**SAD Lab**

**Exp-3**

**Name: Shashwat Tripathi**

**Division: D20A**

**Roll no: 64**

**Aim:** To understand the concept of threat modeling using microsoft threat modeling tool.

**Theory:**

Threat modeling is a structured process used to identify, assess, and prioritize potential security threats to a system, application, or network. It helps in understanding what could go wrong, determining the possible impact, and planning mitigation strategies to protect against vulnerabilities. The goal is to improve security by anticipating and addressing threats before they can be exploited.

Importance of it threat modeling:

1. **Proactive Risk Management**: Identifies potential threats early, allowing for preemptive mitigation strategies to reduce risk.
2. **Cost-Effective Security**: Prevents costly security breaches by addressing vulnerabilities during the design phase.
3. **Improved Decision-Making**: Helps prioritize security efforts based on the potential impact and likelihood of threats.
4. **Enhanced System Security**: Strengthens the overall security posture by ensuring that all possible attack vectors are considered.
5. **Compliance and Standards Alignment**: Supports compliance with industry standards and regulations by systematically addressing security risks.

Process of threat modeling:

1. **Define the Scope**: Identify the system, application, or network to be assessed, including its components, boundaries, and data flow.
2. **Identify Assets**: Determine what needs protection, such as data, resources, and services within the system.
3. **Identify Threats**: Analyze potential threats, including attack vectors, vulnerabilities, and potential attackers.
4. **Assess Risks**: Evaluate the likelihood and impact of each identified threat to prioritize them.
5. **Develop Mitigations**: Plan and implement security controls to reduce or eliminate the risks associated with each threat.
6. **Validate and Review**: Continuously review and update the threat model to adapt to new threats, changes in the system, or evolving security requirements.

Methodologies of threat modeling

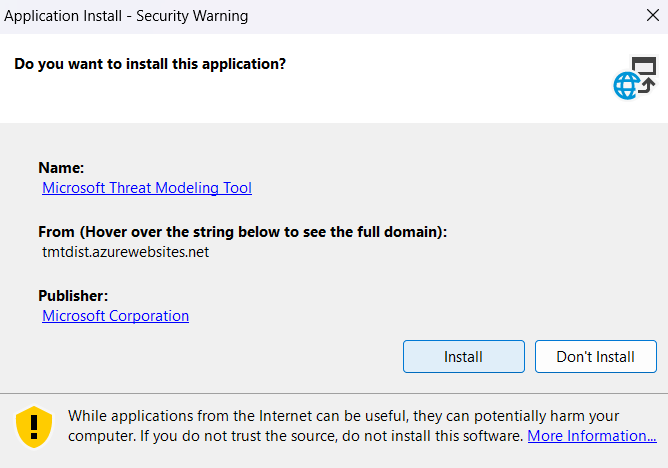
1. **STRIDE**: A framework that categorizes threats into six types: Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege.
2. **DREAD**: A risk assessment model that evaluates threats based on Damage potential, Reproducibility, Exploitability, Affected users, and Discoverability.
3. **PASTA (Process for Attack Simulation and Threat Analysis)**: A risk-centric methodology that aligns business objectives with security requirements through a structured seven-step process.
4. **Trike**: A security auditing framework that combines risk management with a focus on stakeholders' security expectations and risk tolerance.
5. **CVSS (Common Vulnerability Scoring System)**: A standardized method for assessing the severity of security vulnerabilities, used to prioritize threat mitigation.
6. **OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation)**: A methodology focusing on organizational risk and its impact on business objectives.
7. **Attack Trees**: A method that visualizes potential attack paths by breaking down threats into smaller components within a tree structure.
8. **LINDDUN**: A privacy-focused threat modeling methodology that addresses threats related to Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of information, Unawareness, and Non-compliance.
9. **VAST (Visual, Agile, and Simple Threat)**: A model that integrates threat modeling into Agile and DevOps processes, emphasizing scalability and simplicity.
10. **CBR (Cyber Kill Chain)**: A model that outlines the stages of a cyberattack, from initial reconnaissance to data exfiltration, helping to identify and mitigate threats at each stage.

Tools for threat modeling:

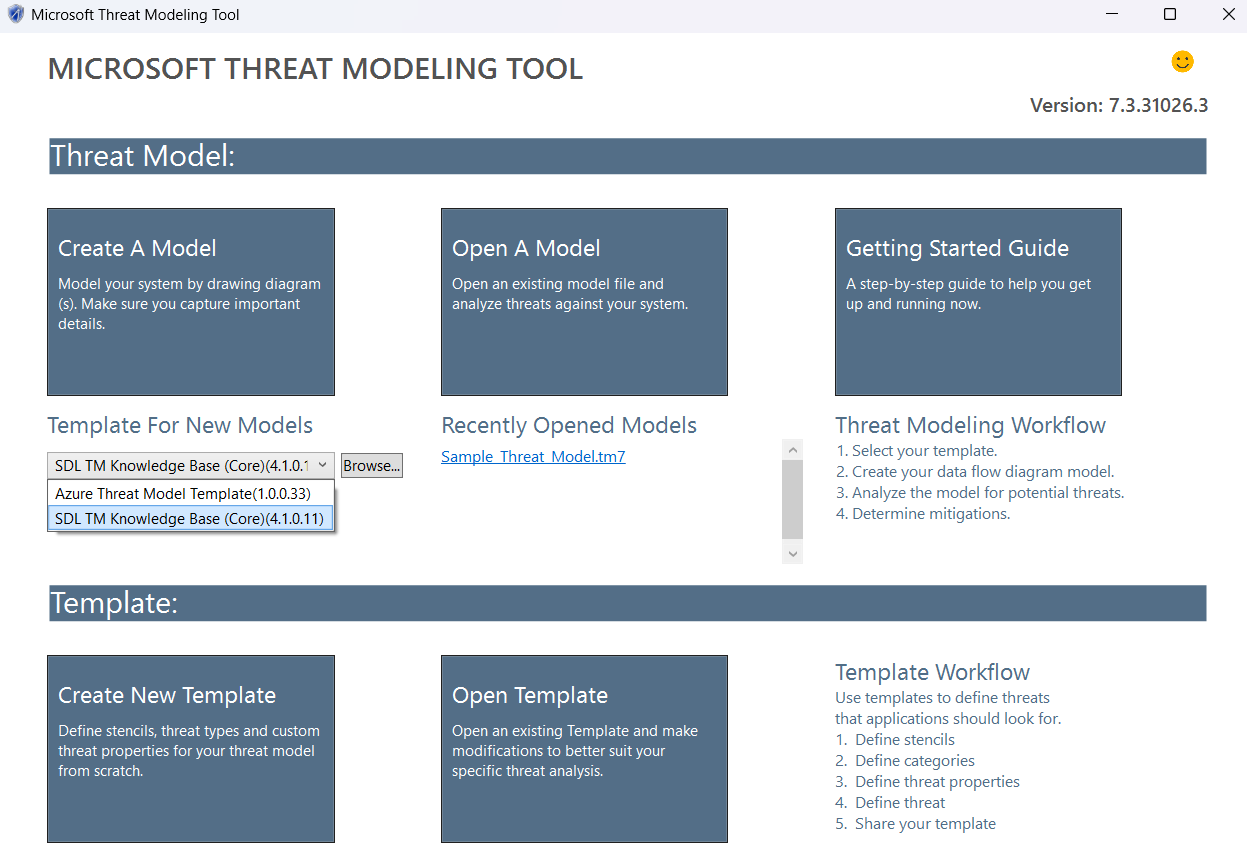
1. **Microsoft Threat Modeling Tool**: A user-friendly tool based on the STRIDE methodology, automating threat identification and mitigation.
2. **OWASP Threat Dragon**: An open-source tool that provides visual diagrams to model threats and generate reports, suitable for Agile environments.
3. **IriusRisk**: A collaborative tool that integrates with development workflows, offering automated threat modeling and risk assessment.
4. **ThreatModeler**: An enterprise-level tool that automates threat modeling, integrates with CI/CD pipelines, and supports multiple frameworks like STRIDE and PASTA.
5. **CAIRIS**: An open-source tool designed for security and usability engineers, offering risk analysis and modeling features.
6. **SecuriCAD**: A tool that simulates attack scenarios on system architectures to identify vulnerabilities and recommend mitigations.
7. **Trike**: A framework that includes a specific toolset for creating threat models and conducting risk assessments, focusing on stakeholder security expectations.
8. **PyTM**: A Python-based library that allows for programmatic creation of threat models, enabling integration with automated testing frameworks.
9. **ArcherySec**: An open-source vulnerability assessment and management tool that includes threat modeling capabilities.
10. **CVSS Calculator**: A tool for calculating the severity of vulnerabilities based on the Common Vulnerability Scoring System, often used alongside other threat modeling tools.

**Practical:**

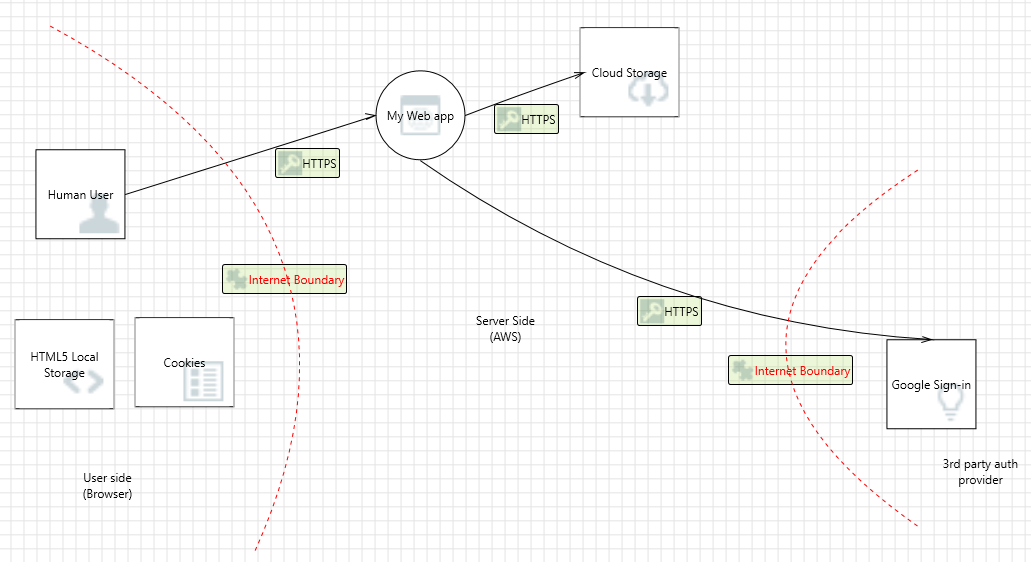
Step 1: Install MS Threat Modeling Tool



Step 2: Choose template for threat model



Step 3: Create your app diagram and generate report



### **Conclusion:** The experiment demonstrated the effectiveness of the Microsoft Threat Modeling Tool in identifying and mitigating security threats. It highlighted the importance of using threat modeling early in development to enhance system security.