St. Francis Institute of Technology
(An Autonomous Institution)
AICTE Approved | Affiliated to University of Mumbai

A+ Grade by NAAC: CMPN, EXTC, INFT NBA Accredited: ISO 9001:2015 Certified

Department of Information Technology

A.Y. 2025-2026
Class: BE-IT A/B, Semester: VII
Subject: Secure Application Development Lab

Student Name: **Keith Fernandes** Student Roll No: **25**

Experiment – 3: Study and exercise on Threat Modeling

Aim: To study and excise on Threat Modeling

Objective: After performing the experiment, the students will be able to –

- To generate Threat model using Microsoft Threat modeling tool
- To get familiar with the features provided by the tool
- To Learn about generated threat categories
- To find mitigations for the generated threats

Lab objective mapped: To understand the methodologies and standards for developing secure code

Prerequisite: Basic knowledge Information Security, software engineering

Requirements: Personal Computer, Windows operating system browser, Internet Connection etc.

Pre-Experiment Theory:

What is threat modeling?
Threat modeling is a structured process with these objectives: identify security requirements, pinpoint security threats and potential vulnerabilities, quantify threat and vulnerability criticality, and prioritize remediation methods.

Threat modeling methods create these artifacts:

- An abstraction of the system
- Profiles of potential attackers, including their goals and methods
- A catalog of threats that could arise

Threat modeling works by identifying the types of threat agents that cause harm to an application or computer system. It adopts the perspective of malicious hackers to see how much damage they could do. When conducting threat modeling, organizations perform a thorough analysis of the software architecture, business context, and other artifacts (e.g., functional specifications, user documentation). This process enables a deeper understanding and discovery of important aspects of the system. Typically, organizations conduct threat modeling during the design stage (but it can occur at other stages) of a new application to help developers find vulnerabilities and become aware of the security implications of their design, code, and configuration decisions. Generally, developers perform threat modeling in four steps:

- **Diagram.** What are we building?
- **Identify threats.** What could go wrong?
- **Mitigate.** What are we doing to defend against threats?
- **Validate.** Have we acted on each of the previous steps?

Discuss various threat categories and mitigations mentioned by Microsoft threat modeling tool

Microsoft uses the STRIDE model categorizes different types of threats and simplifies the overall security conversations.

Category	Description
Spoofing	Involves illegally accessing and then using another user's authentication information, such as username and password
Tampering	Involves the malicious modification of data. Examples include unauthorized changes made to persistent data, such as that held in a database, and the alteration of data as it flows between two computers over an open network, such as the Internet
Repudiation	Associated with users who deny performing an action without other parties having any way to prove otherwise—for example, a user performs an illegal operation in a system that lacks the ability to trace the prohibited operations. Non-Repudiation refers to the ability of a system to counter repudiation threats. For example, a user who purchases an item might have to sign for the item upon receipt. The vendor can then use the signed receipt as evidence that the user did receive the package
Information Disclosure	Involves the exposure of information to individuals who are not supposed to have access to it—for example, the ability of users to read a file that they were not granted access to, or the ability of an intruder to read data in transit between two computers
Denial of Service	Denial of service (DoS) attacks deny service to valid users—for example, by making a Web server temporarily unavailable or unusable. You must protect against certain types of DoS threats simply to improve system availability and reliability
Elevation of Privilege	An unprivileged user gains privileged access and thereby has sufficient access to compromise or destroy the entire system. Elevation of privilege

Category	Description
	threats include those situations in which an attacker has effectively penetrated all system defenses and become part of the trusted system itself, a dangerous situation indeed

Mitigation categories

The Threat Modeling Tool mitigations are categorized according to the Web Application Security Frame, which consists of the following:

Category	Description
Auditing and Logging	Who did what and when? Auditing and logging refer to how your application records security-related events
Authorization	What can you do? Authorization is how your application provides access controls for resources and operations
Authentication	Who are you? Authentication is the process where an entity proves the identity of another entity, typically through credentials, such as a user name and password
Communication Security	Whom are you talking to? Communication Security ensures all communication done is as secure as possible
Configuration Management	Whom does your application run as? Which databases does it connect to? How is your application administered? How are these settings secured? Configuration management refers to how your application handles these operational issues
Cryptography	How are you keeping secrets (confidentiality)? How are you tamper proofing your data or libraries (integrity)? How are you providing seeds for random values that must be cryptographically strong? Cryptography refers to how your application enforces confidentiality and integrity
Exception Management	When a method call in your application fails, what does your application do? How much do you reveal? Do you return friendly error information to end users? Do you pass valuable exception information back to the caller? Does your application fail gracefully?
Input Validation	How do you know that the input your application receives is valid and safe? Input validation refers to how your application filters, scrubs, or rejects input before additional processing. Consider constraining input through entry points and encoding output through exit points. Do you trust data from sources such as databases and file shares?
Sensitive Data	How does your application handle sensitive data? Sensitive data refers to how your application handles any data that must be protected either in memory, over the network, or in persistent stores
Session Management	How does your application handle and protect user sessions? A session refers to a series of related interactions between a user and your Web application

This helps you identify:

- Where are the most common mistakes made
- Where are the most actionable improvements

As a result, one may use these categories to focus and prioritize the security work, so that if the most prevalent security issues occur in the input validation, authentication and authorization categories, these areas can be focused first.

Procedure:

Download and install Microsoft threat modeling tool

- Creating a New Threat Model
- Modifying an Existing Threat Model
- Upgrading a Threat Model to use a new Template
- Analyzing the Threat Modeling Tool Output- various threat categories and mitigations - Reports

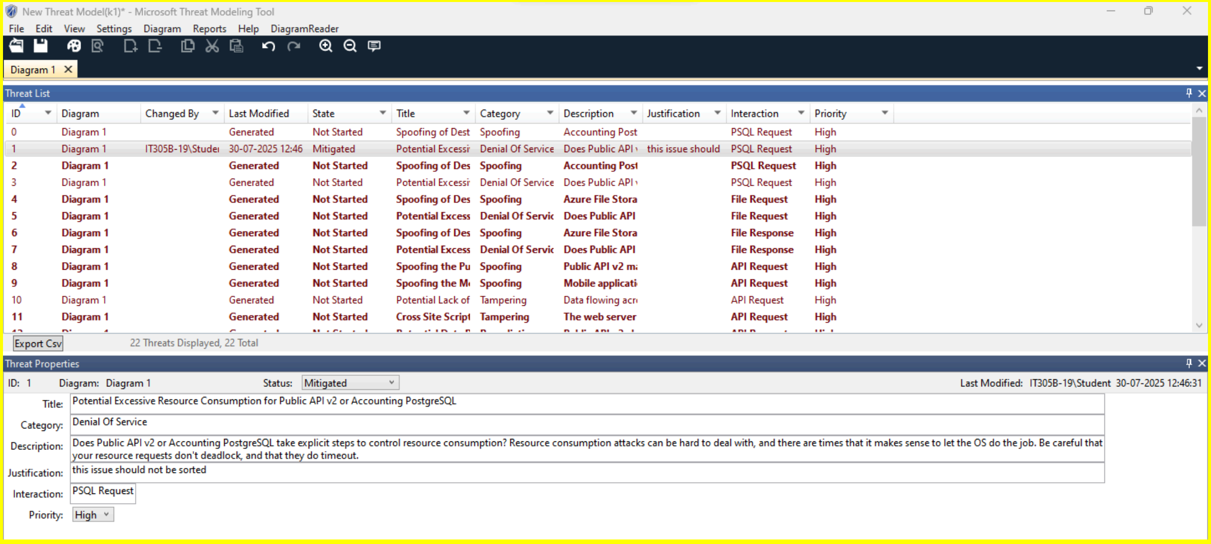
Post-Experimental Exercise
Questions:
List and discuss ten threat-modeling methodologies

Conclusion:

- Write what was performed in the experiment.
- Write the significance of the topic studied in the experiment.

References

- https://www.synopsys.com/glossary/what-is-threat-modeling.html
- https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool
- https://www.youtube.com/watch?v=uOGE0VIcnBo


New Threat Model(k1)* - Microsoft Threat Modeling Tool
File Edit View Settings Diagram Reports Help DiagramReader
Diagram 1
Threat List
Export Csv
Threat Properties

ID	Diagram	Changed By	Last Modified	State	Title	Category	Description	Justification	Interaction	Priority
0	Diagram 1		Generated	Not Started	Spoofing of Dest	Spoofing	Accounting Post		PSQL Request	High
1	Diagram 1	IT305B-19\Studer	30-07-2025 12:46	Mitigated	Potential Excessi	Denial Of Service	Does Public API	this issue should	PSQL Request	High
2	Diagram 1		Generated	Not Started	Spoofing of Des	Spoofing	Accounting Post		PSQL Request	High
3	Diagram 1		Generated	Not Started	Potential Excessi	Denial Of Service	Does Public API		PSQL Request	High
4	Diagram 1		Generated	Not Started	Spoofing of Des	Spoofing	Azure File Stora		File Request	High
5	Diagram 1		Generated	Not Started	Potential Excess	Denial Of Servic	Does Public API		File Request	High
6	Diagram 1		Generated	Not Started	Spoofing of Des	Spoofing	Azure File Stora		File Response	High
7	Diagram 1		Generated	Not Started	Potential Excess	Denial Of Servic	Does Public API		File Response	High
8	Diagram 1		Generated	Not Started	Spoofing the Pu	Spoofing	Public API v2 m		API Request	High
9	Diagram 1		Generated	Not Started	Spoofing the M	Spoofing	Mobile applicati		API Request	High
10	Diagram 1		Generated	Not Started	Potential Lack of	Tampering	Data flowing acr		API Request	High
11	Diagram 1		Generated	Not Started	Cross Site Script	Tampering	The web server		API Request	High

22 Threats Displayed, 22 Total

ID: 1 Diagram: Diagram 1 Status: Mitigated Last Modified: IT305B-19\Student 30-07-2025 12:46:31

Title: Potential Excessive Resource Consumption for Public API v2 or Accounting PostgreSQL

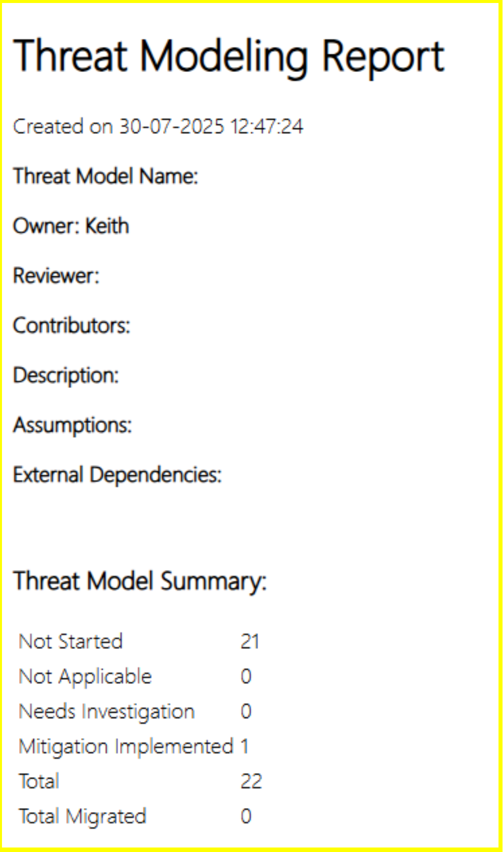
Category: Denial Of Service

Description: Does Public API v2 or Accounting PostgreSQL take explicit steps to control resource consumption? Resource consumption attacks can be hard to deal with, and there are times that it makes sense to let the OS do the job. Be careful that your resource requests don't deadlock, and that they do timeout.

Justification: this issue should not be sorted

Interaction: PSQL Request

Priority: High



Threat Modeling Report

Created on 30-07-2025 12:47:24

Threat Model Name:

Owner: Keith

Reviewer:

Contributors:

Description:

Assumptions:

External Dependencies:

Threat Model Summary:

Not Started	21
Not Applicable	0
Needs Investigation	0
Mitigation Implemented	1
Total	22
Total Migrated	0

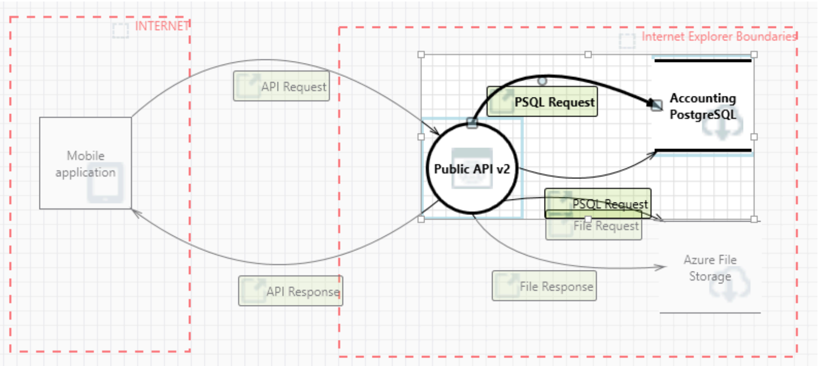
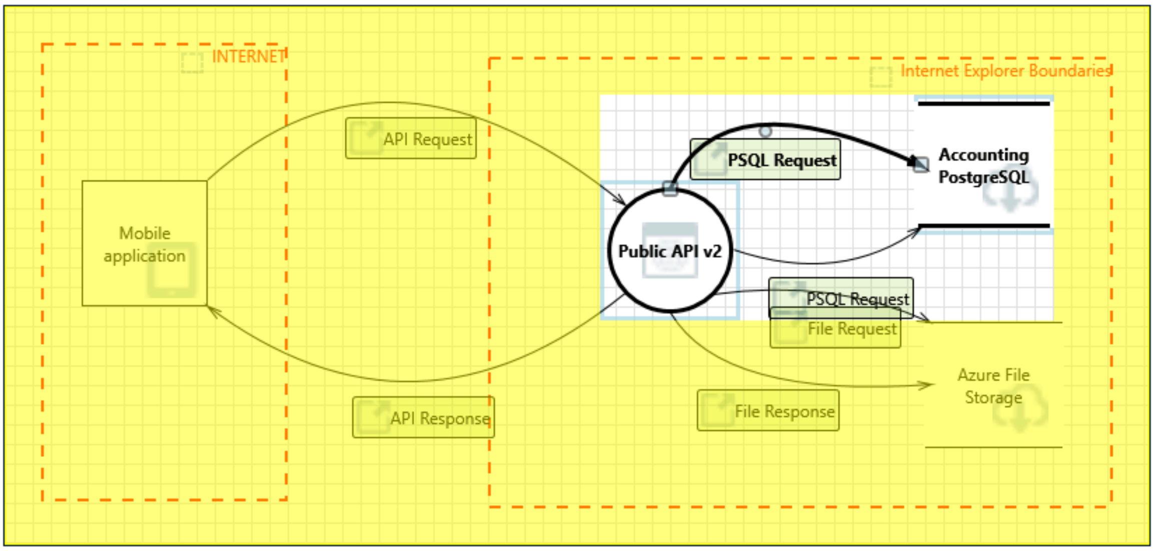


Diagram: Diagram 1



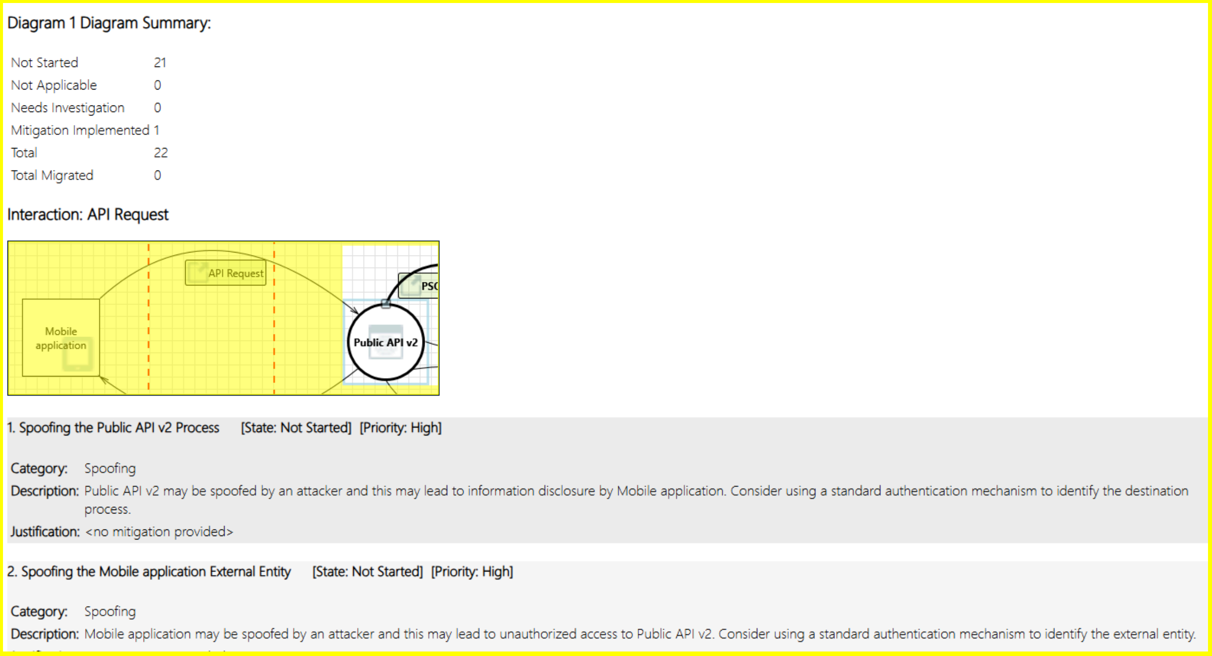


Diagram 1 Diagram Summary:

Not Started	21
Not Applicable	0
Needs Investigation	0
Mitigation Implemented	1
Total	22
Total Migrated	0

Interaction: API Request

1. Spoofing the Public API v2 Process [State: Not Started] [Priority: High]

Category: Spoofing

Description: Public API v2 may be spoofed by an attacker and this may lead to information disclosure by Mobile application. Consider using a standard authentication mechanism to identify the destination process.

Justification: <no mitigation provided>

2. Spoofing the Mobile application External Entity [State: Not Started] [Priority: High]

Category: Spoofing

Description: Mobile application may be spoofed by an attacker and this may lead to unauthorized access to Public API v2. Consider using a standard authentication mechanism to identify the external entity.