1. What is Threat Modelling:

* Threat Modelling: Systematically listing all potential ways one can attack an application.
  + It’s a systematic approach which is repeatable.
  + Actively detecting different ways how something can be exploited.
    - List all Threats
    - Possible Threat Scenarios
  + It’s a holistic approach of reducing risks. The environment as a whole is considered and not just one individual component of the software.

* Weakness: Software Defects

* Vulnerability: Weakness which can be exploited.

* Attack Components:
  + Target: Something of value which is impacted
  + Attack Vector: Path attacker takes to exploit the vulnerability. For eg. The website from which the attacker will inject SQL commands
  + Threat Actor/Source

* Attack Surface:
  + Anything that can be obtained, used or attacked by a threat actor. For e.g. If a website has a dev environment

* Risk = Impact \* Likelihood
  + Impact: Negative Outcome of an attack/successful exploitation of an attack.
  + Likelihood: Probability of something happening

**Why should one perform Threat Modelling:**

* Provides risk reduction by the following.
  + Proactive preparation for potential threats.
  + Helps to detect bugs early one, which save money and efforts later

* Output of Threat Modelling:
  + It provides a list of potential threats and risks. It helps to prioritise workload.
  + Data Flow diagrams: Provides better understanding of security threats to the application
  + Security requirements and non requirements

**Who should Threat Model:**

* Security Architect
* Developer/Tester
* Security Professional

Threat Models should make as less changes possible to existing processes. Smaller the change, easier to replicate the process.

When to perform Threat modeling

As early as possible:

* Ideally during design/requirement phase.
* During each sprints
* Once application is

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | 1. Choosing the right Approach:        * Approach: Where to start      * Methodologies: The entire process      * Each methodology is based on the right approach.     Approaches for threat modelling:   * Asset Centric/Risk Centric * Attacker Centric * Application Centric       **Asset Centric/Risk Centric:**     * Create list of Assets we are protecting. * Draw Assets, components and data flows (give info on how assets are interconnected). * For each element/assets, check for threats.     Advantages:   * Focussed towards business impact * Well suited for risk assessment for auditors.   + Pasta is a risk centric approach   + Trike     Disadvantages:   * Not centred around applications * Assets don’t map to threat one to one.   + As time is a constraint it can be best used to map threats rather than assets.         **Attacker Centric/Security Centric:**     * Threat Actor * Motive * Financial and technical Means * Opportunity (How attacker will reach the goal to exploit the vulnerability)     Advantages:     * Make threats and attacks visible     Disadvantages:     * Easy to miss technical threats * Unrealistic threats * Biased results (if process is repeated we will have different results) * Need special skillsets     **Application Centric Approach:**    Step 1: Visualise the application. Focus on the following:   * The processes involved * Interacting entities * Dataflows * Different users     For eg:          Step 2: List threats using Threat Classification Models. For eg. Stride, OWASP TOP 10    Step 3: Rank the threats using the Threat Classification Model      Advantages:     1. Improves understanding and spread of knowledge     Disadvantages:     1. Documentation is necessary 2. Threats may sound abstract as       Choosing the right approach depends on the skillset available in your team and the type of application      **Methodologies :**      **PASTA:**     * Process for Attack Simulation and Threat Aalysis * It is a both Threat Modelling and Threat Anaysis Modelling (using Threat Intel) * Asset Centric Approach * 7 Stages:   1. Define Business Objectives   2. Define Technical Scope   3. Decompose Application (Data flow diagram of Application)   4. Threat Intelligence/Analysis (obtain inputs from different source: current state of threat actors, there motives, means and capability)   5. Identify vulenrabilities and rank them.   6. Enumerate attacks (Attack Trees: attacks mapped with assets and vulnerabilities)   7. Impact analysis (perform risk analysis whether risks are acceptable to businesses or not\_ * Targeted towards:   1. Medium to large mature companies   2. Having security knowledge * Executive Sponsorship necessary * Iterative and mature process * Outcome is for management     Advantages:     * Great for business integration/it needs specific inputs * Tooling available     Disadvantages:     * Specialised skills necessary * Time consuming * Very verbose * Variable as depends on the intermediatory inputs         **Microsoft Threat Modelling:**     * Pure threat modelling system * Stride is a threat classification system used within Microsoft Threat Modelling * Focusses on technical risk * Developer driven * Application centric approach * Process:   + Identify assets   + Create architecture overview   + Decompose application   + Identify threats   + Documentation of threats   + Rating of threats (using risk classification system like Dread, cvss, owasp risk rating methodology) * Simple,lightweight * Practical and for development teams     Advantages:     * Easy to pick up * Can be integrated into SDLC * Flexible     Disadvantages:     * Practical than academic * Doesn’t include business risks     **Octave (Operationally critically threat, Asset and vulnerability evaluation):**     * Risk Analysis Framework * Focusses on organization(for large companies and its very versions) risks * Octave-s is a lightweight model for smaller companies * Octave allegro (requires less risk assessment knowledge and focus on information assets) * Asset centric * Process:   + Establish drivers for the organization   + Asset profiling   + Identify threats   + Mitigate risks       Advantages:     * Imrpves Risk aware cultute * Indepth * Creates organization wide risk overview * Flexible     Disadvantages:     * Large and complex * Lots of paperwork * Requires investment       **Trike:**     * It is a methodolgy as well as tool * High levels of automation are possible * Asset cenrtic * Process   Two phases : system analysis phase and security analysis phase.   * System analysis phase:   + System is described in this phase as well as high level security goals.   + Identify threats   + Investigate threats   + Identify mitiations   + Investigate mititgations * Targetted towards security auditting teams * Produces three models:   + Requirements model   + Implementation model   + Risk model     Advantages:     * Automatically generate threats * Consistent results * Built in tool     Disadvantages:     * Does not scale * Defunct       **VAST:**     * Visual agile simple threat modelling * Two types:   + Application threat model   + Operational threat model * Uses process flow diagram instead of data flow diagram * Targetted for agile companies     Advantages:   * Flexible * Scalable * Process flow diagram might be easier     Disadvantages:   * Not open methodolgy       To choose methodolgy depends on:   * Team * Organization risks and objectives * Goal |  |  |
|  |  | Irius threat modelling    Threat modelling:  Tools  Approach  Risk Analyis Framework    Intro  Experiences  Questions |
|  |  |  |

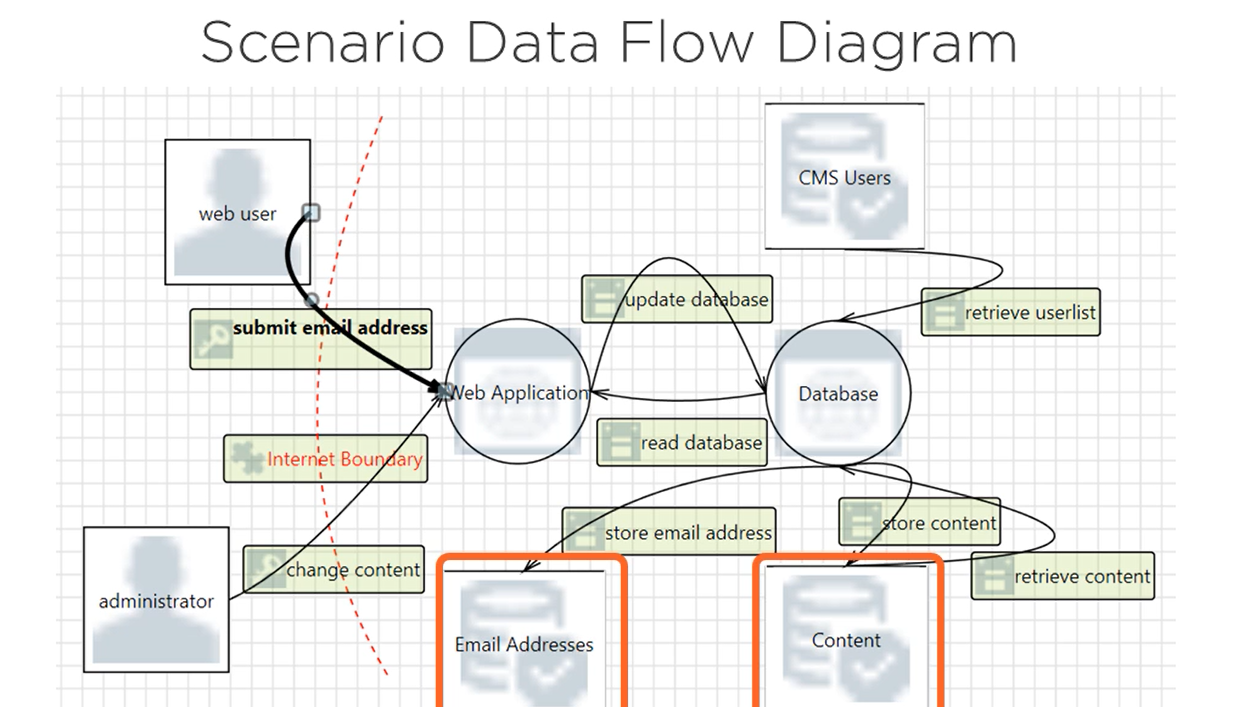
**Threat Modelling process example:**

* Steps
  + Set Scope: Assets/policies etc
  + Analyze target: Draw Data flow diagram to
  + Identify threats
  + Rate threats

* Scope:
  + List assets in scope and out of scope

* Data Flow Diagram:
* Reusable and simple to follow
* Data flow elements:
  + Process
  + Data flow : Denoted by downwards arrow
  + Data Store: Denoted by two parallel horizontal lines (content management data store and email addresses data store)
  + External Entity: denoted by square, these are out of scope
  + Trust Boundary: It is a boundary between elements having different permissions.

Eg.



* Diagram should match items in scope
* It should identify what application does, identify the technology

* Security Profile(part of application decomposing):

A statement of how the application handles various security sensitive areas( eg Session management/cryptography/auditing and logging)

* Documenting Threats:
  + Using STRIDE (its called a threat or attack classification models)
    - Each letter stands for a different kind of threat category
    - For eg. Spoofing of cookies is a threat
  + Other (threat or attack classification models).
    - Attack Libraries (list of existing attacks)
      * Mitre capec (common attack pattern enumeration and classification list) library
    - Attack trees: goals and sub goals of an attack and the attack methodologies
      * Formalized and methodological approach to find threats
    - Desist (Dispute, elevation of privilege etc)
  + Important things to note down during documentation of threats:
    - Target
    - Attack Technique
    - Countermeasure
    - Risk Rating

* Rating Threats
  + You can use a risk rating classification system
    - DREAD (Damage potential, reproducibility, exploitability, affected users, discoverability): High , medium , low
    - OWASP Risk Rating methodology
    - CVSS

* Summary of Threat Modelling Steps:
* 