repeatability:* Threat Modeler provides the facility to reuse application's threat model components for related threat models, as well as the ability to interrelate between outreaching threat models.

- *Organization's Security Policy:* With the introduction of centralized library in MyAppSecurity Threat Modeler the threats can be link to the enterprise-wide application components and to add new threats as well according to the organization policy.

- *Mapping:* Threat Modeler included security controls and provides the facility to correlate the specific threats mentioned by stakeholders.

- *Secure Coding:* Threat Modeler includes coding guidelines to mitigate the threats encountered in the threat model to the developers.

- *Network Component Hardening:* To secure different network components Threat Modeler automatically provides network component hardening guidelines.

- *Threat Analysis and Threat Comparison:* The stakeholders can compare trends between threats across different applications or can analyse the trends across multiple releases of the same applications in Threat Modeler whereas MS-TMT doesn't have this type of facility.

- *Support:* Threat Modeler Tool Product provides technical support for operational and functional assistance.

- *Creating Threat Model:* To create threat model in MS-TMT the average time is 100-200hrs Time/Resources for one person to build a threat model for a medium size of application whereas Threat Modeler needs 16-24 hrs. Time/Resources to build the same.

- *Platform Independent:* MS-TMT is not a Platform Independent as it is Windows based software whereas Threat Modeler is a Platform Independent as it is web based application.

IV. Conclusion

Threat Modeling is a process to identify security threats to software applications. In this short survey, a comparison of Threat Modeler and MS-TMT has been done on different categories.

In the roadmap for Threat Modeling processes and tooling, there is a scope to research work on the following issue:

Research Topic: Threat Model Measurement [1]

Today it's easy to create threat models for security experts and even no technical through different threat modeling tools. There are some simple assessments we could take like number of elements, redundancy measures or various completeness. To check that our goals like assurance, security analysis and training are being met, to relate between indicators and goals the cost for the measurement, analyzing the models and determining the likelihood that it has iterated.

REFERENCES

[1] Adam Shostack, Microsoft. *Experiences Threat Modeling at Microsoft*.

[2] Marwan Abi-Antoun, Carnegie Mellon University. Daniel Wang, Microsoft Corporation. Peter Torr, Microsoft Corporation. *Checking Threat Modeling Data Flow Diagrams for Implementation Conformance and Security.*

[3] Drake Patrick Mirembe, Faculty of Computing and IT Makerere University Kampala. Maybin Muyeba, Intelligent and Distributed Systems Liverpool Hope University, *Threat Modeling Revisited: Improving Expressiveness of Attack.*

[4] *Suvda Myagmar, Adam J.Lee, William Yurcik.* National Centre for Supercomputing Application (NCSA) University of Illinois at Urbana- Champaign, *Threat Modeling as a Basis for Security Requirements.*

[5] Steven F Burns, GIAC Security Essentials Certification (GSEC) Practical Assignment v1.4c January 5, 2005. *Threat Modeling: A Process to Ensure Application Security.*

[6] Michal Howard and David LeBlanc, ebook- *Writing Secure Code 2nd Edition Microsoft Press.*

[7] Shafiq Hussain1, Asif Kamal2, Shabir Ahmad3, Ghulam Rasool4, Sajid Iqbal5, Department of Computer Science, BahauddinZakariya University, Sub-Campus Sahiwal, Pakistan. *Threat Modeling Methodologies: A Survey*.

[8] Guifre Ruiz, Elisa Heymann, Eduardo Cesar and Barton P. Miller, *Automating Threat Modeling through the Software Development Life-Cycle.*

[9] SANS Institute, InfoSec Reading Room. *Threat Modeling: A Process To Ensure Application Security.*

*[10]*Michal Howard and Steve Lipner Foreword by Jim Allichin Co-president, Platforms & Services Division, Microsoft Corporation. *Best Practice: The Security Development Lifecycle at Microsoft.*

[11]Michael Howard, Senior Principal Cybersecurity Architect, Mark Simos, Cybersecurity Architect, Sean Finnegan, Cybersecurity Director , Vic Miles, Retail Technology Strategy. A Systematic Method to Understand Security Risks in a Retail Environment at Microsoft v1.03

[12]Nick Peterman Vrije University IT Amsterdam, Threat modeling of Enterprise Content Management Systems

[13] Comparing ThreatModeler to Microsoft Threat Modeling Tool (TMT) blog by MyAppSecurity June'3, 2014.
Available: *http://myappsecurity.com/comparing-threatmodeler-microsoft-tmt-threat-modeling-tool/*

[14] SDL Threat Modeling Tool by Microsoft.
Available: *http://www.microsoft.com/security/sdl/adopt/threatmodeling.aspx*

[15] Introducing Microsoft Threat Modeling Tool 2014. Microsoft Team Trustworthy Computing, Microsoft April'15, 2014.
Available: *http://blogs.microsoft.com/cybertrust/2014/04/15/introducing-microsoft-threat-modeling-tool-2014/*

[16] Application Threat Modeling, OWASP.
Available: *https://www.owasp.org/index.php/Application_Threat_Modeling*.