Case study and implementation of threat modelling using MTT

[1]K. Bangaru Krishnaveni, [2]D. Lalitha Bhaskari

[1]MTech CN&IS Student, [2]Professor
Computer Science and System Engineering,
Andhra University College of Engineering
Visakhapatnam,530003, India.

***Abstract-* Any IT company deals with the creation and deployment of web applications. In this era when cybercrime has grown to be a serious menace. Threat Modelling (TM) is one of the key strategies for solving this issue. Threat modelling is locating and disseminating information regarding the dangers that could affect a specific system, network, or application. In this paper provides comprehensive view of threat modelling, TM approaches, and the many tools that are accessible. To create secure web applications, using Microsoft Threat Taxonomy (MTT) tool implements stride methodology. The STRIDE method is a popular one for threat modelling.it stands for spoofing, tampering, repudiation, information disclosure, denial of service, elevation of privilege. It examines the value of threat modelling in the current cybersecurity environment and how it aids businesses in identifying and addressing possible threats. For a secure design of online applications, MTT explains the approach to identify threats, threat boundaries, and mitigation actions. This essay serves as a case study for the application of the stride technique to MTT, outlining the procedures followed, the difficulties encountered, and the results obtained. The case study's conclusions and recommendations are explored, along with the best practices for efficient threat modelling. Overall, the goal of this work is to further the field of exploratory data analytics by offering suggestions and helpful advice on how to integrate threat modelling into the software development cycle.**

By doing this, the MTT tool's security was improved, guaranteeing that the necessary defences were put in place to lessen the threats that were discovered.

Keywords: **Threat Modelling, STRIDE methodology, MTT tool, security, vulnerabilities, threats.**

Introduction and Background:

A critical step in guaranteeing the security of a system or application is threat modelling. It entails spotting and countering possible threats that can take advantage of weak spots in the system. The STRIDE methodology is one extensively used strategy in threat modelling. The six areas of threats that are the emphasis of the stride technique are spoofing, tampering, repudiation, information leakage, denial of service, and elevation of privilege. Potential risks and vulnerabilities can be found by assessing the system and taking into account these categories; then, the proper countermeasures can be put into place. In this study, the STRIDE methodology was used to model potential threats to the MTT tool. By spotting and removing potential risks, the MTT tool's security was to be improved. The team was able to thoroughly assess the product and put the appropriate security safeguards in place by adhering to the STRIDE approach.

Literature survey:

With the increasing importance of threat modelling in Software Applications, many researchers have done their research in this area.

Laorden [1] On-line Social Networks (OSN) have been the subject of this study paper's effort. It is the first threat modeling strategy used in online social networks with the goal of identifying threats and exploitable weaknesses. The Circle of Risk (CoR), a pictorial definition of each security component engaged in the threat modeling, is defined next. Although privacy issues are starting to surface, there are still other extremely risky flaws that compromise security and endanger the assets of businesses and people.

Khan [2] A STRIDE Model based Threat Modeling utilizing Unified and-Or Fuzzy Operator for Computer Network Security has been worked on in this research article. The level of attack can be determined using a strategy that uses fuzzy logic-based threat assessment to evaluate the quantity and types of attacks as input. In the work that is being described, a fuzzy operator called the universal AND-OR operator (UAO operator) and a method of making decisions based on fuzzy rules are used.

Nagaas [3] In this study, a threat-driven approach to modeling campus network security was used. It is a collection of methodologies, techniques, and tools combined into a single framework or model. In this study, the STRIDE and DREAD are also employed to locate and estimate potential network security vulnerabilities.

Gattiker [4] This study report uses the STRIDE-model and focuses on the hospital management information system. The STRIDE method's findings demonstrate that a number of threats have been found, including one on the user side, five on the web server, and three on the database. From the lowest level of threat (Low L) to the greatest level (Hill), there are various levels of threat. Starting from the Low L to the Hill and strengthening the security system at SIMRS according to the threat level can serve as a guidance.

Vallant [5] Work has been done on Cyber security requirements engineering for low voltage distribution smart grid designs using Threat Modelling in this research report. They have contributed to this difficulty by developing a methodology for assessing the risk of cyber-attacks in SG systems. They have developed a threat model and have identified potential weaknesses in low-voltage distribution grids. Then, exploitation probability was estimated using realistic attack scenarios.

Camacho [6] An empirical study was used to conduct research on Agile team members' perspectives of non-functional testing affecting elements in this research report. They carried out a case study in a company with five agile development projects. They discovered 21 difficulties to threat modeling as a result of their observations. This study demonstrates that numerous obstacles must still be addressed in order for threat modeling to be properly adopted in agile development initiatives.

Overview of Threat modelling:

Practice of identifying and mitigating potential dangers to a system or application is known as threat modeling. It entails examining the architecture, design, and functionality of the system to detect potential vulnerabilities and threats. The purpose is to address security issues proactively and adopt suitable solutions.

Threat modeling is included into the software development life cycle early in the design phase. It assists in identifying potential security risks during the development or deployment stages, decreasing the chance of costly corrections later on. Organizations can construct more secure and robust systems by adding threat modeling into the software development life cycle.

Purpose of threat modelling:

The goal of threat modeling is to detect, assess, and prioritize potential risks in a methodical manner. It assists companies in understanding the security posture of their systems and making informed decisions about how to fix vulnerabilities. Organizations may effectively allocate resources, implement appropriate security policies, and decrease overall risk to their systems by spotting threats early on.

Benefits of threat modelling:

The advantages of threat modeling are numerous. It enables enterprises to:

1. Identify potential vulnerabilities and threats early in the development process.
2. Prioritize security efforts and deploy resources wisely.
3. Make sound security and regulatory decisions. Controls and countermeasures.
4. Improve the overall security posture of their systems.
5. Lower the risk of security breaches and the associated expenditures.
6. Increase compliance with security standards.

Organizations can proactively address security risks and design more secure and resilient systems by including threat modeling into the software development life cycle.

Threat modelling methodologies and techniques

There are several approaches you can employ while undertaking threat modeling. What kinds of dangers you're aiming to depict and for what reason will determine the best model for your purposes.

STRIDE threat modelling

A threat model called STRIDE was developed by Microsoft engineers to help in the process of finding dangers in a system. It works in conjunction with a model of the intended system. It is therefore most useful for assessing specific systems.

STRIDE stands for the threats it addresses, which are:

(i) Spoofing, in which a user or program impersonates another
(ii) Information disclosure — data is leaked or exposed
(iii) Tampering — attackers alter components or code
(iv) Repudiation — threat events are not logged or monitored
(v) Denial of service (DoS) — services or components are overloaded with traffic to prevent legitimate use
(vi) Elevation of Privilege — attackers give themselves more privileges to gain control over a system

Process for Attack Simulation and Threat Analysis (PASTA)

PASTA is a seven-step approach focused on attackers. It is intended to link technical requirements and business objectives. Teams can dynamically discover, count, and prioritize risks using PASTA's steps.

A PASTA threat model has the following steps:

1. Establish business goals
2. Specify the technical range of the components and assets
3. Decomposing an application and identifying its controls
4. Threat evaluation based on threat information
5. Finding vulnerabilities
6. Modeling and enumeration of attacks
7. Risk analysis and the creation of protective measures

Common Vulnerability Scoring System (CVSS)

A standardized threat scoring method for identified vulnerabilities is called CVSS. The Forum of Incident Response and Security Teams (FIRST) maintains it. It was created by the National Institute of Standards and Technology (NIST). This system is intended to aid security teams in threat analysis, impact analysis, and countermeasure identification. Additionally, it enables security experts to accurately evaluate and use threat intelligence created by others. The influence of the risk factor owing to the passage of time since the vulnerability was initially identified is taken into account by CVSS, along with the threat's fundamental characteristics.

Additionally, it has provisions that enable security teams to specifically alter risk assessments in accordance with different system setups.

Visual, Agile, and Simple Threat (VAST)

The automated threat modeling technique known as Visual, Agile, and Simple Threat (VAST) was developed on the Threat Modeler platform. To produce accurate, usable data and preserve scalability, large businesses employ VAST across their whole infrastructure. VAST can be integrated into the DevOps lifecycle and assist teams in recognizing various operational and infrastructural issues. Two different threat models must be created in order to use VAST:
(i) An application threat model - depicts the threat's architectural component using a process-flow diagram.
(ii)The operational threat model - visualizes the danger from the attacker's point of view using a data-flow diagram.

Trike

Trike is a framework for security audits that uses threat modeling techniques to manage risk and defense. Trike describes a system, and an analyst creates a requirement model by listing the system's resources, actors, rules, and actions. A step matrix created by Trike has rows for the actors and columns for the assets. Each matrix cell includes four components that correspond to potential activities (create, read, update, and delete), as well as a rule tree where the analyst determines whether a particular operation is permitted, prohibited, or permitted.
With the criteria established, Trike creates a data-flow diagram that maps each element to the proper assets and actors. The diagram is used by the analyst to spot threats like as privilege escalation and denial of service (DoS). Trike rates the likelihood of an assault on each CRUD activity and actor on a five-point scale. Additionally, it rates actors according to the degree of permission they have for each activity (always, occasionally, or never).

Attack Trees

Attack trees are diagrams that show the possible routes that attacks in a system can follow. These diagrams show assault objectives as a root and potential routes as branches. Several trees are built for a single system while doing threat modeling, one for each attacker aim.
One of the earliest and most popular threat modeling methodologies is this one. The combination of PASTA, CVSS, and STRIDE is now commonplace, whereas it was originally used alone.

Security Cards

Instead of conventional threat modeling methodologies, the Security Cards methodology is built on brainstorming and original thought. It is intended to assist security teams in planning for less frequent or unusual assaults. This methodology is a great way for security teams to learn more about threats and approaches to threat modeling.
The approach makes use of a deck of 42 cards that aid analysts in providing answers to hypothetical questions concerning potential attacks, such as who may launch one, what might drive them, which systems they might target, and how they might carry one out. To mimic potential attacks and go through the organization's possible responses, analysts can deal the cards in a sort of tabletop game.

Hybrid Threat Modelling Method (HTMM)

Security Equipment Inc. (SEI) created the HTMM methodology, which combines two other methodologies:
(i) Security Quality needs Engineering (SQUARE), a technique for gathering, classifying, and ranking security needs.
(ii)Persona non Grata (PnG), is a technique that focuses on finding ways a system might be misused to serve the objectives of an attacker.
Threat modeling that incorporates all potential threats, yields zero false positives, delivers reliable results, and is reasonably priced is made possible by HTMM.

Understanding the stride methodology:

A technique to integrate earlier in your software development lifecycle (SDLC) is STRIDE threat modelling. The STRIDE framework is used to map out your application based on its distinct use cases and business logic as a threat modelling tool. As a result, it can be used to find and fix potential flaws before writing a single line of code. When releasing new code and at any moment while your application is in development or production, you can refer back to the STRIDE framework to determine how it will impact the entire attack vector. Building networks, systems, and applications that are safe by design should start with the use of threat modelling.

STRIDE as a framework for Threat modelling:

Koren Kohn Felder and Praerit Garg, two Microsoft engineers, created STRIDE in the late 1990s. They discussed the increased security risks to systems brought on by developing technologies in their letter titled "The Threats To Our Products," and they came to the conclusion that a method for pinpointing such dangers was necessary. Six different threat categories are taken into account by STRIDE, including:

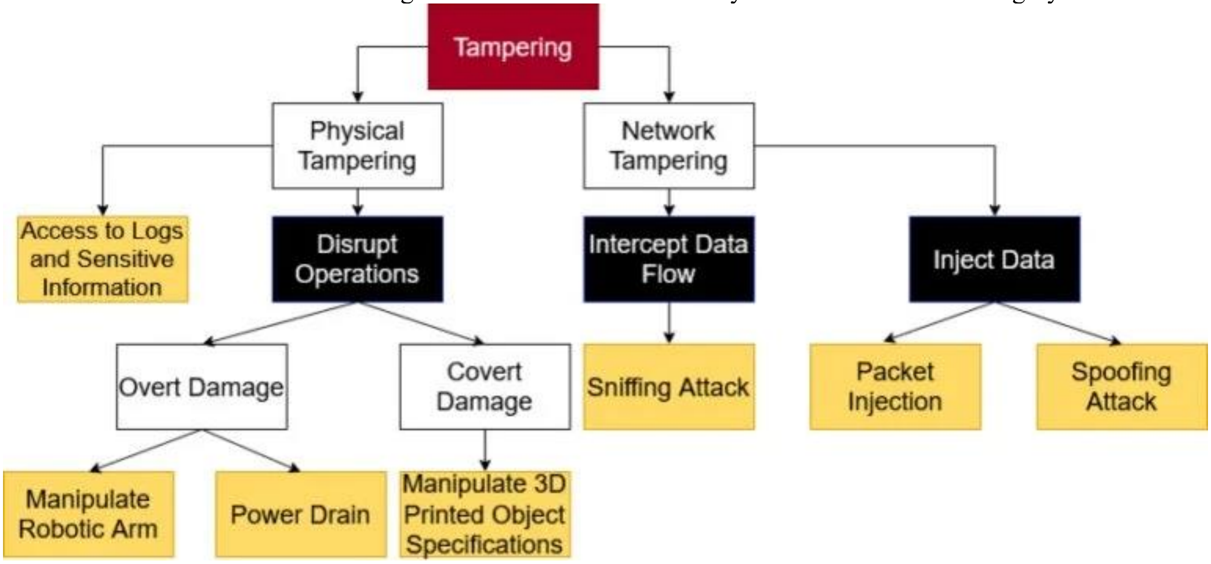
Spoofing identity

When a hacker impersonates someone else and uses their identity and the information contained in it to conduct fraud, this is known as identity spoofing. When an email is sent from a fake email address while purporting to be from someone else, this hazard is fairly

common and is known as a "phishing attack." Usually, these emails ask for private information. When a gullible or uninformed recipient delivers the desired information, the hacker can quickly take the new identity.
Both human and technical identities can be used to commit identity fraud. Spoofing allows the hacker to obtain access using just one weak identity and launch a far more significant cyberattack. Artificial intelligence (AI) is advancing at a rapid pace, making automated phishing tools more convincing than ever.
Among the ways AI engages in phishing are
(i)emails or messages that try to persuade recipients to click on harmful links
(ii)social opposition trolls that aim to harm a brand's reputation
(iii)and fake news websites and social media pages.

Tampering with data

When data or information is altered without consent, this is known as data tampering. A hostile actor may carry out tampering by adding a malicious file, deleting or altering a log file, or changing a configuration file to take control of the system. To determine whether and when data manipulation occurs, it is crucial to include change monitoring, also known as file integrity monitoring (FIM), into your organization. By starting with a baseline of what a "good" file looks like, this technique evaluates files critically. For file monitoring to work well, logging and storing must be done properly. To learn about the dangers of insufficient or excessive logging and auditing, read the Security Playbook here.
The sample tampering attack tree (another threat modeling activity) of a 3D concrete printing system is shown in the image below. Picture taken from the article Threat Modelling in Construction: A Case Study of a 3D Concrete Printing System.



Repudiation Threats

Threats of repudiation occur when a bad actor engages in an unlawful or malicious action within a system and then claims they had nothing to do with the attack. In these attacks, the system is unable to track the malicious behavior to pinpoint the perpetrator. Due to the fact that very few systems check outbound mail for legitimacy, repudiation attacks are comparatively simple to carry out on email systems. These assaults typically start out as access attacks.

Information disclosure

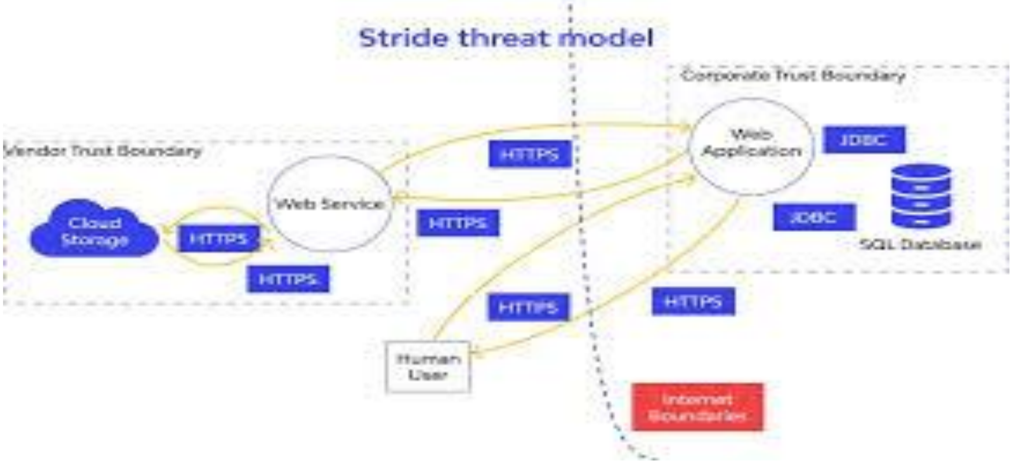
Information disclosure and information leakage are synonyms. It occurs when a website or program mistakenly makes data accessible to unauthorized users. This kind of threat can have an impact on an application's workflow, data storage, and data flow. Information disclosure can take many forms, such as unintended access to source code files through temporary backups, the needless exposure of sensitive data like credit card details, and the inclusion of database information in error messages. These problems are frequent and can be caused by openly shared internal content, unsecured application setups, or inadequate error handling in the application's design.

Denial of service (Dos)

Attacks known as denial of service (DoS) prevent a legitimate user from using resources that they are supposed to have access to. This has an impact on an application's workflow, data storage, and data processing. DoS attacks are growing in size and frequency, with 12.5 million DDoS weapons expected to be discovered in 2020. According to the State of Penetration Testing as a Service study for 2022, the incidence of DoS assaults rose by 133% in 2017.
In 2017, one of the most well-known attacks targeted Google. The attacker exploited multiple networks to fake 167 Mpps (millions of packets per second) to 180,000 exposed CLDAP, DNS, and SMTP servers, which would subsequently return massive answers to us, according to Google's announcement. This depicts the levels an attacker with resources can reach: This was four times bigger than the 623 Gbps attack from the Mirai botnet a year earlier, which set a record. Although DoS attacks have increased, it appears that defence mechanisms like AWS Shield and Cloudflare are still working well.

Elevation of privileges

An authorized or unauthorized user within the system can acquire access to other data that they are not entitled to see through the elevation of privileges. A missing authorization check or even elevation through data manipulation, in which the attacker alters the disk or memory to execute unauthorized commands, could serve as examples of this assault.



By methodically analysing each element of the STRIDE technique and finding potential vulnerabilities and solutions, MTT offers a comprehensive framework for threat modelling. Prioritizing security initiatives and providing a strong Défense against numerous threats are made possible by it.

Introduction to MTT

MTT, or Microsoft Threat Taxonomy, is a comprehensive framework for threat modelling that works in conjunction with the STRIDE approach. By dividing risks into many threat categories, MTT offers a structured and organized method for categorizing and dealing with threats.

Threat modelers can more easily prioritize and address risks by using MTT to discover and analyse threats in a systematic way. Threat modelling is more standardized and effective as a result of the shared vocabulary and taxonomy it provides for discussing and describing threats.

MTT improves the STRIDE technique by offering a more comprehensive and in-depth framework for categorizing threats. MTT broadens this scope by integrating additional threat categories and subcategories, whereas STRIDE focuses on six distinct threat categories. Threat modelers can now take into account a greater variety of potential threats and vulnerabilities.

Threat modelers can have a more thorough grasp of the dangers their systems may face by combining the STRIDE technique with MTT. They are able to analyse every danger category and subcategory, find potential weak spots, and set mitigation priorities accordingly. This all-encompassing approach aids in the development of resilient and secure systems.

In conclusion, MTT enhances STRIDE approach by offering a thorough framework for classifying and managing hazards. It broadens the application of threat modeling and improves the efficiency of the process of analysis and mitigation.

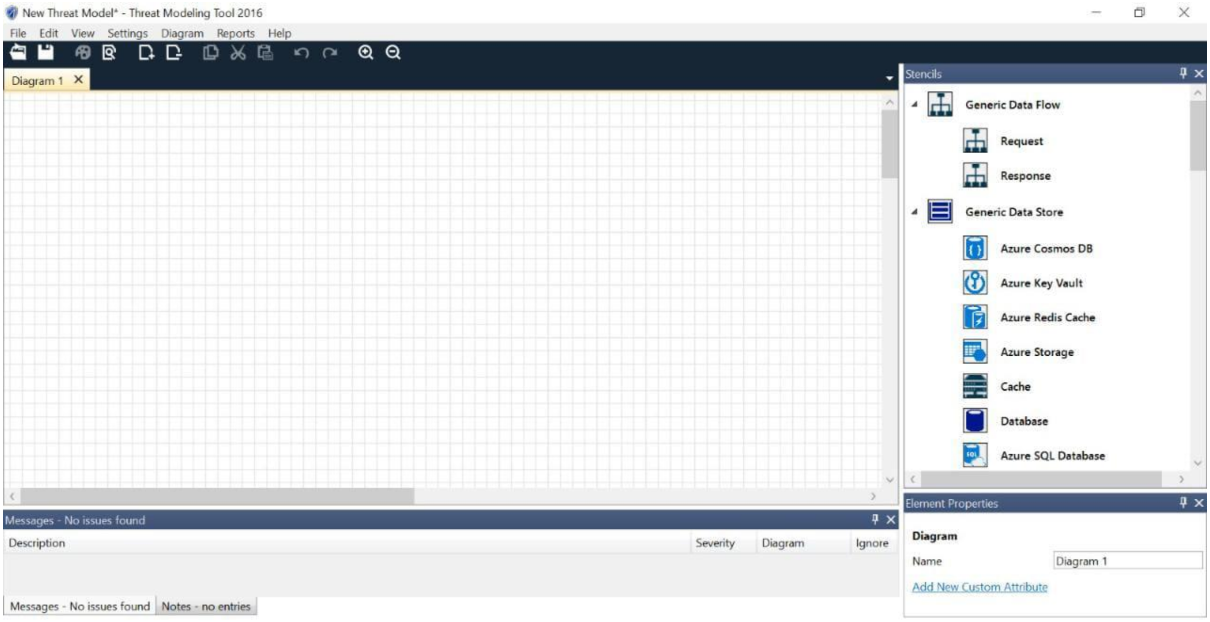
Applying STRIDE to MTT:

Threat modelling using MTT can be done using the STRIDE technique.

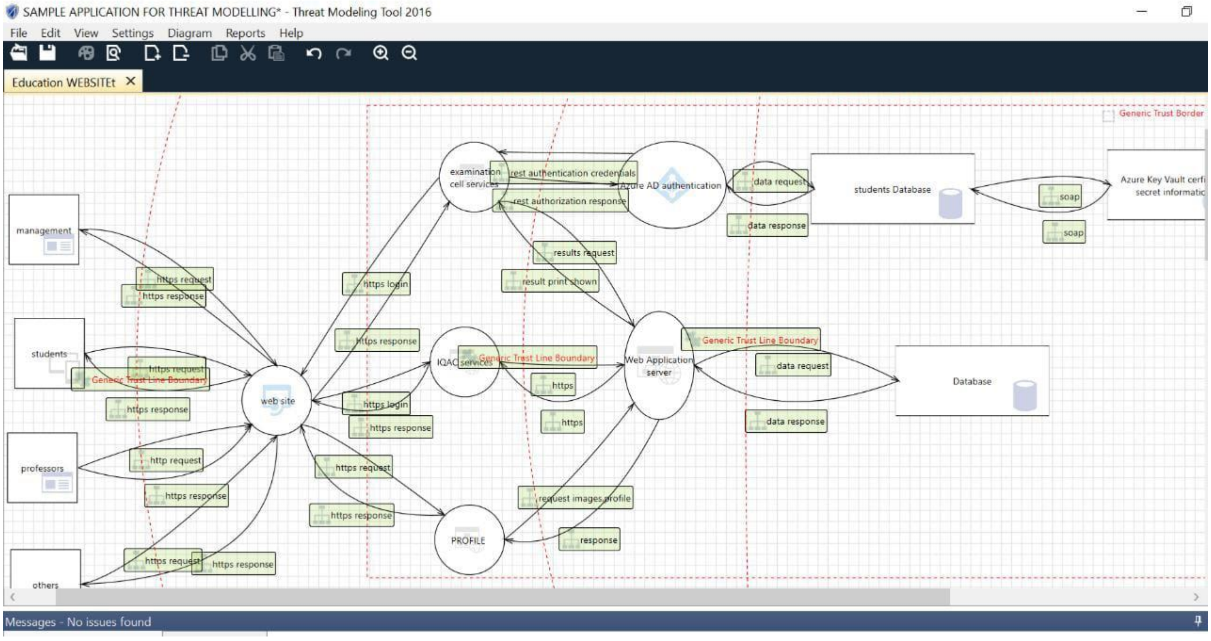
Step1: Determine the components of the system and how they interact.

-Understanding the system's architecture and design, including its parts, interfaces, and data flows, is required for this.

-You can draw your diagram using the design view, which also offers objects and characteristics to help you accurately depict the component's design.



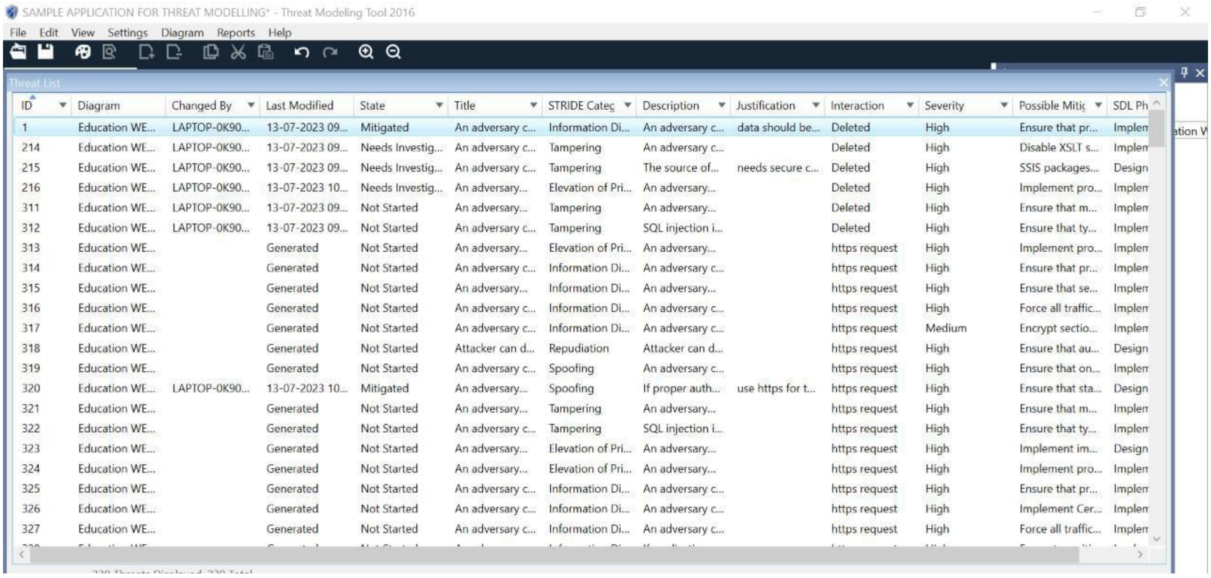
All diagrams must at a minimum include:
(i) A procedure or processes.
(ii) The directional data flows within and between the processes as well as between external interactors.
Important data repositories (iii).
(iv) A third party that interacts with the processes externally, frequently a user.
(v) A trust limit or limits



Sample university application

Step 2: Treat each component using the STRIDE approach.

- Based on the STRIDE categories, evaluate possible hazards for each component: - Spoofing: Look out for any instances of impersonation or unauthorized access.
- Tampering: Take into account the likelihood of unintentional data modification or alteration.
- Repudiation: Evaluate the capacity to reject or contest deeds or occurrences.
- Information disclosure: Assess the possibility of unauthorized access to private data.
- Denial of service: Look for any weaknesses that can cause a service interruption.
- Privilege escalation: Take into account any chance of an unlawful privilege escalation.

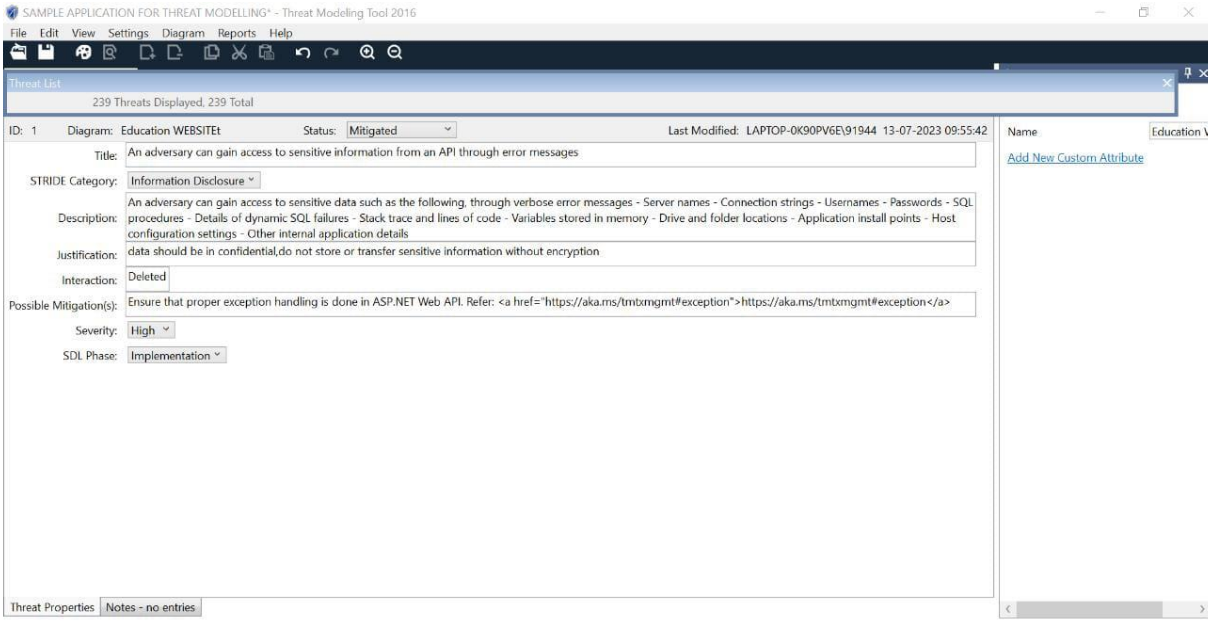

SAMPLE APPLICATION FOR THREAT MODELLING* - Threat Modeling Tool 2016
File Edit View Settings Diagram Reports Help
Threat List

ID	Diagram	Changed By	Last Modified	State	Title	STRIDE Categ	Description	Justification	Interaction	Severity	Possible Mitig	SDL Ph
1	Education WE...	LAPTOP-0K90...	13-07-2023 09...	Mitigated	An adversary c...	Information Di...	An adversary c...	data should be...	Deleted	High	Ensure that pr...	Implem
214	Education WE...	LAPTOP-0K90...	13-07-2023 09...	Needs Investig...	An adversary c...	Tampering	An adversary c...		Deleted	High	Disable XSLT s...	Implem
215	Education WE...	LAPTOP-0K90...	13-07-2023 09...	Needs Investig...	An adversary c...	Tampering	The source of...	needs secure c...	Deleted	High	SSIS packages...	Design
216	Education WE...	LAPTOP-0K90...	13-07-2023 10...	Needs Investig...	An adversary...	Elevation of Pri...	An adversary...		Deleted	High	Implement pro...	Implem
311	Education WE...	LAPTOP-0K90...	13-07-2023 09...	Not Started	An adversary...	Tampering	An adversary...		Deleted	High	Ensure that m...	Implem
312	Education WE...	LAPTOP-0K90...	13-07-2023 09...	Not Started	An adversary c...	Tampering	SQL injection i...		Deleted	High	Ensure that ty...	Implem
313	Education WE...		Generated	Not Started	An adversary...	Elevation of Pri...	An adversary...		https request	High	Implement pro...	Implem
314	Education WE...		Generated	Not Started	An adversary c...	Information Di...	An adversary c...		https request	High	Ensure that pr...	Implem
315	Education WE...		Generated	Not Started	An adversary...	Information Di...	An adversary...		https request	High	Ensure that se...	Implem
316	Education WE...		Generated	Not Started	An adversary c...	Information Di...	An adversary c...		https request	High	Force all traffic...	Implem
317	Education WE...		Generated	Not Started	An adversary c...	Information Di...	An adversary c...		https request	Medium	Encrypt sectio...	Implem
318	Education WE...		Generated	Not Started	Attacker can d...	Repudiation	Attacker can d...		https request	High	Ensure that au...	Design
319	Education WE...		Generated	Not Started	An adversary c...	Spoofing	An adversary c...		https request	High	Ensure that on...	Implem
320	Education WE...	LAPTOP-0K90...	13-07-2023 10...	Mitigated	An adversary...	Spoofing	If proper auth...	use https for t...	https request	High	Ensure that sta...	Design
321	Education WE...		Generated	Not Started	An adversary...	Tampering	An adversary...		https request	High	Ensure that m...	Implem
322	Education WE...		Generated	Not Started	An adversary c...	Tampering	SQL injection i...		https request	High	Ensure that ty...	Implem
323	Education WE...		Generated	Not Started	An adversary...	Elevation of Pri...	An adversary...		https request	High	Implement im...	Design
324	Education WE...		Generated	Not Started	An adversary...	Elevation of Pri...	An adversary...		https request	High	Implement pro...	Implem
325	Education WE...		Generated	Not Started	An adversary c...	Information Di...	An adversary c...		https request	High	Ensure that pr...	Implem
326	Education WE...		Generated	Not Started	An adversary c...	Information Di...	An adversary c...		https request	High	Implement Cer...	Implem
327	Education WE...		Generated	Not Started	An adversary c...	Information Di...	An adversary c...		https request	High	Force all traffic...	Implem

Analysis view

Step 3: Using MTT, classify and rank the threats that have been found.

- Classify the discovered risks into distinct threat categories and subcategories using the MTT framework.
- Give the threats a priority ranking based on their seriousness and probable system impact.


SAMPLE APPLICATION FOR THREAT MODELLING* - Threat Modeling Tool 2016
File Edit View Settings Diagram Reports Help
Threat List
Threat Properties
Notes - no entries
Add New Custom Attribute

239 Threats Displayed, 239 Total

ID: 1 Diagram: Education WEBSITEt Status: Mitigated Last Modified: LAPTOP-0K90PV6E\91944 13-07-2023 09:55:42

Title: An adversary can gain access to sensitive information from an API through error messages

STRIDE Category: Information Disclosure

Description: An adversary can gain access to sensitive data such as the following, through verbose error messages - Server names - Connection strings - Usernames - Passwords - SQL procedures - Details of dynamic SQL failures - Stack trace and lines of code - Variables stored in memory - Drive and folder locations - Application install points - Host configuration settings - Other internal application details

Justification: data should be in confidential,do not store or transfer sensitive information without encryption

Interaction: Deleted

Possible Mitigation(s): Ensure that proper exception handling is done in ASP.NET Web API. Refer: <a href="https://aka.ms/tmtxmgmt#exception">https://aka.ms/tmtxmgmt#exception</a>

Severity: High

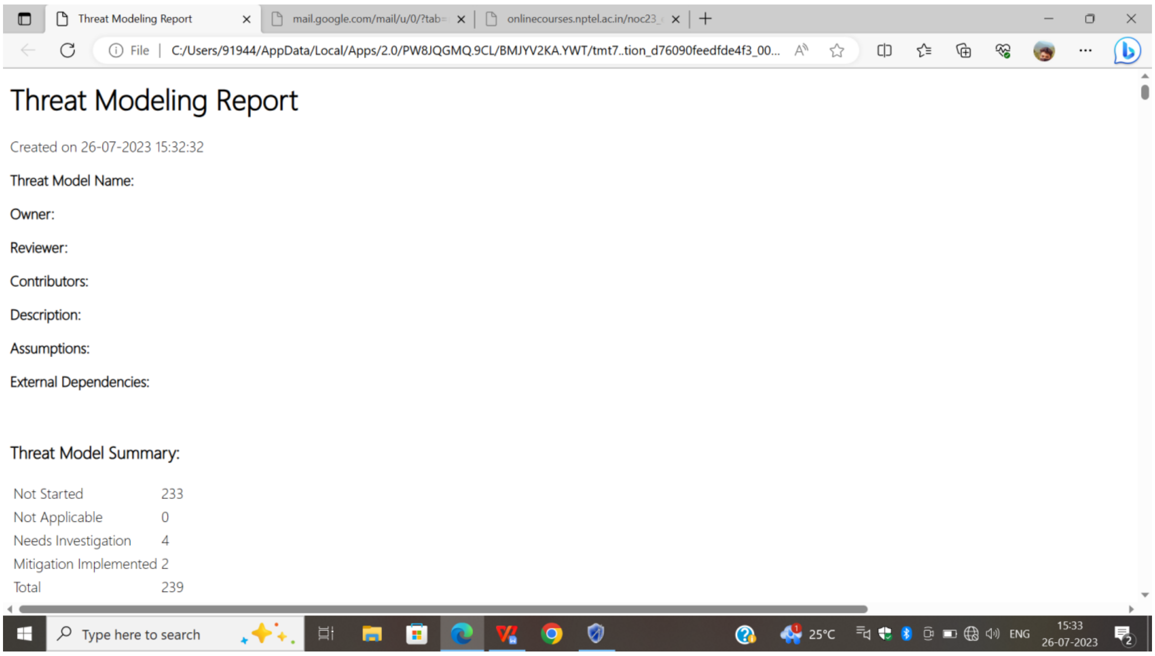
SDL Phase: Implementation

Name: Education \

Analysis view

Step 4: Reduce the threats that have been found.

- Create effective defenses and mitigation plans for each uncovered threat.
- Implement security measures like input validation, access controls, encryption, and secure coding techniques. During this procedure, there may be difficulties with: - Accurately identifying all system components and their relationships.
- Ensuring thorough protection against potential threats.
- Balancing the ranking of threats according to their impact and seriousness.
- Putting into action sensible and efficient mitigation measures.


Threat Modeling Report
mail.google.com/mail/u/0/?tab=
onlinecourses.nptel.ac.in/noc23_
File | C:/Users/91944/AppData/Local/Apps/2.0/PW8JQGMQ.9CL/BMJYV2KA.YWT/tmt7..tion_d76090feedfde4f3_00...
Type here to search
25°C
ENG
15:33
26-07-2023

Threat Modeling Report

Created on 26-07-2023 15:32:32

Threat Model Name:

Owner:

Reviewer:

Contributors:

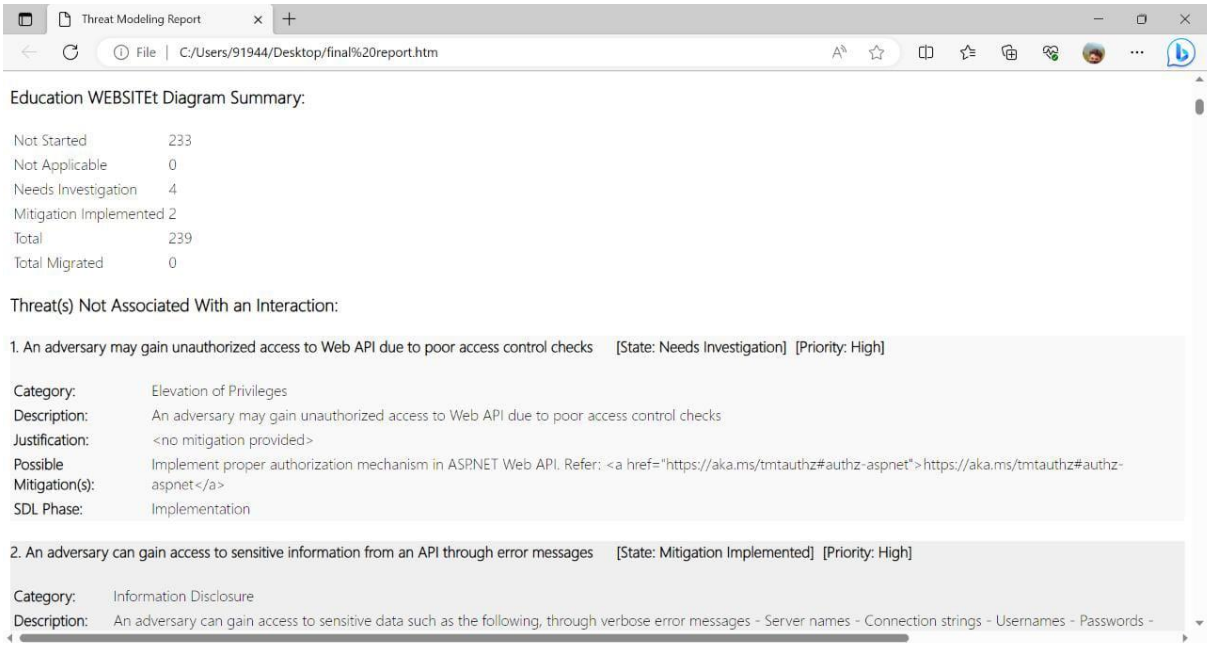
Description:

Assumptions:

External Dependencies:

Threat Model Summary:

Not Started	233
Not Applicable	0
Needs Investigation	4
Mitigation Implemented	2
Total	239



Education WEBSITEt Diagram Summary:

Not Started	233
Not Applicable	0
Needs Investigation	4
Mitigation Implemented	2
Total	239
Total Migrated	0

Threat(s) Not Associated With an Interaction:

1. An adversary may gain unauthorized access to Web API due to poor access control checks [State: Needs Investigation] [Priority: High]

Category:	Elevation of Privileges
Description:	An adversary may gain unauthorized access to Web API due to poor access control checks
Justification:	<no mitigation provided>
Possible Mitigation(s):	Implement proper authorization mechanism in ASP.NET Web API. Refer: <a href="https://aka.ms/tmtauthz#authz-aspnet">https://aka.ms/tmtauthz#authz-aspnet</a>
SDL Phase:	Implementation

2. An adversary can gain access to sensitive information from an API through error messages [State: Mitigation Implemented] [Priority: High]

Category:	Information Disclosure
Description:	An adversary can gain access to sensitive data such as the following, through verbose error messages - Server names - Connection strings - Usernames - Passwords -

Threat modelling Report

Results and findings:

I found a number of vulnerabilities and dangers in case study and STRIDE technique deployment on MTT for the university online application. There was a chance of spoofing attacks, altering with test results data, repudiating activities, information exposure, denial-of-service attacks, and privilege elevation.

We took numerous security procedures to mitigate these issues. Data integrity checks were put in place to identify manipulation, two-factor authentication was added to avoid spoofing attacks, and audit logs were introduced to guarantee non-repudiation. In order to prevent data leakage and to secure student and management data from denial-of-service attacks, encryption, load balancing, and redundancy were used. Role-based access control was added to the system to stop unauthorized privilege elevation.

Overall, the implemented security measures were successful in reducing the risks and vulnerabilities found. They contributed to strengthening the university online application's security posture by guaranteeing the privacy, accuracy, and accessibility of management and student data.

The university online application was able to provide a more secure environment for its customers by adhering to the STRIDE approach and putting in place suitable security measures, which decreased the possibility of successful assaults and protected sensitive data.

STRIDE Threat & Mitigation Techniques

Threat Type	**Mitigation Techniques**
Spoofing Identity	1.Appropriate-authentication 2.Protect-secretdata 3. Don't store secrets
Tampering with data	1.Appropriate-authorization 2.Hashes 3.MACs 4.Digital-signatures 5. Tamper resistant protocols
Repudiation	1.Digita-lsignatures 2.Timestamps 3. Audit trails
Information Disclosure	1.Authorization 2.Privacy-enhanced-Protocols 3.Encryption 4.Protectsecrets 5. Don't store secrets
Denial of Service	1.Appropriate-authentication 2.Appropriate-authorization 3.Filtering 4.Throttling 5. Quality of service
Elevation of privilege	1. Run with least privilege

Best practices and suggestions:
I have a few suggestions and best practices for enhancing an organization's security posture through efficient threat modeling based on my experience with using STRIDE and MTT:
1. Start moving now.
2. Include the appropriate stakeholders: To achieve a thorough understanding of the system and its possible dangers, include representatives from several teams, such as developers, architects, security specialists, and business owner analyze potential threats, making sure that no crucial areas are missed.
3. Take a methodical approach: Use a structured technique, such as STRIDE and MTT, to methodically identify and aly Utilize threat modeling early on in the development lifecycle to recognize and mitigate potential security concerns.
4. Prioritize mitigation activities by evaluating the gravity and impact of each identified hazard.
5. Constantly revise and update: Threat modeling is a process that is iterated upon. As the system develops, fresh threats appear, or the system architecture changes, it is important to periodically assess and update threat models.
6. Record findings and spread them: Keep thorough records of the threat modeling procedure, detected threats, and suggested mitigation techniques. To guarantee knowledge and coordination within the organization, share this information with the appropriate parties.

Organizations can improve their security posture by proactively recognizing and mitigating possible risks, lowering the risk of security events, by adhering to these suggestions and best practices.

Conclusion and future work:

Conclusion:

Threat modeling is a crucial process for identifying and mitigating potential security risks in software systems. The STRIDE methodology provides a structured approach for identifying and analyzing threats based on six categories: Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege. The MTT (Microsoft Threat Modeling Tool) is a valuable tool for visualizing and documenting threat models, aiding in the identification and mitigation of potential risks. Ongoing threat modeling efforts are essential to keep pace with evolving threats and changes in the system architecture. Collaboration and communication among stakeholders are vital for a comprehensive understanding of the system and effective threat mitigation. Documentation and sharing of threat modeling findings help raise awareness and ensure alignment across the organization.

Future work:

It is crucial to investigate the following areas for future research:

1. To increase efficiency and effectiveness, threat modeling methods should be automated and integrated into the software development lifecycle.
2. Using machine learning and threat intelligence to better identify threats and develop mitigation plans.
3. Evaluation and comparison of various threat modeling approaches to identify their advantages and disadvantages in various situations.
4. Examining how threat modeling approaches are affected by developing technologies including cloud computing, the Internet of Things (IoT), and artificial intelligence.

To remain ahead of changing threats and guarantee the security of software systems, ongoing threat modeling efforts and continual research in this area are essential.

REFERENCES:

[1] Carlos Laorden, Borja Sanz, Gonzalo Alvarez, Pablo G. Bringas.” A Threat Model Approach to Threats and Vulnerabilities in On-line Social Networks”. International Conference on Computational Intelligence in Security for Information Systems (CISIS'10), León, Spain, November 11-12, 2010.

[2] Salman A. Khan,” A STRIDE Model based Threat Modelling using Unified and-Or Fuzzy Operator for Computer Network Security”. International journal of computing and network technology, SSN (2210-1519),2017.

[3] Marlon A. Naagas,” A Threat-Driven Approach to Modelling a Campus Network Security”. Conference: the 6th International Conference,2018.

[4] Gattiker, U.E.: “Hospital management information system using STRIDE-Model. (Kluwer international series in engineering and computer science). Kluwer Academic Publishers, Norwell, MA, USA (2004).

[5] Stefan Marksteiner, Heribert Vallant, Kai Nahrgang.” Cyber security requirements engineering for low voltage distribution smart grid architectures using threat modelling”. Journal of information security and applications, Vol-6, id 102389,2019.

[6] C. R. Camacho, S. Marczak, and D. S. Cruzes, “Agile team members perceptions on non-functional testing influencing factors from an empirical Study,” in Proceedings of the 11th International Conference on Availability, Reliability and Security, ARES 2016, 2016, pp. 582–589.