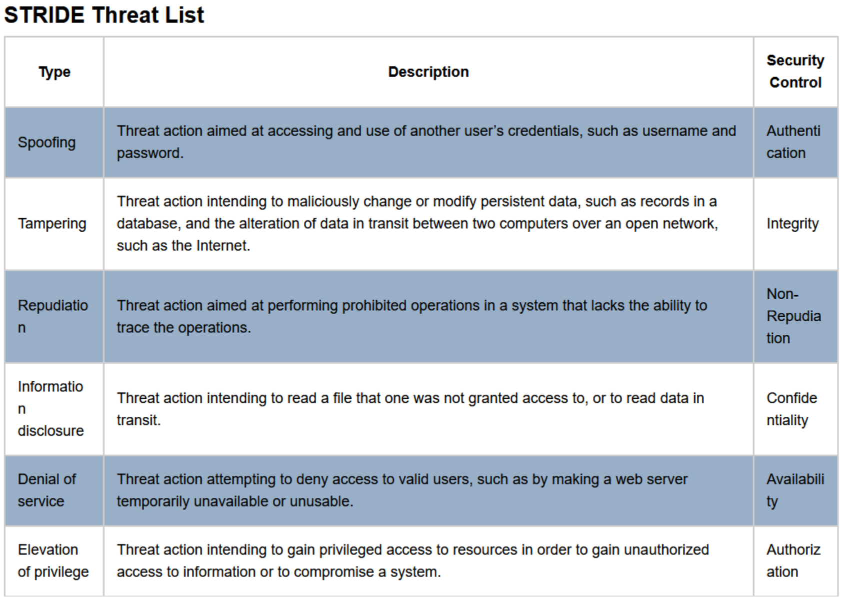
practice-2

Student: Tendikov Noyan Maratovich
Group: SSE-2401
Assignment: https://lms.astanait.edu.kz/mod/assign/view.php?id=5514

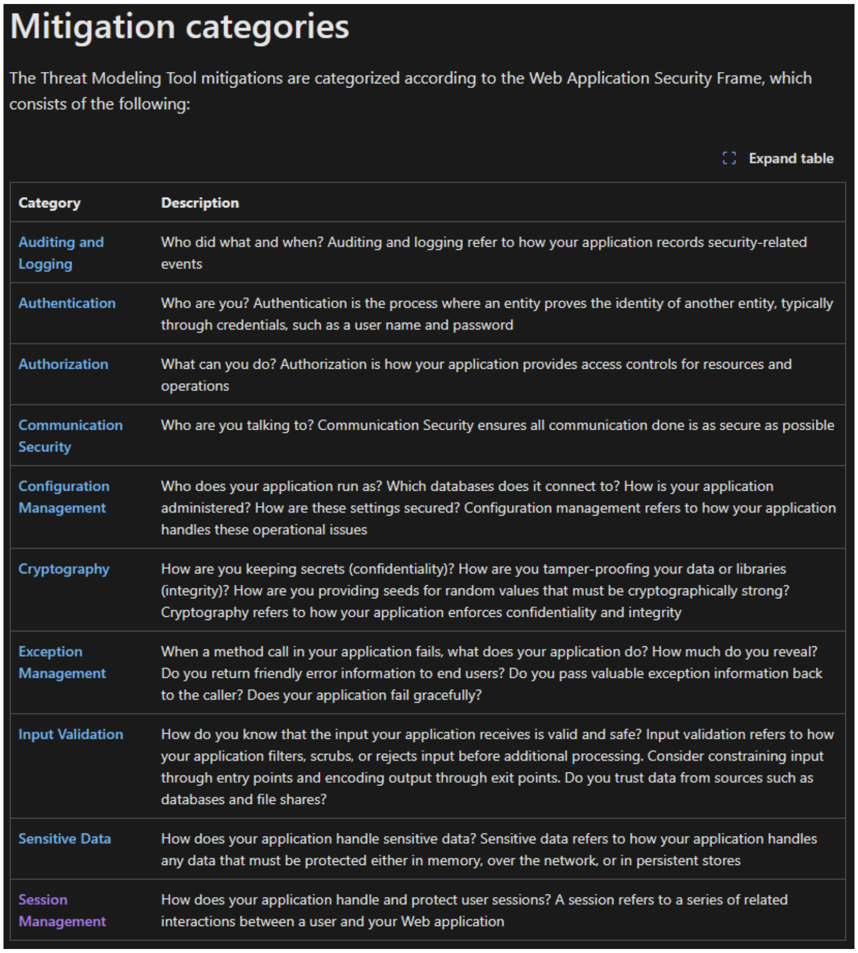
Description and theory

1. STRIDE - the model / framework for security attack surface modeling and risk management made by Microsoft. It identifies different categories of threats and their security controls / mitigation: spoofing, tampering, repudiation, information disclosure, denial of service, elevation of privilage.



STRIDE Threat List

Type	Description	Security Control
Spoofing	Threat action aimed at accessing and use of another user's credentials, such as username and password.	Authentication
Tampering	Threat action intending to maliciously change or modify persistent data, such as records in a database, and the alteration of data in transit between two computers over an open network, such as the Internet.	Integrity
Repudiation	Threat action aimed at performing prohibited operations in a system that lacks the ability to trace the operations.	Non-Repudiation
Information disclosure	Threat action intending to read a file that one was not granted access to, or to read data in transit.	Confidentiality
Denial of service	Threat action attempting to deny access to valid users, such as by making a web server temporarily unavailable or unusable.	Availability
Elevation of privilege	Threat action intending to gain privileged access to resources in order to gain unauthorized access to information or to compromise a system.	Authorization



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
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
Authorization	What can you do? Authorization is how your application provides access controls for resources and operations
Communication Security	Who are you talking to? Communication Security ensures all communication done is as secure as possible
Configuration Management	Who 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

Alternatives: PASTA (Process fot Attack Simulations and Threat Analysis) according to Risk Centric Threat Modeling book, DREAD, CIA, CIADIE, LINDDUN, PLOT4ai, etc.

Additional sources and resources: - https://owasp.org/www-community/Threat_Modeling_Process - https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-mitigations - https://cheatsheetseries.owasp.org/cheatsheets/Threat_Modeling_Cheat_Sheet.html - Standards: NIST SP 800-30 (Risk Assessment), NIST SP 800-53 (Security Controls), ISO/IEC 27005 (Risk Management). 2. Data Flow Diagram - notation for designing diagrams of business processes, but also applied well in threat modeling. - It provides a high-level understanding of the available resources and the direction of data. The main elements of DFD: entities / actors, processes, data stores, and data flows. - Similarly to C4 model, there are different levels of visualization (usually from 0 to 1, or even multi-level). - Various tools are used for visualization: Microsoft Threat Modeling Tool, OWASP Threat Dragon, draw.io, pseudocode, etc.

Solution and practice

1. System scope, description, and overview

- Given PyPI platform that hosts large amount of different Python packages. Similar systems in the category of software repositories are Maven Central, NPM, Crates.io, and many others.
- Source code and documentation: https://github.com/pypi/warehouse/
- Overall, the analysis was broken down into single features rather than the entire product at once. In addition, some elements and details were omitted (assumptions and presumptions were made), since not all architectural aspects of the systems are always known to us. In other words, it is not always possible to work with a white-box approach - sometimes we have to deal with black-box systems, where the specific technologies are unknown, which implies that we only have a high-level representation (focus on specification and general details rather than implementation and specific details). Overdetailing requires additional time and resources (and is more suitable for reverse engineering), but in threat modeling the primary objective is to obtain a representation model that is sufficient enough.

2. Data flow diagrams

- Firstly, we constructed level 0 DFD, as shown in image 3
- Secondly, based on level 0 we dig in details and for each feature provided subdiagrams of level 1 DFD

3. STRIDE analysis mirrors as threats For each DFD element (process, flow, data store, entity), perform STRIDE analysis: Spoofing, Tampering, Repudiation, Information disclosure, Denial of Service, Elevation of privilege. For each identified threat, document: short description, potential attack vectors / prerequisites, associated DFD element, likelihood (H/M/L) and impact (H/M/L). Deliverable: a STRIDE table, e.g.: DFD Element | STRIDE Category | Threat Description | Attack Vector | Likelihood | Impact https://peps.python.org/pep-0449/

4. Attack scenario (Kill Chain / MITRE ATT&CK) Select 1–2 high-risk threats from the STRIDE analysis. (detailed attack scenarios) Expand them into a detailed attack scenario using either: Lockheed Martin Cyber Kill Chain (Recon → Weaponization → Delivery → Exploitation → Installation → C2 → Objectives), or MITRE ATT&CK tactics/techniques (with IDs if possible). For each step, specify: attacker goal, techniques used, possible Indicators of Compromise (IoCs), and detection opportunities. Example Scenarios Phishing campaign → stolen admin credentials → lateral movement → database exfiltration. Insecure CI/CD pipeline → malicious code injection → supply-chain compromise MITRE ATT&CK Framework: https://attack.mitre.org/ MITRE D3FFENSE Framework: https://d3fend.mitre.org/ CWE List:

https://cwe.mitre.org/data/index.html systematic threat and attack modeling at the system architecture level: model the attack lifecycle using Kill Chain, justify the scenario's plausibility, assess risks, and propose technical and organizational countermeasures. Corporate web application with API and database. Cloud environment (Kubernetes containers with microservices).

5. Risk assessment, mitigation and counter measures Assess each scenario's risk using a simple matrix: Risk = Likelihood × Impact (H/M/L). Propose technical controls (MFA, WAF, segmentation, TLS 1.3, patches) and organizational measures (policies, awareness training, logging). For each measure, explain (justifications): expected risk reduction, implementation effort, and validation metrics. Risk assessment matrix Recommendations for monitoring and detection. Countermeasures and implementation plan.