SEC-301: Security Challenges in Modern AI Systems

Lecture 2 – Security Threat Analysis Techniques

Instructed By:

Dr. Charnon Pattiyanon

Assistant Director of IT and Instructor
CMKL University

Artificial Intelligence and Computer
Engineering (AICE) Program

Class Outline

- Upon successful of this lecture, you will know about:
  - The conventional approaches and techniques to **model security threats** of an AI system and the benefits on how they help system analysts to identify hidden security threats.
  - **Basic security models** that could be helpful to protect AI systems against security threats.

Recap on Security Threats

Security Threats!

- **Security threat (n.)** in computer engineering and science field is a set of harmful activities that could potentially threaten computer users or systems.

The Internet Engineering Task Force (IETF) defines "Security Threat" in **RFC 4545** as "A potential for violation of security, which exists when there is an entity, circumstance, capability, action, or event that could cause harm".

The National Institute of Standardization and Technology (NIST) defines "Security Threat" in **NIST SP800-160, Vol. 2** as "An event or condition that has the potential for causing asset loss and the undesirable consequences or impact from such loss".

Basic Types of Security Threat – STRIDE Model

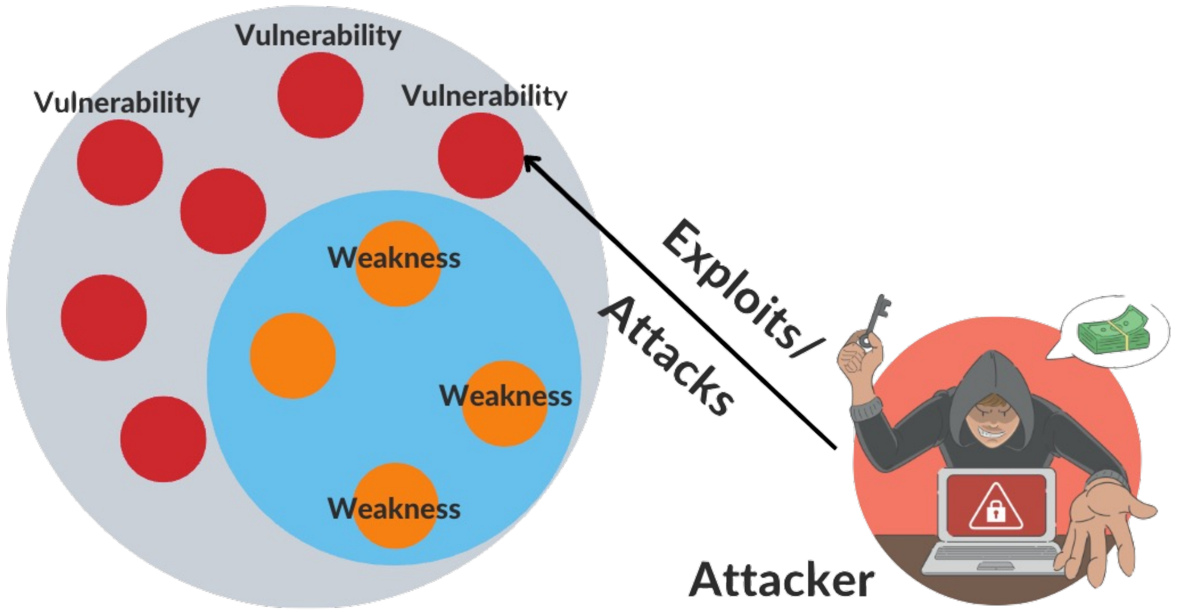
Threat	Meaning/Example	Related Security Property
Identity Spoofing	An attempt to use someone else's password to authenticate to services as that person.	Authentication
Data Tampering	An attempt to modify data either the data at rest, or the data during the trasmission.	Integrity
Repudiation	An attempt to deny having performed data manipulation.	Non-repudiation
Information Disclosure	An attempt to read user data without permission, or eavedrop a data communication channel.	Confidentiality
Denial of Service	An attempt to provide flood of traffic to a system with the intention to make it unavailable.	Availability
Elevation of Privilege	An attempt to gain higher privilege to a less privilege account.	Authorization

Weaknesses and Vulnerabilities Lead to Threats

- Security threats occurred when attackers exploit on system's weaknesses, or vulnerabilities to gain some benefits.

"A ***weakness*** is a condition in a software, firmware, hardware, or service component that, under certain circumstances, could contribute to the introduction of vulnerabilities." – ***MITRE foundation***.

"A ***vulnerability*** is a flaw in a software, firmware, hardware, or service component resulting from a weakness that can be exploited, causing a negative impact to the confidentiality, integrity, or availability of an impacted component or components." – ***MITRE foundation***.

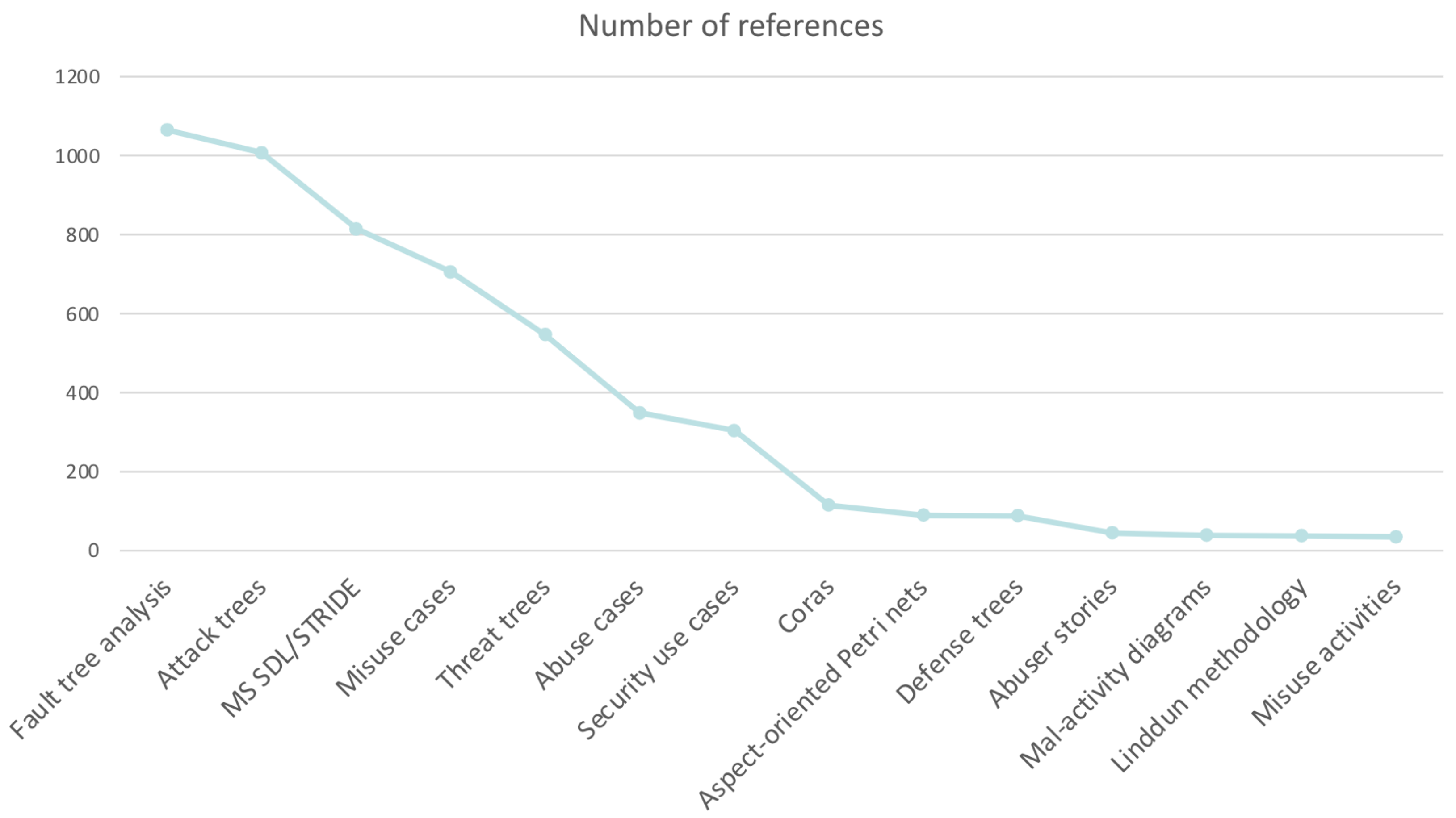


Security Threat Modeling Techniques

- **Security Threat Modeling** is a group of analysis techniques or methods to **identify** potential threats to a target system.

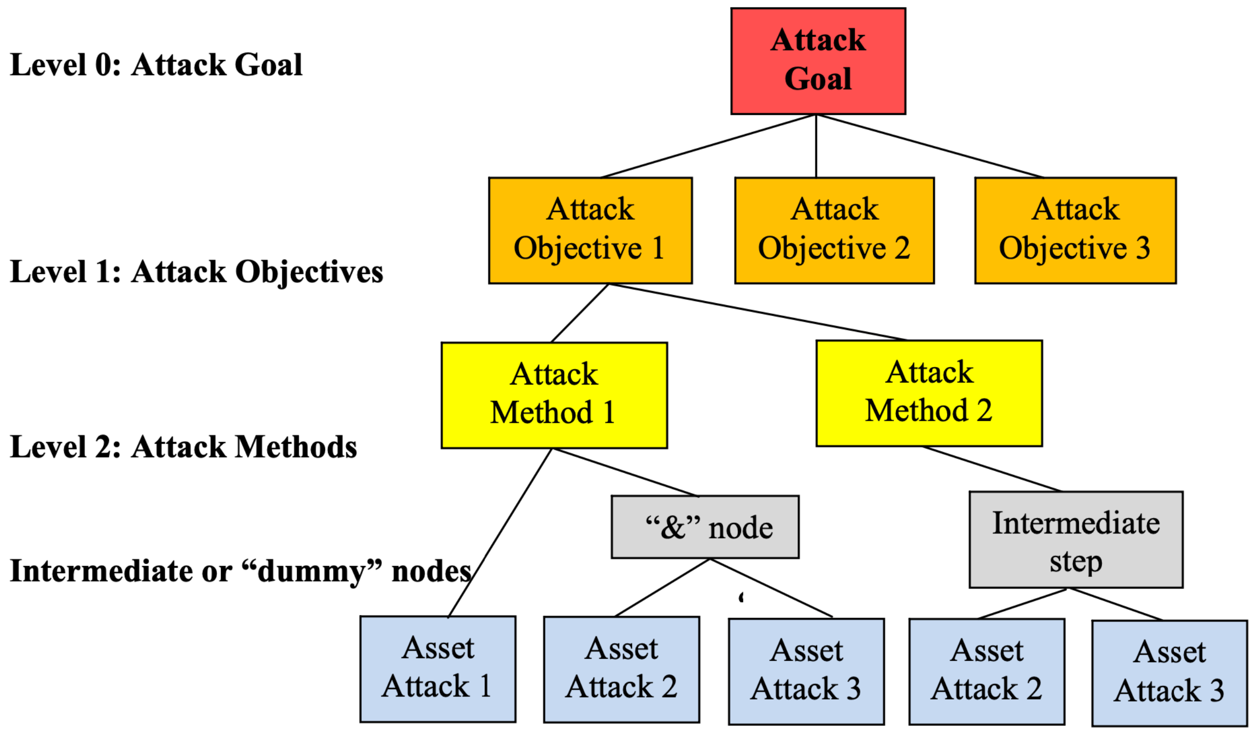
- There are **three main approaches**, which are:
  - Attacker Approaches: This group of approaches aims to identify **who are the attacker**, what are their intentions to attack, and how they can exploit the system.
  - Asset Approaches: This group of approaches aims to **identify values of asset** in the target system, and how attackers can reach the asset.
  - Software Approaches: This group of approaches aims to **identify potential vulnerabilities** that attackers are likely to exploit in a software system, and how attackers can access to those vulnerabilities.

Security Threat Modeling Techniques



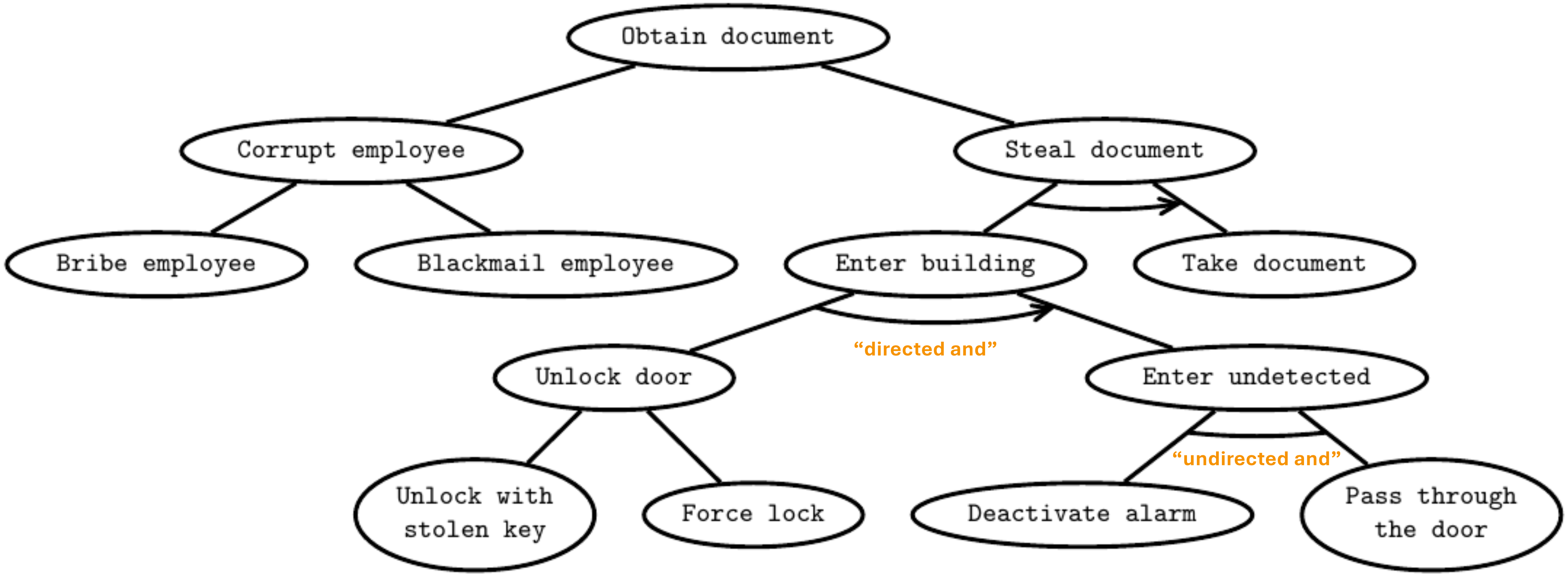
1. Attack Tree Analysis

- **Attack Tree Analysis** formulates potential attacks on the target system as a tree structure, where the root nodes represent attack goals and objectives, and the leaf nodes represent the various methods for achieving those goals.



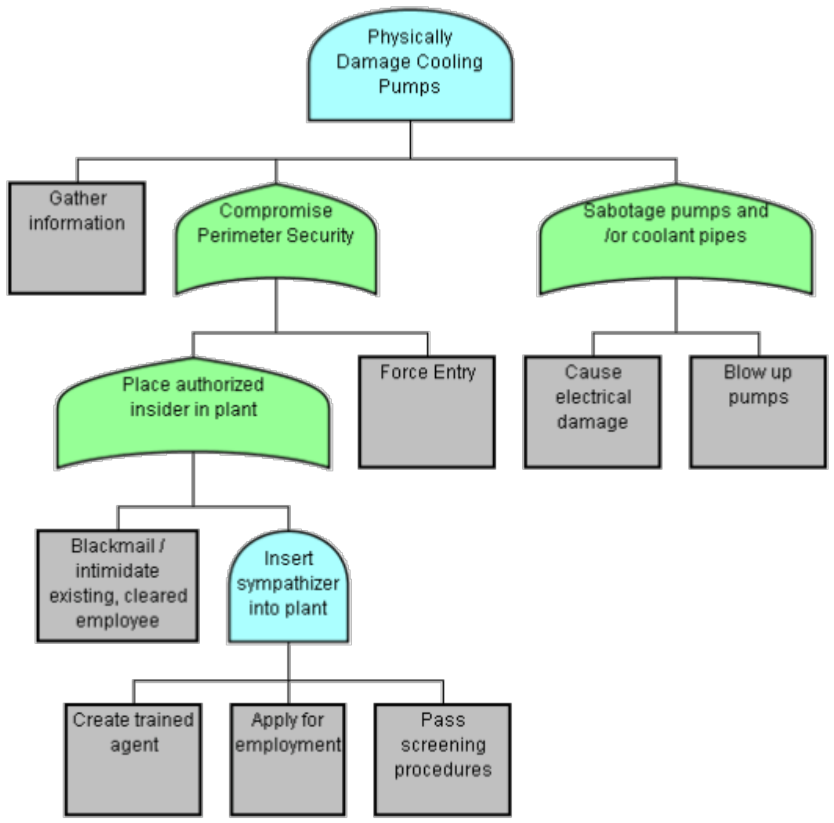
1. Attack Tree Analysis

- **Examples of An Attack Tree Analysis Result:**

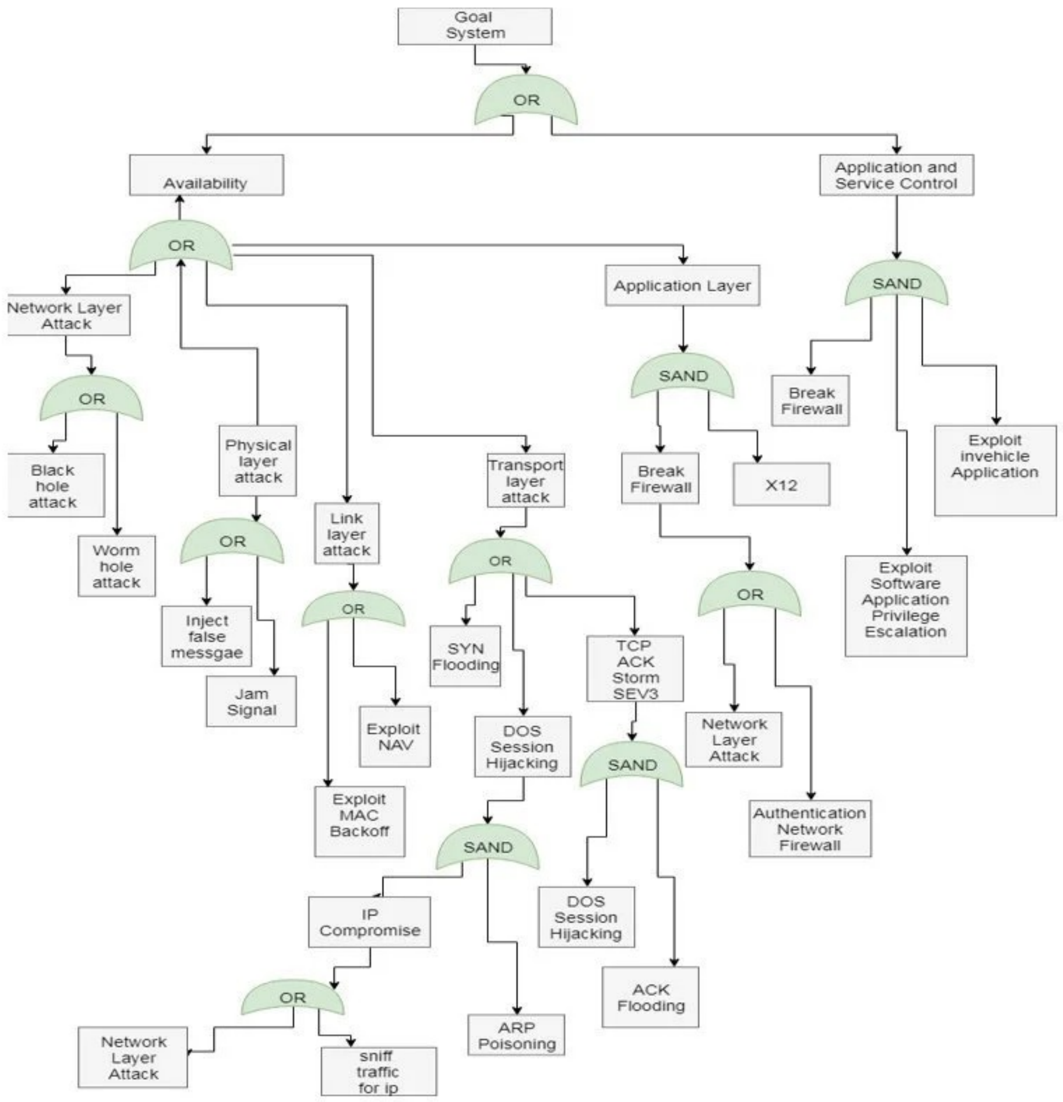


1. Attack Tree Analysis

- **Examples of An Attack Tree Analysis Result:**

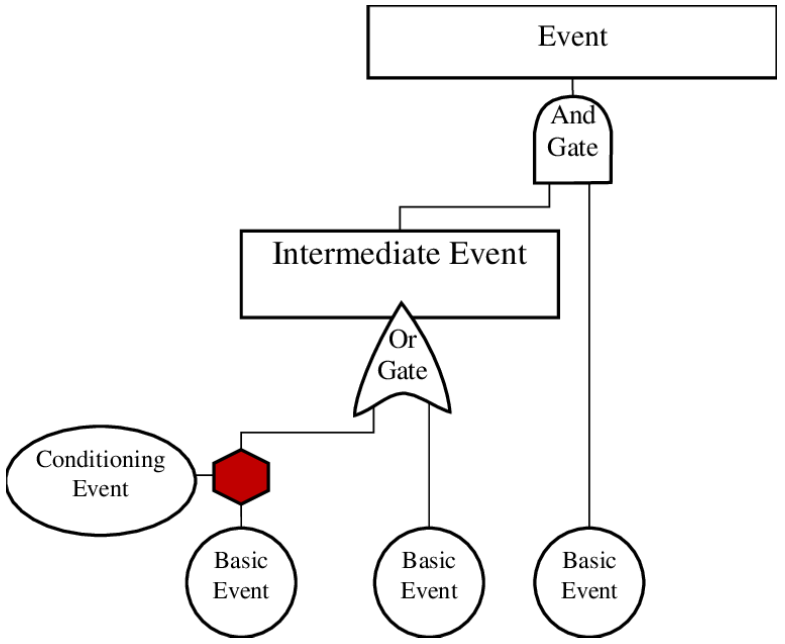


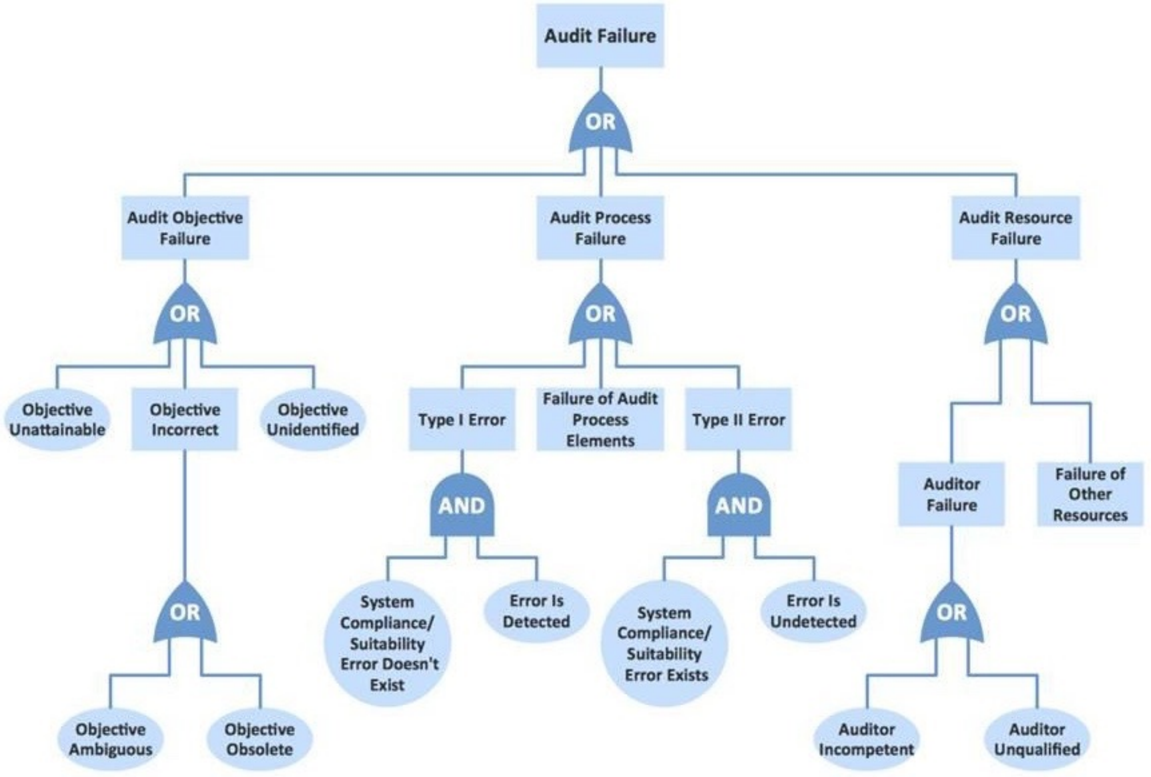
Physically Damage Cooling Pumps
Gather information
Compromise Perimeter Security
Sabotage pumps and /or coolant pipes
Place authorized insider in plant
Force Entry
Cause electrical damage
Blow up pumps
Blackmail / intimidate existing, cleared employee
Insert sympathizer into plant
Create trained agent
Apply for employment
Pass screening procedures



2. Fault Tree Analysis

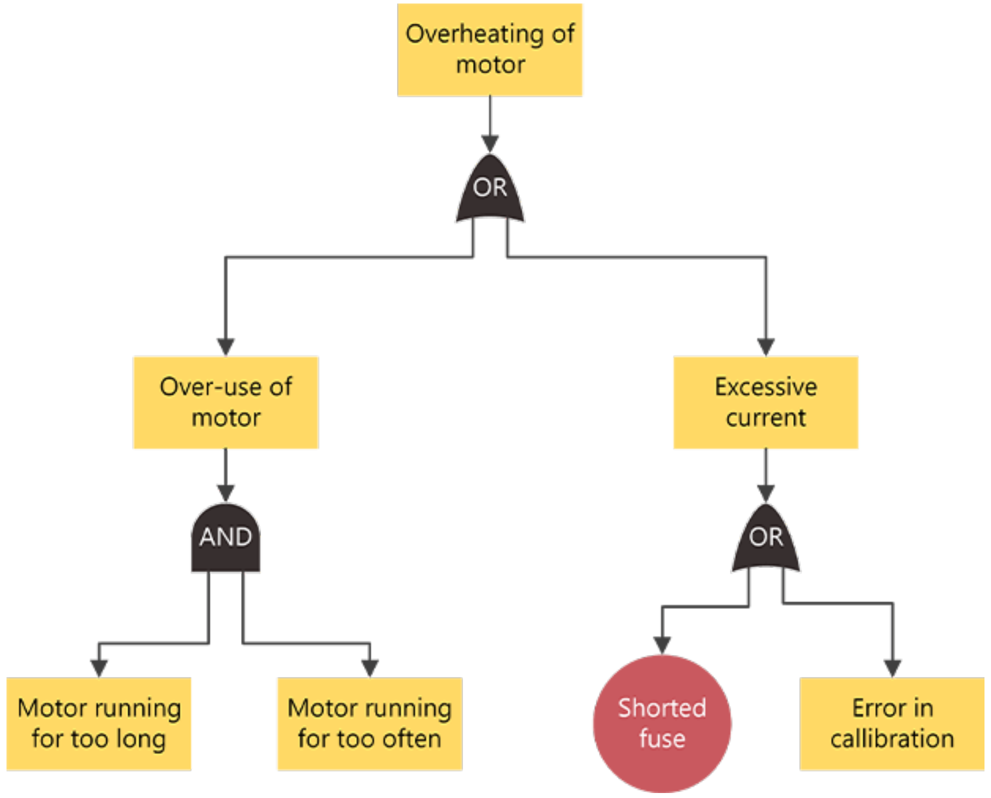
- **Fault Tree Analysis** is similar to the attack tree analysis, but it is a top-down approach to identify the potential component-level failures (basic events) that cause the system-level failures (top events) to occur.

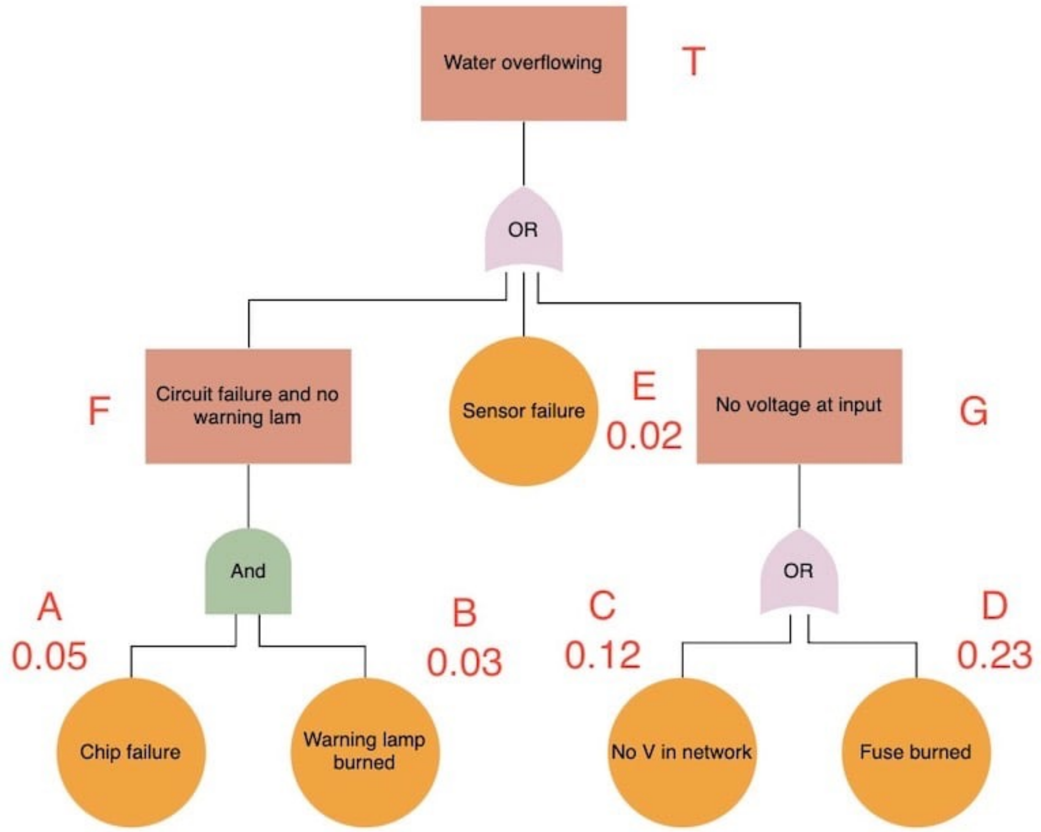




2. Fault Tree Analysis

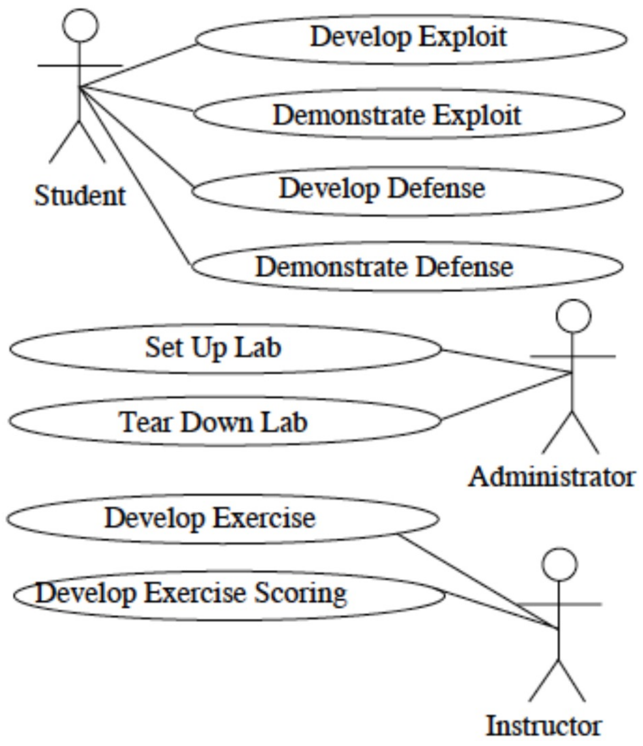
- **Examples of Fault Tree Analysis Results:**

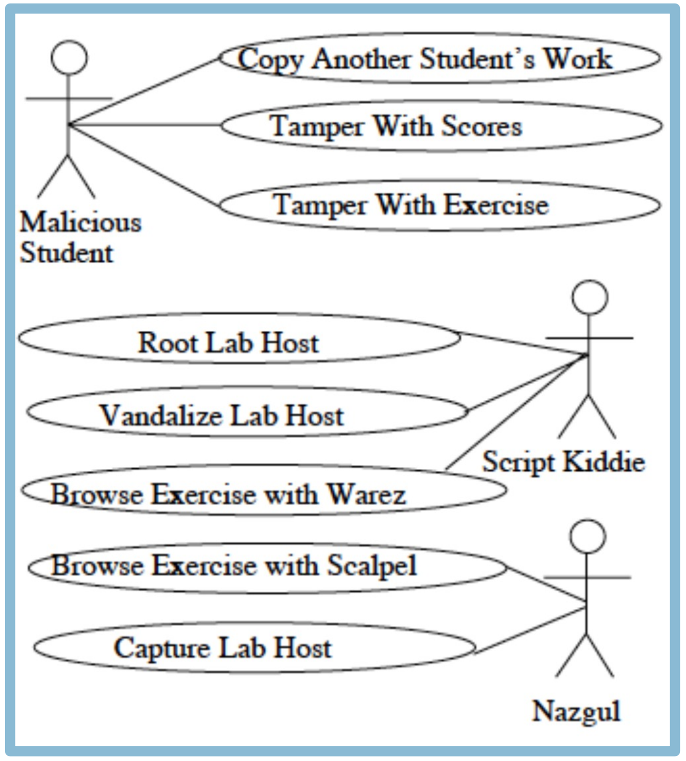




3. Abuse Case Analysis

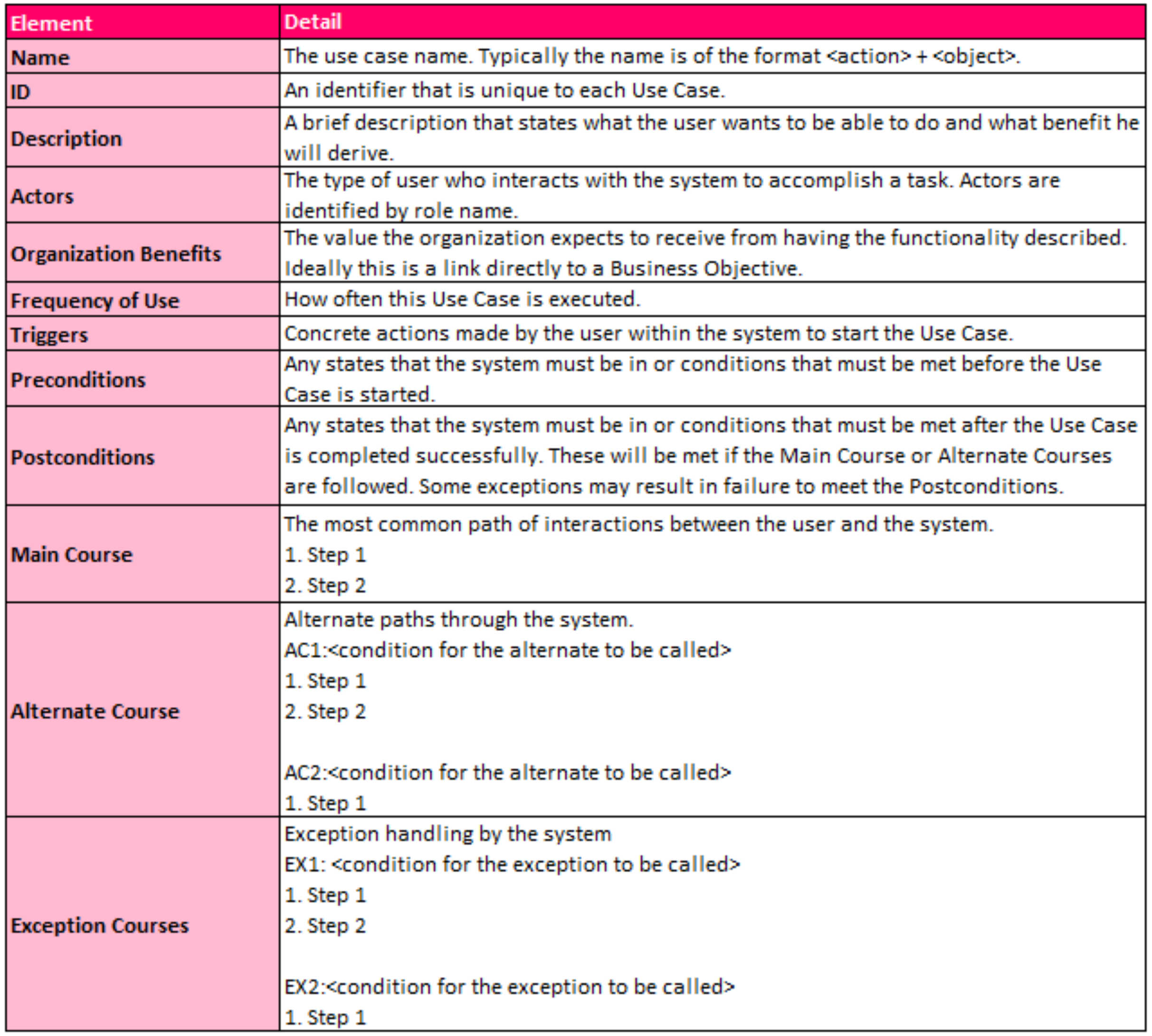
- **Abuse Case Analysis** is the opposite technique of use case analysis (i.e., modeling use cases that users will interact with the target system), which aims to identify cases that are considered abuse to the system.





3. Abuse Case Analysis

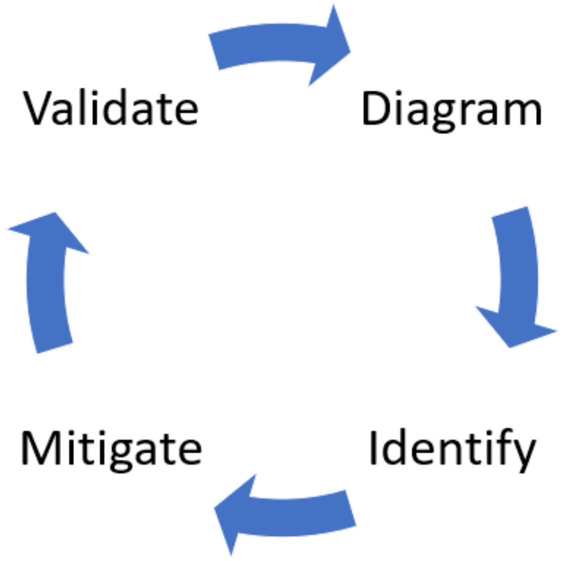
- **Abuse Case Description:** Similar to use case description, it is required to clearly describe abuse cases with steps.

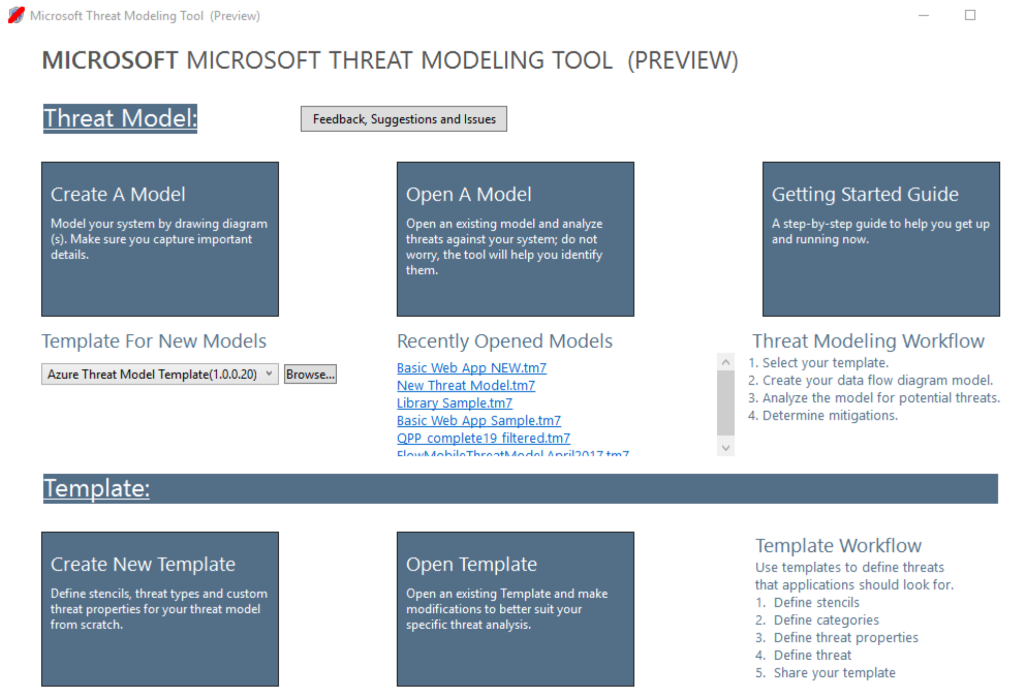


Element	Detail
Name	The use case name. Typically the name is of the format <action> + <object>.
ID	An identifier that is unique to each Use Case.
Description	A brief description that states what the user wants to be able to do and what benefit he will derive.
Actors	The type of user who interacts with the system to accomplish a task. Actors are identified by role name.
Organization Benefits	The value the organization expects to receive from having the functionality described. Ideally this is a link directly to a Business Objective.
Frequency of Use	How often this Use Case is executed.
Triggers	Concrete actions made by the user within the system to start the Use Case.
Preconditions	Any states that the system must be in or conditions that must be met before the Use Case is started.
Postconditions	Any states that the system must be in or conditions that must be met after the Use Case is completed successfully. These will be met if the Main Course or Alternate Courses are followed. Some exceptions may result in failure to meet the Postconditions.
Main Course	The most common path of interactions between the user and the system. 1. Step 1 2. Step 2
Alternate Course	Alternate paths through the system. AC1:<condition for the alternate to be called> 1. Step 1 2. Step 2 AC2:<condition for the alternate to be called> 1. Step 1
Exception Courses	Exception handling by the system EX1: <condition for the exception to be called> 1. Step 1 2. Step 2 EX2:<condition for the exception to be called> 1. Step 1

4. Microsoft Secure Development Lifecycle

- **Microsoft's Security Development Lifecycle (MS SDL) Threat Modeling Tool** is a well-known technique and set of tools introduced by Microsoft to help model target systems, analyze threats from predictable categories, and document them systematically.





Microsoft Threat Modeling Tool (Preview)
Feedback, Suggestions and Issues
Browse...

MICROSOFT MICROSOFT THREAT MODELING TOOL (PREVIEW)

Threat Model:

Create A Model
Model your system by drawing diagram (s). Make sure you capture important details.

Open A Model
Open an existing model and analyze threats against your system; do not worry, the tool will help you identify them.

Getting Started Guide
A step-by-step guide to help you get up and running now.

Template For New Models
Azure Threat Model Template(1.0.0.20)

Recently Opened Models
Basic Web App NEW.tm7
New Threat Model.tm7
Library Sample.tm7
Basic Web App Sample.tm7
QPP complete19 filtered.tm7
FlowMobileThreatModel April2017.tm7

Threat Modeling Workflow
1. Select your template.
2. Create your data flow diagram model.
3. Analyze the model for potential threats.
4. Determine mitigations.

Template:

Create New Template
Define stencils, threat types and custom threat properties for your threat model from scratch.

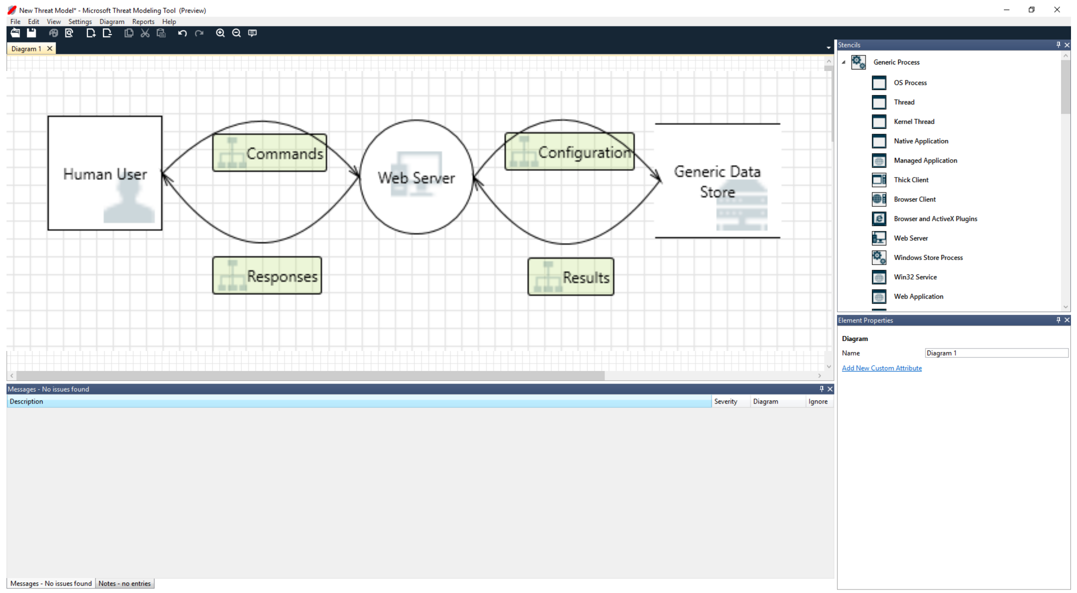
Open Template
Open an existing Template and make modifications to better suit your specific threat analysis.

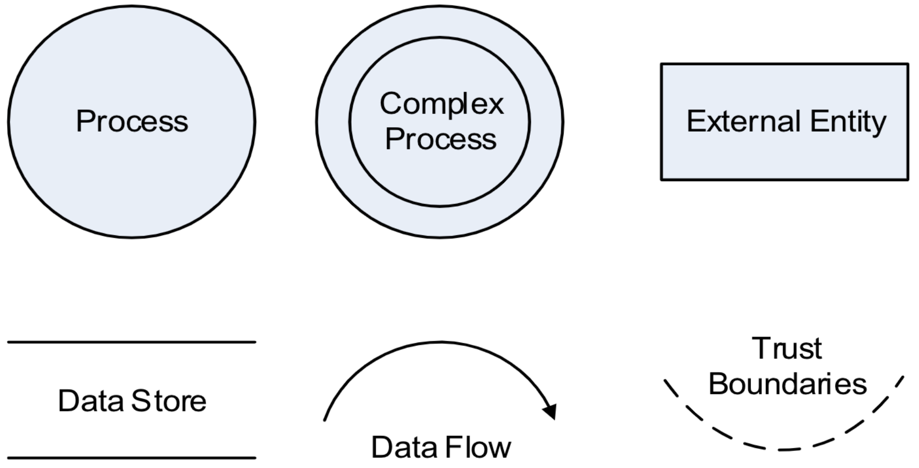
Template Workflow
Use templates to define threats that applications should look for.
1. Define stencils
2. Define categories
3. Define threat properties
4. Define threat
5. Share your template

4. Microsoft Secure Development Lifecycle

- **Microsoft's Security Development Lifecycle (MS SDL) Threat Modeling Tool** is a well-known technique and set of tools introduced by Microsoft to help model target systems, analyze threats from predictable categories, and document them systematically.

Step 1: Modeling the target system with a Data Flow Diagram (DFD)

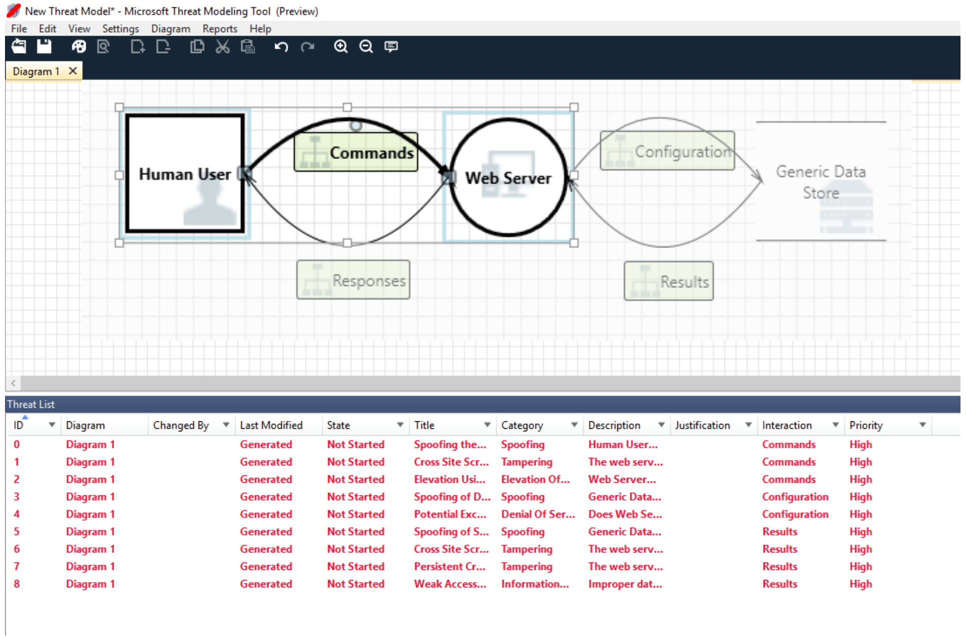




4. Microsoft Secure Development Lifecycle

- **Microsoft's Security Development Lifecycle (MS SDL) Threat Modeling Tool** is a well-known technique and set of tools introduced by Microsoft to help model target systems, analyze threats from predictable categories, and document them systematically.

Step 2: The tool automatically analyzes the model against predefined sets of threats.


New Threat Model* - Microsoft Threat Modeling Tool (Preview)
File Edit View Settings Diagram Reports Help
Diagram 1
Human User
Commands
Web Server
Configuration
Generic Data Store
Responses
Results
Threat List

ID	Diagram	Changed By	Last Modified	State	Title	Category	Description	Justification	Interaction	Priority
0	Diagram 1		Generated	Not Started	Spoofing the...	Spoofing	Human User...		Commands	High
1	Diagram 1		Generated	Not Started	Cross Site Scr...	Tampering	The web serv...		Commands	High
2	Diagram 1		Generated	Not Started	Elevation Usi...	Elevation Of...	Web Server...		Commands	High
3	Diagram 1		Generated	Not Started	Spoofing of D...	Spoofing	Generic Data...		Configuration	High
4	Diagram 1		Generated	Not Started	Potential Exc...	Denial Of Ser...	Does Web Se...		Configuration	High
5	Diagram 1		Generated	Not Started	Spoofing of S...	Spoofing	Generic Data...		Results	High
6	Diagram 1		Generated	Not Started	Cross Site Scr...	Tampering	The web serv...		Results	High
7	Diagram 1		Generated	Not Started	Persistent Cr...	Tampering	The web serv...		Results	High
8	Diagram 1		Generated	Not Started	Weak Access...	Information...	Improper dat...		Results	High

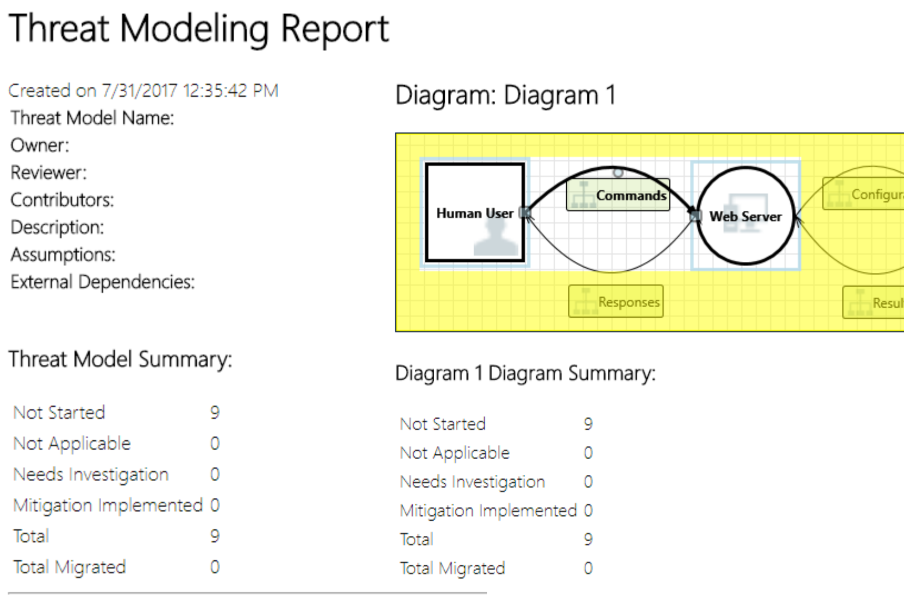
The tool analyzes threats using the **STRIDE model**:
Spoofing, **T**ampering, **R**epudiation,
Information Disclosure, **D**enial of Service, and
Elevation of Privilege.

DFD entity	S	T	R	I	D	E
External Entity	X		X			
Data Flow		X		X	X	
Data Store		X	(X)	X	X	
Process	X	X	X	X	X	X

4. Microsoft Secure Development Lifecycle

- **Microsoft's Security Development Lifecycle (MS SDL) Threat Modeling Tool** is a well-known technique and set of tools introduced by Microsoft to help model target systems, analyze threats from predictable categories, and document them systematically.

Step 3: The tool automatically generates a threat modeling report.



Threat Modeling Report

Created on 7/31/2017 12:35:42 PM
Threat Model Name:
Owner:
Reviewer:
Contributors:
Description:
Assumptions:
External Dependencies:

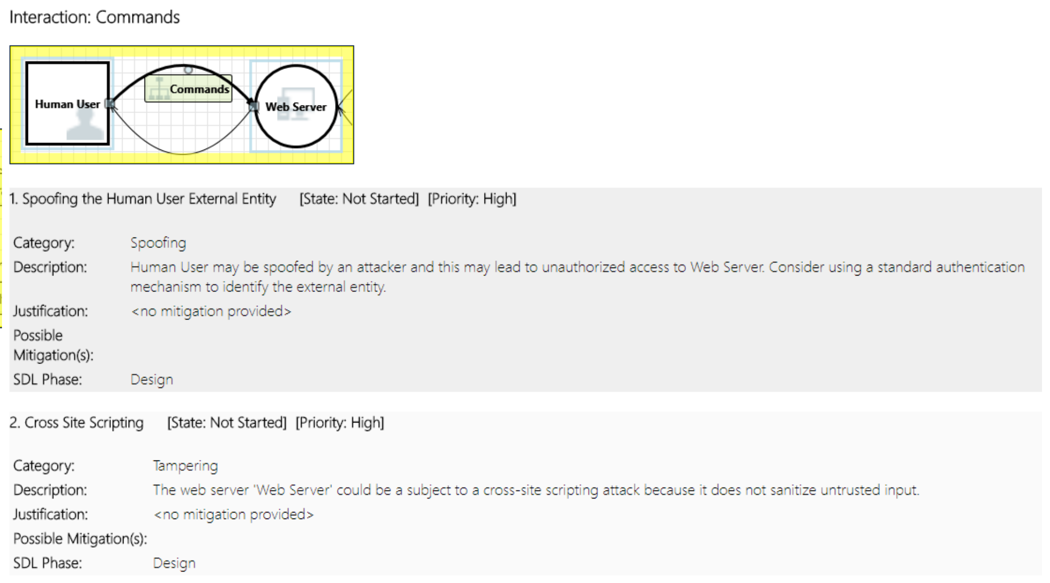
Threat Model Summary:

Not Started	9
Not Applicable	0
Needs Investigation	0
Mitigation Implemented	0
Total	9
Total Migrated	0

Diagram: Diagram 1

Diagram 1 Diagram Summary:

Not Started	9
Not Applicable	0
Needs Investigation	0
Mitigation Implemented	0
Total	9
Total Migrated	0



Interaction: Commands

1. Spoofing the Human User External Entity [State: Not Started] [Priority: High]

Category:	Spoofing
Description:	Human User 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.
Justification:	<no mitigation provided>
Possible Mitigation(s):	
SDL Phase:	Design

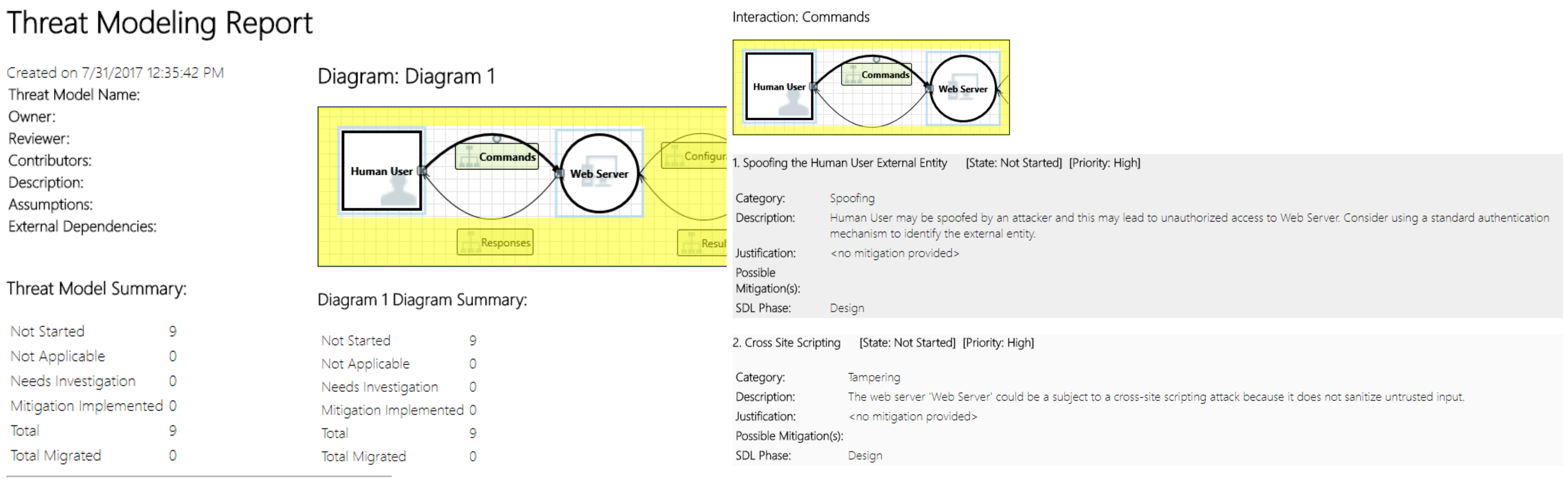
2. Cross Site Scripting [State: Not Started] [Priority: High]

Category:	Tampering
Description:	The web server 'Web Server' could be a subject to a cross-site scripting attack because it does not sanitize untrusted input.
Justification:	<no mitigation provided>
Possible Mitigation(s):	
SDL Phase:	Design

4. Microsoft Secure Development Lifecycle

- **Microsoft's Security Development Lifecycle (MS SDL) Threat Modeling Tool** is a well-known technique and set of tools introduced by Microsoft to help model target systems, analyze threats from predictable categories, and document them systematically.

  Step 3: The tool automatically generates a threat modeling report.



Threat Modeling Report

Created on 7/31/2017 12:35:42 PM
Threat Model Name:
Owner:
Reviewer:
Contributors:
Description:
Assumptions:
External Dependencies:

Threat Model Summary:

Not Started	9
Not Applicable	0
Needs Investigation	0
Mitigation Implemented	0
Total	9
Total Migrated	0

Diagram: Diagram 1

Diagram 1 Diagram Summary:

Not Started	9
Not Applicable	0
Needs Investigation	0
Mitigation Implemented	0
Total	9
Total Migrated	0

Interaction: Commands

1. Spoofing the Human User External Entity [State: Not Started] [Priority: High]

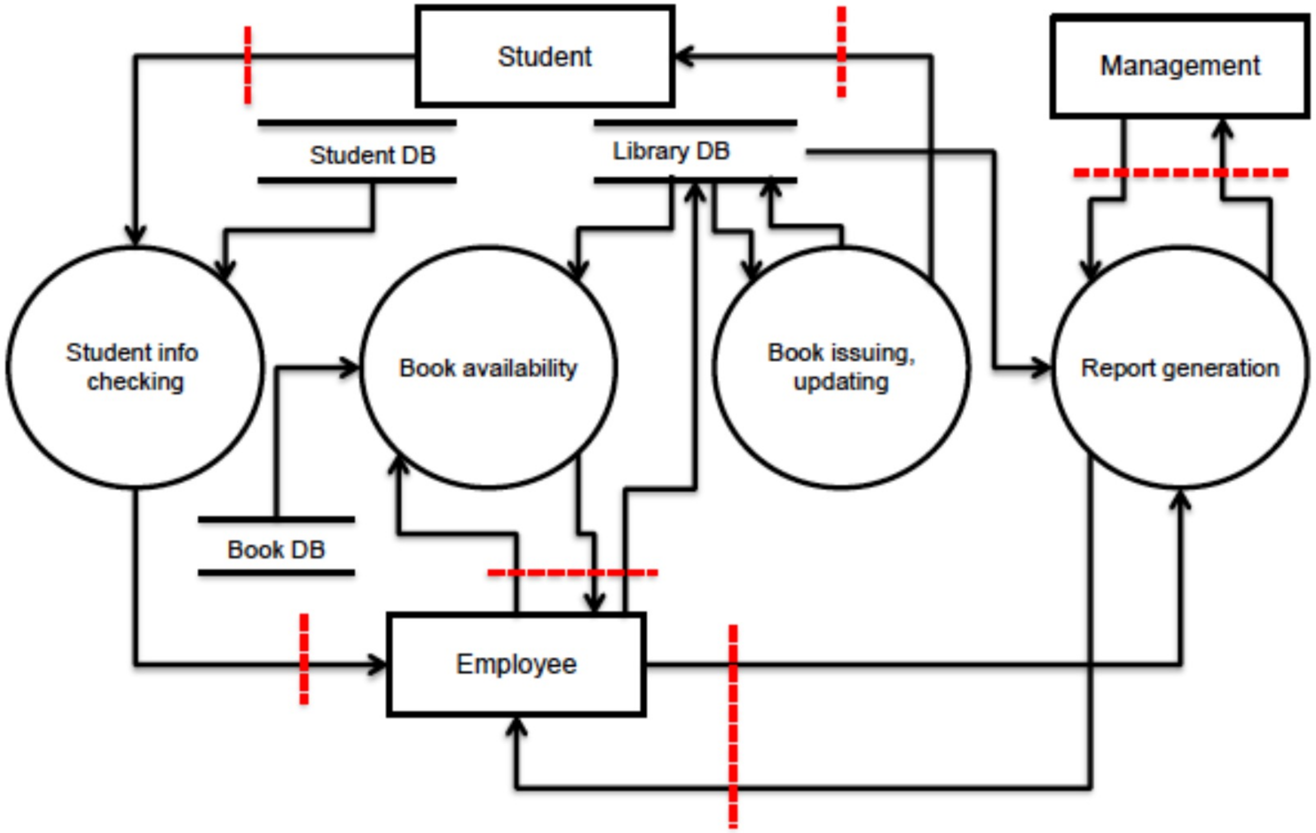
Category:	Spoofing
Description:	Human User may be spoofed by an attacker and this may lead to unauthorized access to Web Server. Consider using a standard authentication mechanism to identify the external entity.
Justification:	<no mitigation provided>
Possible Mitigation(s):	
SDL Phase:	Design

2. Cross Site Scripting [State: Not Started] [Priority: High]

Category:	Tampering
Description:	The web server 'Web Server' could be a subject to a cross-site scripting attack because it does not sanitize untrusted input.
Justification:	<no mitigation provided>
Possible Mitigation(s):	
SDL Phase:	Design

4. Microsoft Secure Development Lifecycle

- **An example of DFD**, following the MS SDL's threat modeling tool.



A Summary of Security Threat Modeling Techniques

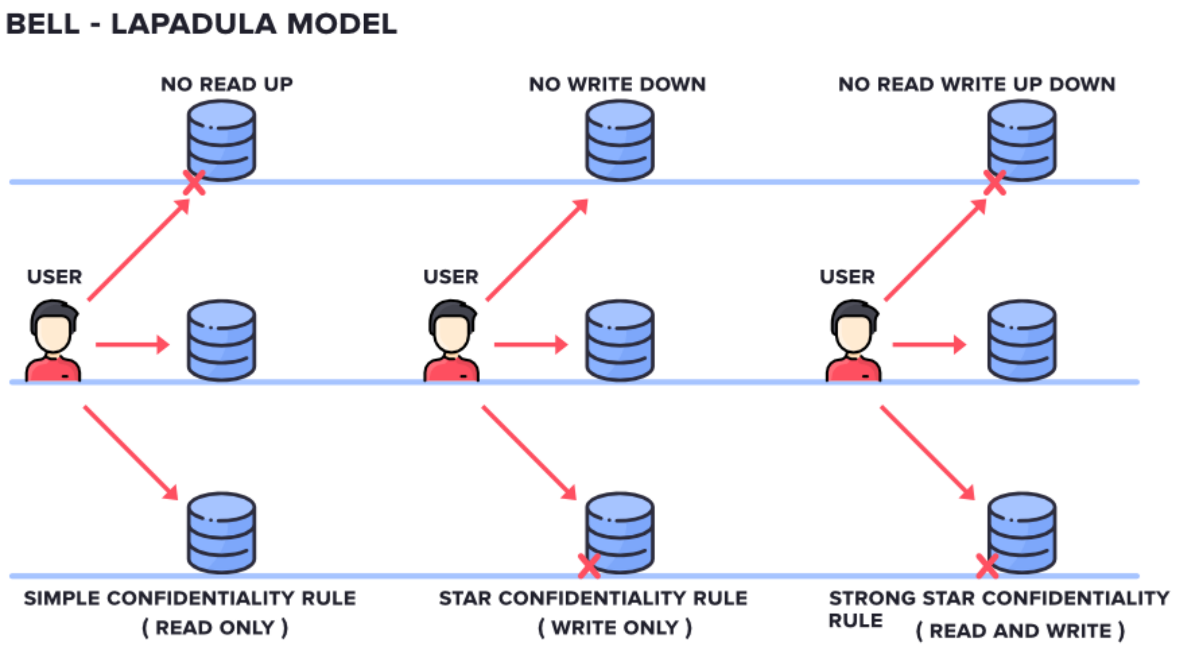
- Threat modeling techniques are aimed to **identify hidden security threats** that potentially occur in a target system.
- The main target of security threats are the ***system*** itself, ***assets***, or ***software*** application.
- The key expected outcomes of security threats are
    1. **Denial of Services** – Make the target system unavailable or failure
    2. **Access to unauthorized parts** – Make benefits from assets or information that should not be accessible.
    3. **Takeover the system or cause harmful events** – Make chaotic activities that cause harmful to the system's users or information.
- Threat modeling techniques did not protect security threats directly. So, how to protect?

Basic Security Model

- Security models are functional, protocol, or process models that are defined to ensure the alignment with security goals.
- There are four classic security models that are typically discussed:
    1. **Bell-LaPadula Security Model**
    2. **Biba Security Model**
    3. **Clarke Wilson Security Model**
    4. **Brewer-Nash Security Model**

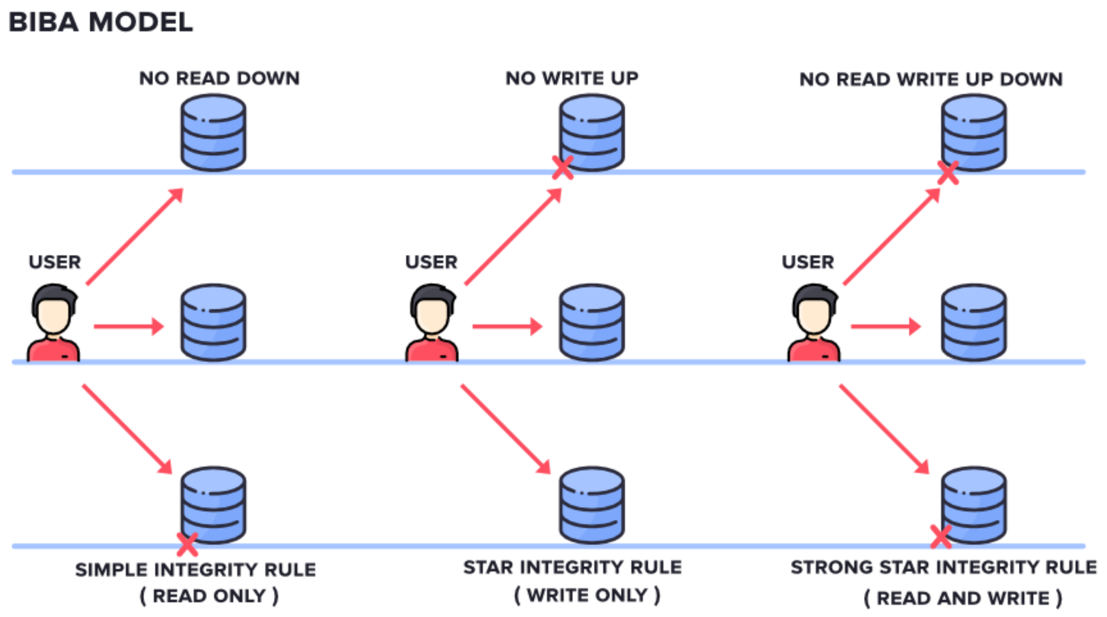
1. Bell-Lapadula Model

- **Bell-Lapadula Security Model** was developed by two scientists, David Elliot Bell, and Leonard J. LaPadula, to help maintaining the **confidentiality** of information.
- There are **three important rules**:
  - Simple Confidentiality Rule (***No Read Up***),
  - Star Confidentiality Rule (***No Write Down***),
  - Strong Star Confidentiality Rule (***No Read Write Up Down***)

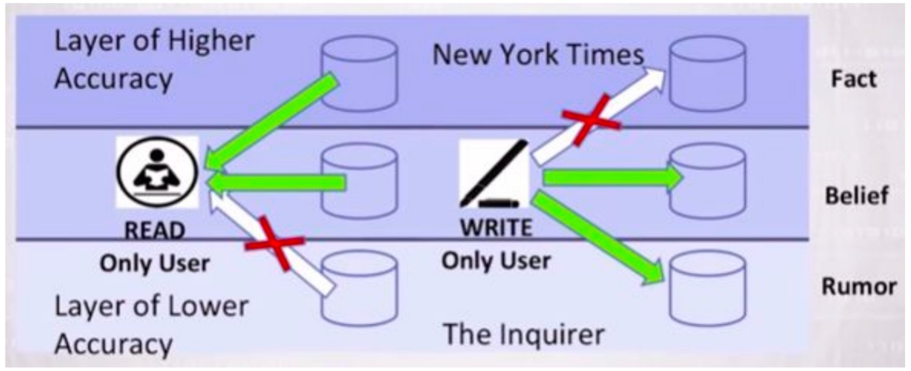


2. Biba Model

- **Biba Security Model** was developed by a scientist, Kenneth J. Biba, to help maintaining the **integrity** of information. It works the exact reverse of Bell-LaPadula security model.
- There are **three important rules**:
  - Simple Integrity Rule (***No Read Down***),
  - Star Integrity Rule (***No Write Up***),
  - Strong Star Integrity Rule (***No Read Write Up Down***)

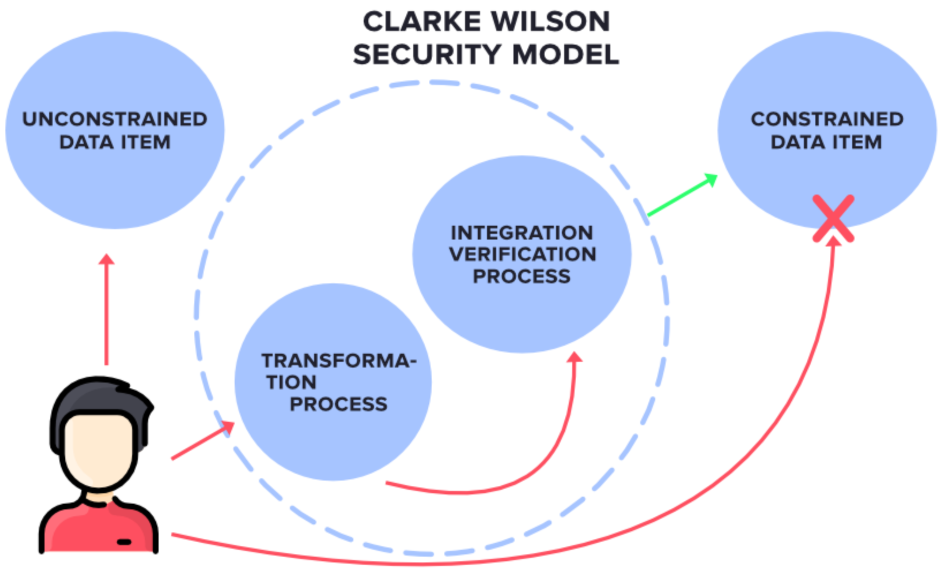


BIBA MODEL
NO READ DOWN
NO WRITE UP
NO READ WRITE UP DOWN
USER
USER
USER
SIMPLE INTEGRITY RULE (READ ONLY)
STAR INTEGRITY RULE (WRITE ONLY)
STRONG STAR INTEGRITY RULE (READ AND WRITE)



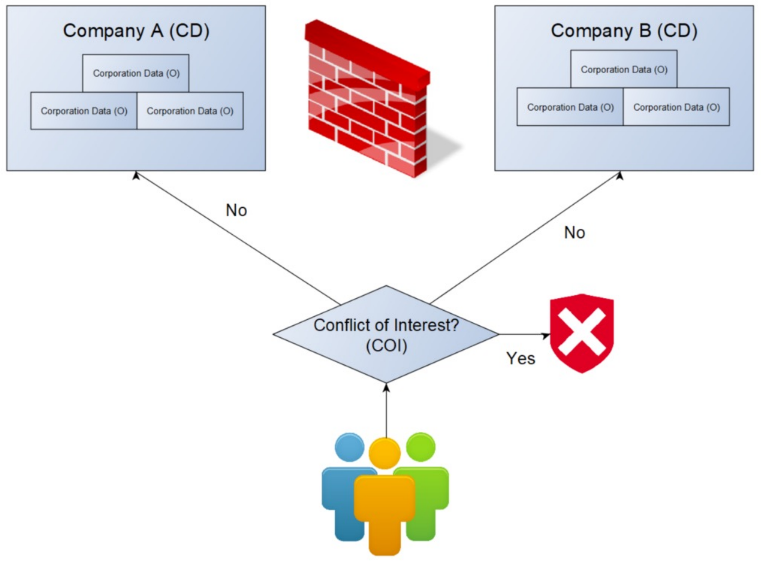
3. Clarke Wilson Model

- Clarke Wilson Security Model is a highly secured model where the accessibility scope of each item within a target system is carefully evaluated.
- There are **three entities** in this model: Subjects, Constrained Data Items, and Unconstrained Data Items.
- There are **two components**: ***Transformation Processes***, and ***Integration Verification Processes***.



4. Brewer-Nash Model

- **Brewer-Nash Security Model**, or Chinese Wall Model defines how to control read and write access based on the "***conflict of interest***" rules.
- Once a subject accesses a data object, he/she are automatically assigned into a group.
- From then on, the subject can only access objects of the same group.



End of the Lecture

Please do not hesitate to ask any questions to free your curiosity,

If you have any further questions after the class, please contact me via email (charnon@cmkl.ac.th).