SSD Project threat modeling

Technical Threat report

Wed Apr 23 2025 09:21:21 GMT+0000 (Coordinated Universal Time)

Description: *No description*

Filtered by: *No filters*

Workflow status: Draft

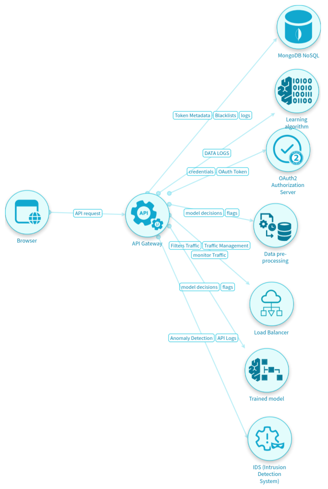
Project tags: *No value*

Unique ID: ssd-project-threat-modeling-1745397657250

Owner: Ahmed Amjad

Updated: Apr 23, 2025, 9:15 AM

Project number: ssd-project-threat-modeling-1745397657250



Content menu

Risk Mitigation Summary

Risk summary

Current risk level compared with the inherent and projected risk levels.

Inherent Risk:	70%
Current Risk:	9%
Projected Risk:	9%

Risk distribution

Number of threats in each risk rating level.

Very Low:	37
Low:	0
Medium:	7
High:	1
Critical:	0

Unmitigated : 0 Partly-Mitigated : 1 Mitigated : 0 Accepted : 0 N/A : 0

Vulnerabilities: 47 **Total number of threats:** 45

Countermeasures state

Implemented:	51
Required:	0
Recommended:	5

Rejected : 0 N/A : 0

Failed	0	Verified	0

Unmitigated Threats

Listed below are threats per component where all countermeasures are not implemented or the weaknesses test result failed.

Component: Browser

Threats of this component

Use case: CCPA Requirements

UT1. Threat name: Client application does not offer any mechanism to let the user exercise their Right to Correct

- **Inherent risk:** Medium
- **Current risk:** Medium
- **Projected risk:** Medium
- **Planned mitigation:** 0%

UT2. Threat name: Client application does not offer any mechanism to let the user exercise their Right to Delete

- **Inherent risk:** Medium
- **Current risk:** Medium
- **Projected risk:** Medium
- **Planned mitigation:** 0%

UT3. Threat name: Client application does not offer any mechanism to let the user exercise their Right to Know

- **Inherent risk:** Medium
- **Current risk:** Medium
- **Projected risk:** Medium
- **Planned mitigation:** 0%

Component: MongoDB NoSQL

Threats of this component

Use case: Tampering

UT4. Threat name: Insider threats can compromise data integrity

- **Inherent risk:** High
- **Current risk:** High
- **Projected risk:** High
- **Planned mitigation:** 0%

Component: OAuth2 Authorization Server

Threats of this component

Use case: CCPA Requirements

UT5. Threat name: Server application does not offer any mechanism to let the user exercise their Right to Correct

- **Inherent risk:** Medium
- **Current risk:** Medium
- **Projected risk:** Medium
- **Planned mitigation:** 0%

UT6. Threat name: Server application does not offer any mechanism to let the user exercise their Right to Delete

- **Inherent risk:** Medium
- **Current risk:** Medium
- **Projected risk:** Medium
- **Planned mitigation:** 0%

UT7. Threat name: Server application does not offer any mechanism to let the user exercise their Right to Know

- **Inherent risk:** Medium
- **Current risk:** Medium
- **Projected risk:** Medium
- **Planned mitigation:** 0%

Partly Mitigated Threats

Listed below are threats per component where some countermeasures are not implemented or the weaknesses test result failed.

Component: Browser

Threats of this component

Use case: **Spoofing**

PMT1. Threat name: Attackers conduct phishing attacks through deceptive websites

- **Inherent risk:** Critical
- **Current risk:** Medium
- **Projected risk:** Medium
- **Planned mitigation:** 50%

Mitigated Threats

Below is the list of threats per component where all countermeasures are implemented and have passed their tests and there are no failed weakness tests.

Component: API Gateway

Threats of this component

Use case: Elevation of Privilege

MT1. Threat name: Authentication bypass

- **Inherent risk:** High
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

Use case: Repudiation

MT2. Threat name: Exploitation of insufficient logging and monitoring

- **Inherent risk:** High
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

Component: Browser

Threats of this component

Use case: Elevation of Privilege

MT3. Threat name: Attackers exploit browser vulnerabilities to execute malicious code

- **Inherent risk:** Critical
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

Use case: Information Disclosure

MT4. Threat name: Attackers inject malicious scripts via cross-site scripting (XSS)

- **Inherent risk:** High
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

Use case: Spoofing

MT5. Threat name: Attackers intercept browser communications through man-in-the-middle (MitM) attacks

- **Inherent risk:** Critical
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

Use case: Tampering

MT6. Threat name: Attackers distribute malware through compromised browser extensions

- **Inherent risk:** Critical
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

Component: Data pre-processing

Threats of this component

Use case: Tampering

MT7. Threat name: Changing data distribution and properties will affect the performance of the future model

- **Inherent risk:** High
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

MT8. Threat name: Data encoding, normalization, filtering, feature selection, and annotation, may all introduce biases or affect the predictive and generalization qualities of the model

- **Inherent risk:** Critical
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

Component: IDS (Intrusion Detection System)

Threats of this component

Use case: Elevation of Privilege

MT9. Threat name: Accessing functionality not properly constrained by ACLs

- **Inherent risk:** High
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

Use case: Information Disclosure

MT10. Threat name: Attackers gain access to the system and are not detected

- **Inherent risk:** High
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

MT11. Threat name: Attackers gain access to unauthorised data by exploiting vulnerabilities in the service

- **Inherent risk:** High
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

Component: Learning algorithm

Threats of this component

Use case: Information Disclosure

MT12. Threat name: Adversaries may exploit algorithmic leakage or data representation issues to extract data or attack the model

- **Inherent risk:** Critical
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

Use case: Tampering

MT13. Threat name: An adversary may be able to manipulate an online learning system

- **Inherent risk:** Critical
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

MT14. Threat name: An adversary may be able to manipulate model parameters or hyperparameters

- **Inherent risk:** Critical
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

Component: Load Balancer

Threats of this component

Use case: Denial of Service

MT15. Threat name: Attackers use DDoS attacks to overwhelm the Load Balancer

- **Inherent risk:** High
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

MT16. Threat name: Attackers use resource exhaustion tactics to overwhelm the Load Balancer

- **Inherent risk:** High
- **Current risk:** Very Low

- **Projected risk:** Very Low
- **Planned mitigation:** 100%

Use case: Information Disclosure

MT17. Threat name: Attackers can intercept traffic through Load Balancer vulnerabilities

- **Inherent risk:** High
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

MT18. Threat name: Attackers exploit security misconfigurations

- **Inherent risk:** Critical
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

Use case: Spoofing

MT19. Threat name: Attackers can hijack sessions by gaining unauthorized access to a user's active session

- **Inherent risk:** Critical
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

Component: MongoDB NoSQL

Threats of this component

Use case: Denial of Service

MT20. Threat name: Attackers can overload the database with DoS attacks

- **Inherent risk:** High
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

Use case: Elevation of Privilege

MT21. Threat name: Attackers can exploit default configurations

- **Inherent risk:** Critical
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

MT22. Threat name: Attackers can gain unauthorized access

- **Inherent risk:** High
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

MT23. Threat name: Malicious users can escalate privileges

- **Inherent risk:** Medium
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

Use case: Information Disclosure

MT24. Threat name: Attackers can access unsecured backups

- **Inherent risk:** High
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

MT25. Threat name: Attackers can intercept unencrypted data

- **Inherent risk:** Critical
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

Use case: Tampering

MT26. Threat name: Attackers take advantage of injection vulnerabilities

- **Inherent risk:** High
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

Component: OAuth2 Authorization Server

Threats of this component

Use case: Elevation of Privilege

MT27. Threat name: Attackers exploit weak access policies

- **Inherent risk:** Critical
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

Use case: Information Disclosure

MT28. Threat name: Attackers exploit open redirects

- **Inherent risk:** Medium
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

MT29. Threat name: Attackers intercept tokens over unencrypted communication channels

- **Inherent risk:** High
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

Use case: Repudiation

MT30. Threat name: Attackers bypass detection due to lack of logging and monitoring

- **Inherent risk:** Medium
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

Use case: Spoofing

MT31. Threat name: Attackers inject malicious authorization codes

- **Inherent risk:** High
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

MT32. Threat name: Attackers perform brute force on client credentials

- **Inherent risk:** Critical
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

MT33. Threat name: Attackers perform token replay attacks

- **Inherent risk:** High
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

Use case: Tampering

MT34. Threat name: Attackers exploit insufficient token validation

- **Inherent risk:** High
- **Current risk:** Very Low
- **Projected risk:** Very Low
- **Planned mitigation:** 100%

Component: Trained model

Threats of this component

Use case: Information Disclosure

MT35. Threat name: An adversary may be able to reveal sensitive information available in the model or the data used to build it

- **Inherent risk:** ⌃ Critical
- **Current risk:** ⌄ Very Low
- **Projected risk:** ⌄ Very Low
- **Planned mitigation:** 100%

Use case: Repudiation

MT36. Threat name: Lack of sufficient details about the model (algorithm), its architecture, or its data

- **Inherent risk:** ^ High
- **Current risk:** ⌄ Very Low
- **Projected risk:** ⌄ Very Low
- **Planned mitigation:** 100%

Use case: Tampering

MT37. Threat name: An adversary may take advantage of model sharing and/or transfer

- **Inherent risk:** ⌃ Critical
- **Current risk:** ⌄ Very Low
- **Projected risk:** ⌄ Very Low
- **Planned mitigation:** 100%

Accepted Threats

List of accepted threats per component, with the corresponding reason for acceptance.

No Data

Not-Applicable Threats

List of not-applicable threats per component, with the corresponding reason for not-applicability.

No Data

Failed Countermeasure Tests

No Data

Appendix A: Threat Details

The number of threats in each risk rating level.

Component: API Gateway

Threats of this component in each risk rating level

MT1. Threat name: Authentication bypass [T-API-GATEWAY-T-API-GW-01]

- **State:** Mitigate
- **Description:** **General threat description**
  An attack scenario in which malicious actors find ways to circumvent the authentication mechanisms, leading to unapproved access to API functionalities.
  Threat agents/Attack vectors
  Attackers may exploit vulnerabilities in the authentication process or use stolen credentials to gain access without proper authorization.
  Impacts
  Such actions can result in unauthorized data access, data breaches, or service misuse, compromising both security and privacy.
  Example Attack Scenarios
  An attacker could intercept unencrypted credentials or leverage a brute-force attack to bypass login controls, gaining entry to sensitive API endpoints without detection.
- **C0. Countermeasure name:** The API gateway should have a connector to an artifact that can generate an access token for the client request
  - **State:** IMPLEMENTED
- **C1. Countermeasure name:** Connectors should be provided for integrating with identity providers (IdPs)
  - **State:** IMPLEMENTED
- **C2. Countermeasure name:** Integrate the API gateway with an identity management application
  - **State:** IMPLEMENTED
- **C3. Countermeasure name:** Distributed gateway deployments should have a token translation (exchange) service between gateways
  - **State:** IMPLEMENTED

MT2. Threat name: Exploitation of insufficient logging and monitoring [T-API-GATEWAY-T-API-GW-02]

- **State:** Mitigate
- **Description:**

General threat description
The absence of adequate logging and monitoring allows attackers to execute malicious actions without detection. This can lead to unauthorized access and potential data breaches with the exploitation remaining hidden from defenders.
Threat agents/Attack vectors
Cybercriminals may exploit this weakness by injecting malicious code into the API gateway or by initiating a series of unauthorized requests. Without proper monitoring, these activities can go unnoticed.
Impacts
This threat can lead to compromised systems, unauthorized data access, and potential data loss or manipulation. Moreover, it may result in delayed response to breaches and increased recovery costs.
Example Attack Scenarios
An attacker successfully bypasses authentication controls and gains access to sensitive data through the API gateway. Due to the absence of logging, the attacker maintains access over an extended period, exfiltrating data without triggering alerts.

- **C4. Countermeasure name:** Securely channel all traffic information to a monitoring and/or analytics application
  - **State:** IMPLEMENTED

Component: Browser

UT1. Threat name: Client application does not offer any mechanism to let the user exercise their Right to Correct [T-CCPA-CLIENT-SIDE-RIGHT-TO-CORRECT]

- **State:** Expose
- **Description:**

If a client application does not offer any mechanism to let the user exercise their Right to Correct the user is unable to correct any inaccurate personal information that the business has collected about them. This can lead to incorrect decisions being made about users, such as denying them access to services or products, or even exposing them to identity theft or fraud.

- **C5. Countermeasure name:** Develop an online form that users can fill out to submit their request
  - **State:** RECOMMENDED

UT2. Threat name: Client application does not offer any mechanism to let the user exercise their Right to Delete [T-CCPA-CLIENT-SIDE-RIGHT-TO-DELETE]

- **State:** Expose
- **Description:**

If a client application does not offer any mechanism to let the user exercise their Right to Delete the user is unable to delete any personal information that the business has collected about them. This can be a threat to the user's privacy and security because personal information can be used for malicious purposes, such as identity theft or fraud.

- **C6. Countermeasure name:** Develop an online form that users can fill out to submit their request
  - **State:** RECOMMENDED

UT3. Threat name: Client application does not offer any mechanism to let the user exercise their Right to Know [T-CCPA-CLIENT-SIDE-RIGHT-TO-KNOW]

- **State:** Expose
- **Description:**

If a client application does not offer any mechanism to let the user exercise their Right to Know the user is unable to know what personal information the business has collected about them and how it has been used. This can be a threat to the user's privacy and security because they may not be aware of how their personal information is being used or shared, which can lead to unwanted solicitations or even identity theft.

- **C7. Countermeasure name:** Inform the user about which data will be collected
  - **State:** RECOMMENDED
- **C8. Countermeasure name:** Develop an online form that users can fill out to submit their request
  - **State:** RECOMMENDED

PMT1. Threat name: Attackers conduct phishing attacks through deceptive websites [T-BROWSER-T-BROWSER-02]

- **State:** Partly-Mitigated
- **Description:** **General Threat Description:**
  Adversaries create deceptive websites that mimic legitimate ones to trick users into revealing sensitive information.
  Threat Agents/Attack Vectors:
  Cybercriminals using social engineering techniques
  Fake websites or compromised legitimate sites hosting phishing pages
  Impacts:
  Credential theft and identity compromise
  Unauthorized access to sensitive accounts or systems
  Financial loss and reputational damage
  Example Attack Scenarios:
  A user receives an email with a link to a fake banking website that looks identical to the real one, prompting them to enter their login details.
  An attacker registers a domain similar to a popular e-commerce site and lures users into entering credit card information during checkout.
- **C9. Countermeasure name:** Deploy anti-phishing protection
  - **State:** RECOMMENDED
- **C10. Countermeasure name:** Activate URL filtering mechanisms
  - **State:** IMPLEMENTED

MT3. Threat name: Attackers exploit browser vulnerabilities to execute malicious code [T-BROWSER-T-BROWSER-04]

- **State:** ✓ Mitigate
- **Description:** **General Threat Description:**
  Adversaries leverage flaws in the browser's code to run unauthorized code, bypassing security measures and potentially compromising the entire system.
  Threat Agents/Attack Vectors:
  Cybercriminals targeting known or zero-day browser vulnerabilities
  Malicious websites and compromised ads delivering exploit code
  Exploited browser extensions or plugins
  Impacts:
  Unauthorized system access and control
  Data theft or manipulation
  Escalation of privileges on the host system
  Example Attack Scenarios:
  An attacker uses a zero-day exploit on a popular browser via a compromised website, leading to malware installation.
  A malicious browser extension exploits a known vulnerability to gain remote access and steal sensitive data.
- **C11. Countermeasure name:** Apply security hardening measures
  - **State:** IMPLEMENTED
- **C12. Countermeasure name:** Configure automatic browser updates
  - **State:** IMPLEMENTED

MT4. Threat name: Attackers inject malicious scripts via cross-site scripting (XSS) [T-BROWSER-T-BROWSER-01]

- **State:** ✓ Mitigate
- **Description:** **General Threat Description:**
  Adversaries exploit vulnerabilities in web applications and browsers to inject malicious scripts, which then execute in users' browsers.
  Threat Agents/Attack Vectors:
  Cybercriminals exploiting unvalidated input fields
  Compromised or malicious websites hosting injected scripts
  Impacts:
  Theft of session data and credentials
  Unauthorized access to sensitive user information
  Redirection to phishing or malicious sites
  Example Attack Scenarios:
  An attacker injects a script into a forum post that steals users' cookies when viewed.
  A vulnerable web form accepts unfiltered input, allowing an attacker to embed a script that executes upon page load.
- **C13. Countermeasure name:** Activate built-in browser security filters
  - **State:** IMPLEMENTED
- **C14. Countermeasure name:** Implement client-side script blockers
  - **State:** IMPLEMENTED

MT5. Threat name: Attackers intercept browser communications through man-in-the-middle (MitM) attacks [T-BROWSER-T-BROWSER-03]

- **State:** ✓ Mitigate
- **Description:** **General Threat Description:**
  Adversaries intercept and potentially alter communication between browsers and websites by exploiting insecure or misconfigured network protocols.
  Threat Agents/Attack Vectors:
  Cybercriminals targeting unsecured Wi-Fi networks or misconfigured network devices
  Attackers exploiting weak TLS/SSL configurations
  Use of rogue access points or compromised routers
  Impacts:
  Interception of sensitive data such as credentials and personal information
  Data manipulation or session hijacking
  Unauthorized access to private communications
  Example Attack Scenarios:
  An attacker sets up a rogue Wi-Fi hotspot to capture unencrypted browser traffic from unsuspecting users.
  Exploiting a vulnerability in TLS certificate validation, an attacker intercepts and modifies data transmitted between a user and a secure website.
- **C15. Countermeasure name:** Utilize encrypted communication tools
  - **State:** IMPLEMENTED
- **C16. Countermeasure name:** Enforce strict certificate validation
  - **State:** IMPLEMENTED

MT6. Threat name: Attackers distribute malware through compromised browser extensions [T-BROWSER-T-BROWSER-05]

- **State:** ✓ Mitigate
- **Description:** **General Threat Description:**
  Adversaries exploit or compromise browser extensions to distribute malware, leveraging the trust users place in these add-ons to execute malicious code within the browser environment.

Threat Agents/Attack Vectors:
Cybercriminals submitting malicious extensions to official stores or hijacking updates of legitimate ones.
Social engineering tactics encouraging users to install unverified or counterfeit extensions.
Impacts:
Unauthorized access to browser data and credentials
Installation of malware that may compromise the system
Potential lateral movement within a network through compromised systems
Example Attack Scenarios:
An attacker uploads a seemingly useful extension to a browser store, which, once installed, quietly collects sensitive data.
A legitimate extension is compromised during an update, injecting malware that hijacks user sessions and exfiltrates data.

- **C17. Countermeasure name:** Implement extension whitelisting policies
  - **State:** IMPLEMENTED
- **C18. Countermeasure name:** Manage browser extensions securely
  - **State:** IMPLEMENTED

Component: Data pre-processing

MT7. Threat name: Changing data distribution and properties will affect the performance of the future model [T-ML-AI-CHANGING-DATA]

- **State:** Mitigate
- **Description:**

General Threat Description: This threat concerns the potential impact of evolving data properties and distribution on the performance of machine learning (ML) and artificial intelligence (AI) systems. As data evolves over time, its distribution may change due to shifts in underlying patterns, behaviors, or external factors influencing the data generation process. These changes can affect how well an ML model, initially trained on a specific dataset, performs when exposed to new data. If not anticipated and managed appropriately, such changes can degrade model accuracy and utility, leading to poor decision-making.
Threat Agents/Attack Vectors: While often a natural consequence of dynamic environments, deliberate manipulation of data distribution by adversaries can also induce this threat. Attack vectors include data poisoning, where malicious inputs are introduced to subtly shift the data distribution, and model evasion techniques, where attackers craft inputs specifically designed to exploit model weaknesses that emerge as data changes.
Impacts: The impacts include reduced model reliability, increased error rates, and potentially costly adjustments such as retraining or redesigning the model. In critical applications, such as medical diagnostics or financial forecasting, these impacts can translate into significant financial losses, endangerment of lives, and loss of credibility.
Example Attack Scenarios:
A financial fraud detection model gradually becomes ineffective as fraud techniques evolve, leading to increased false negatives and significant financial losses due to undetected fraudulent transactions.
Attackers manipulate the data input to a real-time traffic management system during a major public event, causing the system to mispredict traffic flows and leading to congestion or safety incidents.

- **C19. Countermeasure name:** Monitor for data drift, especially for new data and future online models
  - **State:** IMPLEMENTED

MT8. Threat name: Data encoding, normalization, filtering, feature selection, and annotation, may all introduce biases or affect the predictive and generalization qualities of the model [T-ML-AI-DATA-PRE-PROCESSING]

- **State:** Mitigate
- **Description:**

General Threat Description: This threat concerns the potential introduction of biases or the compromise of predictive accuracy during the data preparation stages of ML/AI model development. Key processes such as data encoding, normalization, filtering, feature selection, and annotation are critical in transforming raw data into a format suitable for learning algorithms. If improperly managed, these processes can inadvertently introduce biases or distortions that compromise the model's performance and security. For instance, poor feature engineering might oversimplify complex data, inadequate encoding could lead to information loss, and improper normalization may skew the data distribution, affecting the model's ability to generalize from training to real-world application. Important aspects to take into consideration are:
Feature discovery/engineering, which relies on the type of data.
Encoding: both data and feature encoding should be carefully considered, e.g., to avoid information loss.
Normalization: essentially to use a common scale throughout the data.
Partitioning: care must be taken when creating data partitions.
Other data manipulation or helpers: such as data filters or randomness, which plays an important role in stochastic systems.
Threat Agents/Attack Vectors: Threat agents can range from data scientists who make unintentional errors to malicious insiders or external attackers who deliberately manipulate preprocessing steps. Attack vectors include the introduction of biased or skewed data at the input stage, manipulation of data handling rules, and exploitation of weaknesses in data preprocessing algorithms.
Impacts: The impacts of such biases and distortions include reduced model accuracy, unfair outcomes (especially in sensitive applications like hiring or law enforcement), and decreased user trust. In critical systems, these issues can lead to significant operational risks, regulatory non-compliance, and reputational damage.
Example Attack Scenarios:
A data scientist unintentionally uses an inappropriate method for data normalization in a credit scoring model, causing certain demographic groups to be unfairly penalized based on their spending behaviors which are misinterpreted by the model.
An attacker manipulates the feature selection process in a healthcare diagnostic tool, causing it to ignore critical indicators of a specific disease, leading to widespread misdiagnoses and inadequate patient care.

- **C20. Countermeasure name:** Maintain data quality and integrity during data pre-processing
  - **State:** IMPLEMENTED
- **C21. Countermeasure name:** Identify potential bias in the data
  - **State:** IMPLEMENTED

Component: IDS (Intrusion Detection System)

MT10. Threat name: Attackers gain access to the system and are not detected [T-IDS-INTRUSION-DETECTION-SYSTEM-T-IDS-02]

- **State:** Mitigate
- **Description:**

General threat description
Intruders accessing the system may go undetected if the IDS is not properly configured or monitored.
Threat agents/Attack vectors
Unmonitored or misconfigured IDS components can be exploited by attackers to infiltrate network systems without being flagged.
Impacts
This could lead to unauthorized access to sensitive data, disruption of services, and potential damage to the network infrastructure.
Example Attack Scenarios
An attacker uses legitimate credentials obtained through phishing to access the network. Due to improper IDS configuration, these activities are not detected, allowing the attacker to exfiltrate confidential information.

- **C22. Countermeasure name:** Configure the IDS to send alerts to a central location

- **State:** IMPLEMENTED

MT11. Threat name: Attackers gain access to unauthorised data by exploiting vulnerabilities in the service [T-IDS-INTRUSION-DETECTION-SYSTEM-T-IDS-01]

- **State:** ✓ Mitigate
- **Description:**

General threat description
Attackers may exploit known vulnerabilities in the intrusion detection system, leading to unauthorized data access.
Threat agents/Attack vectors
Attackers might leverage publicly available information on vulnerabilities, social engineering, or miss-configuration within the IDS to gain unauthorized access.
Impacts
The exploitation of vulnerabilities can result in data breaches, information theft, or the compromise of sensitive or critical information systems.
Example Attack Scenarios
An attacker might find a vulnerability in the IDS software that has not been patched by the organization, allowing them to bypass detection controls and access confidential data. Alternatively, they could use social engineering tactics to trick an insider into enabling an exploit that grants unauthorized access to sensitive information.

- **C23. Countermeasure name:** Update IDS regularly
  - **State:** IMPLEMENTED

MT9. Threat name: Accessing functionality not properly constrained by ACLs [T-IDS-INTRUSION-DETECTION-SYSTEM-T-IDS-03]

- **State:** ✓ Mitigate
- **Description:**

General threat description
An incorrectly set up Access Control List can create vulnerabilities by permitting unauthorized access to the components and functionalities of the Intrusion Detection System.
Threat agents/Attack vectors
Malicious actors may exploit misconfigurations in the Access Control Lists through network access, leveraging weak or default settings to gain unauthorized control or information.
Impacts
The compromise of the IDS can lead to unauthorized data exposure, manipulation of IDS settings, interruption of monitoring capabilities, and potential exploitation of other network vulnerabilities.
Example Attack Scenarios
An attacker gains access to the IDS management interface due to a weak or incorrectly configured ACL, allowing them to disable alerts for specific attacks, delete logs, or conduct malicious activities unnoticed within the network.

- **C24. Countermeasure name:** Use an out-of-band management connection for IDS
  - **State:** IMPLEMENTED

Component: Learning algorithm

MT12. Threat name: Adversaries may exploit algorithmic leakage or data representation issues to extract data or attack the model [T-ML-AI-DATA-REPRESENTATION]

- **State:** ✓ Mitigate
- **Description:**

General Threat Description: This threat revolves around the exploitation of vulnerabilities inherent in the algorithms and data representation methods used within AI and machine learning systems. Certain algorithms may inadvertently leak information if they are not suited to handle confidential data securely. Such vulnerabilities might arise from how algorithms represent and store training data, as well as from initial model configurations that are susceptible to adversarial attacks. These weaknesses can be exploited to extract sensitive data or compromise the model's integrity. On the other hand, the model initialization and data representation might have an effect on how to defend against strong adversarial attacks.
Threat Agents/Attack Vectors: Potential threat agents include hackers, competitors, or researchers looking for vulnerabilities to expose for academic or malicious reasons. Attack vectors involve exploiting weaknesses in algorithm design or implementation, such as differential privacy breaches or exploiting initial model states. Attackers might also manipulate input data in ways that cause the model to reveal more information than intended, particularly through adversarial machine learning techniques.
Impacts: The impacts include unauthorized access to sensitive data, manipulation of model behavior, and potential reputational damage if the vulnerabilities lead to significant breaches. For sectors relying heavily on data confidentiality, such as finance and healthcare, the consequences can extend to financial losses and regulatory penalties.
Example Attack Scenarios:

> In a financial services company, a machine learning model used for credit scoring inadvertently leaks information about applicants' financial history due to an inadequately secured algorithm. A competitor exploits this flaw to gather confidential data, gaining a competitive advantage.
> Researchers identify a vulnerability in the data representation of a deep learning model used in facial recognition. They develop a series of adversarial images that, when processed by the model, cause it to misclassify inputs, effectively enabling unauthorized access to a secure facility.

- **C25. Countermeasure name:** Carefully consider the model choice
  - **State:** IMPLEMENTED
- **C26. Countermeasure name:** Review the representation robustness
  - **State:** IMPLEMENTED

MT13. Threat name: An adversary may be able to manipulate an online learning system [T-ML-AI-MANIPULATE-ONLINE-SYSTEM]

- **State:** ✓ Mitigate
- **Description:**

General Threat Description: This threat involves adversaries targeting online learning systems, which are designed to continuously update and adjust their models based on new data received during operation. Such systems are particularly vulnerable to data poisoning, where attackers inject maliciously crafted data to manipulate the learning process. This can cause the system to deviate from its original purpose and lead to inaccurate or harmful outputs, known as model drift, where the system's decisions increasingly diverge from intended outcomes.
Threat Agents/Attack Vectors: Threat agents include competitors, malicious insiders, or external hackers who can influence the data input streams of online learning models. Attack vectors involve the injection of skewed or false data into the system’s learning inputs, which could be facilitated through compromised data sources, direct interaction with the model's API, or by exploiting vulnerabilities in data ingestion mechanisms.
Impacts: The primary impact is the degradation of the model's utility and reliability, which can lead to incorrect decision-making. For systems deployed in critical areas like finance, healthcare, or autonomous operations, the consequences could include significant financial losses, endangerment of lives, and erosion of trust in automated systems.
Example Attack Scenarios:

> A malicious actor targets a stock trading algorithm that learns from online data feeds, injecting false trend data that causes the model to make poor trading decisions, resulting in substantial financial loss.
> An insider within a healthcare organization manipulates data inputs to an online diagnostic tool, causing it to misdiagnose conditions based on biased data, leading to inappropriate treatment recommendations.

- **C27. Countermeasure name:** Consider robust learning and defenses against poisoning attacks for online models
  - **State:** IMPLEMENTED

MT14. Threat name: An adversary may be able to manipulate model parameters or hyperparameters [T-ML-AI-MANIPULATE-PARAMETERS]

- **State:** ✓ Mitigate
- **Description:**

General Threat Description: This threat involves the malicious alteration of the parameters or hyperparameters within ML/AI systems. Parameters, which the model learns to map inputs to desired outputs, and hyperparameters, which govern the learning process itself, are critical to the model's performance. Due to the complex and sometimes opaque nature of these settings—often considered an art more than a science—they present attractive targets for attackers aiming to degrade or control model behavior. Manipulation of these elements can lead to incorrect outputs, rendering the system unreliable or manipulated to serve adversarial goals.
On the other hand, note that certain models may also make parameters available to end users, such as temperature in LLMs. These can have direct impact on models performance and evaluation.
Threat Agents/Attack Vectors: Threat agents include hackers, competitive entities, and malicious insiders with access to the model's configuration. Attack vectors may involve direct access to the model's training environment, exploiting insecure APIs, or social engineering to gain administrative access that would allow the modification of model settings.
Impacts: The primary impacts are severe degradation in model accuracy, reliability, and validity. For critical applications, such as autonomous driving, financial forecasting, or healthcare diagnostics, the consequences could be catastrophic, including physical harm, financial ruin, and loss of life.
Example Attack Scenarios:
An insider at a financial institution subtly changes the hyperparameters of a high-frequency trading model, causing it to execute unprofitable trades that benefit a competitor.
A cyber attacker gains access to a cloud-based AI service for medical image processing and alters its parameters to misdiagnose conditions, affecting patient care and exposing the service provider to legal action.

- **C28. Countermeasure name:** Perform sensitivity analyses and secure/restrict the set of model parameters and hyperparameters
  - **State:** IMPLEMENTED

Component: Load Balancer

MT15. Threat name: Attackers use DDoS attacks to overwhelm the Load Balancer [T-LOAD-BALANCER-T-LOADBALANCER-01]

- **State:** ✓ Mitigate
- **Description:**

General threat description
Distributed Denial of Service (DDoS) attacks involve overwhelming a load balancer with a massive volume of requests from multiple sources. These attacks aim to exhaust system resources, disrupt traffic distribution, and render services unavailable to legitimate users.
Threat agents/Attack vectors
Botnets: Large networks of compromised devices used to generate excessive traffic targeting the load balancer.
Application Layer Attacks: Exploit specific application endpoints to overload backend systems via the load balancer.
Amplification Attacks: Use reflection and amplification techniques (e.g., DNS amplification) to increase traffic volume.
Malicious Actors: Attackers with intent to disrupt business operations or extort ransom.
Impacts
Service Downtime: Load balancer resources are exhausted, leading to denial of service for legitimate users.
Operational Disruption: Backend systems may become overwhelmed due to uneven traffic distribution.
Financial Loss: Downtime can result in revenue loss, SLA violations, and remediation costs.
Reputation Damage: Customers and partners may lose trust due to repeated service unavailability.
Example Attack Scenarios
Botnet Flooding: An attacker launches a volumetric DDoS attack using a botnet, sending millions of requests to the load balancer, overwhelming its capacity and causing downtime.
HTTP Flood Attack: A malicious actor generates excessive HTTP GET/POST requests targeting application endpoints, making the backend servers inaccessible via the load balancer.
DNS Amplification Attack: An attacker exploits misconfigured DNS servers to send amplified traffic toward the load balancer, crippling its ability to route legitimate traffic.
Slowloris Attack: The attacker sends partial HTTP requests through the load balancer, holding connections open and consuming server resources until legitimate connections are blocked.

- **C29. Countermeasure name:** Implement DDoS protection services to safeguard Load Balancers against attacks
  - **State:** IMPLEMENTED

MT16. Threat name: Attackers use resource exhaustion tactics to overwhelm the Load Balancer [T-LOAD-BALANCER-T-LOADBALANCER-04]

- **State:** ✓ Mitigate
- **Description:**

General threat description
Resource exhaustion occurs when a load balancer's capacity is overwhelmed, causing it to fail or degrade in performance. This can result from excessive traffic, inefficient configurations, or targeted attacks, impacting service availability and reliability.
Threat agents/Attack vectors
Malicious Actors: Deliberately generate traffic to overwhelm resources.
Botnets: Flood the load balancer with automated traffic.
Misconfigurations: Poorly optimized rules or policies leading to uneven resource utilization.
Impacts
Service Downtime: Disrupts access for legitimate users.
Performance Degradation: Slows response times and increases latency.
Operational Strain: Overloads backend systems, reducing efficiency.
Example Attack Scenarios
SYN Flood Attack: Incomplete connection requests exhaust connection resources.
Traffic Surge: A legitimate spike in usage overwhelms the load balancer.
Policy Misconfigurations: Overloaded nodes result from unoptimized traffic distribution.

- **C30. Countermeasure name:** Implement monitoring tools to track Load Balancer resource usage and prevent exhaustion
  - **State:** IMPLEMENTED

MT17. Threat name: Attackers can intercept traffic through Load Balancer vulnerabilities [T-LOAD-BALANCER-T-LOADBALANCER-02]

- **State:** ✓ Mitigate
- **Description:**

General threat description
Traffic interception occurs when attackers capture and potentially manipulate data transmitted between clients, the load balancer, and backend systems. This can lead to data breaches, unauthorized access, and compromised communications, often exploiting insecure configurations or weak encryption.
Threat agents/Attack vectors
Man-in-the-Middle (MITM) Attacks: Attackers position themselves between the client and the load balancer to intercept traffic.
Exploitation of Weak Encryption: Use of outdated or improperly configured encryption protocols allows attackers to decrypt sensitive data.
Compromised Networks: Attackers leverage unsecured or public networks to intercept unencrypted traffic.
Impacts
Data Breaches: Sensitive information such as credentials or financial data is exposed.
Session Hijacking: Attackers take over legitimate user sessions to impersonate users.
Data Manipulation: Intercepted traffic is altered to inject malicious content or disrupt operations.
Example Attack Scenarios
Unencrypted Traffic: An attacker captures sensitive information transmitted over an HTTP connection.
TLS Downgrade Attack: An attacker forces the connection to use a weaker encryption protocol to facilitate decryption.
Public Wi-Fi Exploit: An attacker intercepts traffic from clients connecting to the load balancer via an unsecured network.

- **C31. Countermeasure name:** Enforce TLS encryption for all traffic through the Load Balancer to prevent interception
  - **State:** IMPLEMENTED

MT18. Threat name: Attackers exploit security misconfigurations [T-LOAD-BALANCER-T-LOADBALANCER-03]

- **State:** Mitigate
- **Description:**

General threat description
Security misconfigurations in a load balancer expose vulnerabilities that attackers can exploit to gain unauthorized access, bypass security controls, or disrupt operations. Misconfigurations may include improper access controls, weak default settings, or overly permissive policies.
Threat agents/Attack vectors
Malicious Actors: Exploit weak configurations to bypass security controls.
Automated Tools: Scan for open ports, weak credentials, or misconfigured services.
Insider Threats: Take advantage of unintentional or deliberate misconfigurations.
Impacts
Unauthorized Access: Compromises sensitive data or systems.
Operational Disruption: Misused configurations lead to service downtime or instability.
Increased Vulnerabilities: Weak settings expose the system to further attacks.
Example Attack Scenarios
Default Credentials: An attacker uses factory default passwords to access the load balancer's admin console.
Open Management Ports: Unsecured ports allow unauthorized remote management access.
Overly Permissive Rules: An attacker bypasses filtering policies due to lenient traffic rules.

- **C32. Countermeasure name:** Conduct regular audits of Load Balancer configurations to ensure adherence to security best practices
  - **State:** IMPLEMENTED

MT19. Threat name: Attackers can hijack sessions by gaining unauthorized access to a user's active session [T-LOAD-BALANCER-T-LOADBALANCER-05]

- **State:** Mitigate
- **Description:**

General threat description
Session hijacking occurs when attackers take control of a legitimate user session by stealing or manipulating session identifiers. This allows attackers to impersonate users, gain unauthorized access, and perform malicious activities within the application.
Threat agents/Attack vectors
Man-in-the-Middle (MITM): Attackers intercept session tokens during transmission.
Session Token Theft: Exploitation of insecure storage or transmission of session cookies.
Cross-Site Scripting (XSS): Attackers inject malicious scripts to steal session identifiers.
Impacts
Unauthorized Access: Attackers gain access to sensitive resources and perform unauthorized actions.
Data Compromise: Sensitive information, such as user data, is exposed.
Service Disruption: Malicious actions by hijacked sessions disrupt normal operations.
Example Attack Scenarios
Insecure HTTP Connection: An attacker intercepts session cookies transmitted over an unencrypted connection.
XSS Exploit: A user clicks on a malicious link, allowing the attacker to steal session tokens stored in the browser.
Session Fixation: An attacker sets a predefined session ID, which the user unknowingly adopts during login, granting the attacker access.

- **C33. Countermeasure name:** Implement secure session handling to prevent session hijacking through the Load Balancer
  - **State:** IMPLEMENTED

Component: MongoDB NoSQL

UT4. Threat name: Insider threats can compromise data integrity [T-MONGODB-NOSQL-T-MONGO-06]

- **State:** Expose
- **Description:**

General threat description
Malicious or negligent insiders with access to MongoDB can alter, delete, or corrupt data, either intentionally or accidentally. Weak access controls and lack of monitoring make it easier for insiders to manipulate critical data.
Threat agents/Attack vectors
Threat Agents : Disgruntled employees, compromised accounts, negligent users
Attack Vectors : Deleting or modifying records without authorization
Abusing administrative privileges to alter database settings
Executing unauthorized scripts or bulk operations
Impacts
Data corruption : Loss of accuracy and reliability in stored information
Data loss : Critical records may be permanently deleted
Operational disruption : Business functions relying on MongoDB may be impacted
Example attack scenarios
Malicious Deletion : A disgruntled employee with admin access deletes customer records.
Unauthorized Modifications : An insider alters financial data to commit fraud.
Accidental Corruption : A careless user runs a faulty script that overwrites important data.

- **C34. Countermeasure name:** Restrict access and conduct regular audits
  - **State:** RECOMMENDED

MT20. Threat name: Attackers can overload the database with DoS attacks [T-MONGODB-NOSQL-T-MONGO-05]

- **State:** Mitigate
- **Description:** **General threat description**
  Attackers can overwhelm MongoDB with excessive requests, complex queries, or connection exhaustion, causing performance degradation or complete unavailability.
  Threat agents/Attack vectors
  Threat Agents : Hackers, botnets, disgruntled employees
  Attack Vectors : Flooding the database with read/write requests
  Running resource-intensive queries (e.g., complex aggregations)
  Exhausting connection pools by opening numerous sessions
  Impacts
  Availability loss : The database slows down or crashes, disrupting services
  Performance degradation : Increased query response times affect applications
  Operational costs : Higher infrastructure costs or downtime-related losses
  Example attack scenarios
  Query Flooding : Attackers send millions of read queries, consuming CPU/memory.
  Connection Pool Exhaustion : Excessive connections block legitimate users.
  Aggregation Overload : Complex queries lock up system resources.
  Storage Exhaustion : Uncontrolled data insertion fills disk space.

- **C35. Countermeasure name:** Use rate limiting and connection pooling
  - **State:** IMPLEMENTED

MT21. Threat name: Attackers can exploit default configurations [T-MONGODB-NOSQL-T-MONGO-07]

- **State:** ✓ Mitigate
- **Description:**

General threat description
MongoDB instances with default settings are vulnerable to unauthorized access, data breaches, and system compromise. Attackers exploit weak authentication, open network exposure, and overly permissive access controls to gain control over the database.
Threat agents/Attack vectors
Threat Agents : Hackers, automated bots, malicious insiders
Attack Vectors : Accessing publicly exposed MongoDB instances without authentication
Exploiting default or weak credentials
Abusing default role-based access control (RBAC) settings
Impacts
Unauthorized access : Attackers gain full control over the database
Data breaches : Sensitive information is exposed or stolen
Data loss/manipulation : Attackers can delete, modify, or encrypt data
Example attack scenarios
Public Exposure : An unsecured MongoDB instance is discovered online, allowing attackers to access and extract data.
Default Credentials : An attacker logs in using common default credentials (e.g., admin:admin).
Overly Permissive Access : A low-privileged user escalates access due to misconfigured RBAC.

- **C36. Countermeasure name:** Harden configuration and disable unused features
  - **State:** IMPLEMENTED

MT22. Threat name: Attackers can gain unauthorized access [T-MONGODB-NOSQL-T-MONGO-01]

- **State:** ✓ Mitigate
- **Description:**

General threat description
Unauthorized access to MongoDB can occur due to weak authentication, misconfigurations, or credential leaks. Attackers exploit these weaknesses to steal, modify, or delete data, potentially compromising entire applications.
Threat agents/Attack vectors
Threat Agents : Hackers, malicious insiders, automated bots
Attack Vectors : Exploiting weak or missing authentication
Using stolen or default credentials
Brute-force or dictionary attacks on login credentials
Impacts
Data breaches : Exposure of sensitive information
Data loss/manipulation : Attackers can delete or alter records
Privilege escalation : Gaining administrative access to the database
Example attack scenarios
No Authentication : An open MongoDB instance allows attackers to access and exfiltrate data.
Credential Theft : Stolen database credentials from a data breach grant unauthorized entry.
Brute-Force Attack : Attackers repeatedly try common passwords to gain access.

- **C37. Countermeasure name:** Implement strong authentication and RBAC
  - **State:** IMPLEMENTED

MT23. Threat name: Malicious users can escalate privileges [T-MONGODB-NOSQL-T-MONGO-04]

- **State:** ✓ Mitigate
- **Description:**

General threat description
Malicious users with limited access may exploit misconfigurations, vulnerabilities, or weak access controls to gain higher privileges in MongoDB. This can lead to unauthorized data access, modification, or complete system takeover.
Threat agents/Attack vectors
Threat Agents : Malicious insiders, compromised accounts, advanced attackers
Attack Vectors : Exploiting misconfigured role-based access control (RBAC)
Abusing vulnerabilities to elevate privileges
Gaining access to admin credentials through phishing or credential leaks
Impacts
Unauthorized access : Attackers gain admin-level control
Data manipulation : Malicious users alter or delete critical data
Persistence : Attackers create new privileged accounts for long-term access
Example attack scenarios
Weak RBAC Configuration : A user with read-only access exploits a misconfiguration to gain write or admin privileges.
Credential Theft : An attacker steals admin credentials and escalates access.
Privilege Exploitation : A user abuses a system flaw to execute unauthorized commands.

- **C38. Countermeasure name:** Enforce strict role management and audit logs
  - **State:** IMPLEMENTED

MT24. Threat name: Attackers can access unsecured backups [T-MONGODB-NOSQL-T-MONGO-08]

- **State:** ✓ Mitigate
- **Description:**

General threat description
Unsecured MongoDB backups can be accessed by attackers if they are stored in exposed locations, lack encryption, or have weak access controls. This can lead to data breaches, ransomware attacks, or unauthorized data restoration.
Threat agents/Attack vectors
Threat Agents : Hackers, malicious insiders, automated bots
Attack Vectors : Accessing publicly exposed backup files in cloud storage or local servers
Exploiting weak or missing encryption on backup data
Using stolen credentials to retrieve and manipulate backups
Impacts
Data breaches : Sensitive data is leaked or stolen
Data manipulation : Attackers alter backups, compromising data integrity
Ransomware attacks : Threat actors encrypt or delete backups for extortion
Example attack scenarios
Exposed Cloud Storage : A MongoDB backup is left publicly accessible in an S3 bucket, allowing attackers to download and exploit it.
Unencrypted Backup Theft : Attackers gain access to an unprotected backup and extract sensitive data.
Credential Compromise : A leaked admin password allows an attacker to restore and manipulate backups.

- **C39. Countermeasure name:** Encrypt backups and limit backup access
  - **State:** IMPLEMENTED

MT25. Threat name: Attackers can intercept unencrypted data [T-MONGODB-NOSQL-T-MONGO-03]

- **State:** ✓ Mitigate
- **Description:**

General threat description
MongoDB data transmitted without encryption is vulnerable to interception by attackers using network sniffing techniques. This can lead to unauthorized data access, manipulation, or credential theft, compromising the confidentiality and integrity of the database.
Threat agents/Attack vectors
Threat Agents : Hackers, malicious insiders, network attackers
Attack Vectors : Capturing unencrypted database traffic over unsecured networks
Performing Man-in-the-Middle (MitM) attacks to modify data in transit
Exploiting misconfigured or disabled TLS/SSL settings
Impacts
Data exposure : Sensitive information, including credentials, can be stolen
Data integrity risks : Attackers can alter transmitted data
Regulatory non-compliance : Violations of data protection regulations (e.g., GDPR, HIPAA)
Example attack scenarios
Unencrypted Credentials : An attacker eavesdrops on MongoDB traffic and captures login credentials sent in plaintext.
Man-in-the-Middle Attack : A hacker intercepts and modifies sensitive data during transmission.
Wi-Fi Sniffing : A malicious actor on a shared network captures unencrypted database queries and responses.

- **C40. Countermeasure name:** Enable encryption for data in transit and at rest
  - **State:** IMPLEMENTED

MT26. Threat name: Attackers take advantage of injection vulnerabilities [T-MONGODB-NOSQL-T-MONGO-02]

- **State:** ✓ Mitigate
- **Description:**

General threat description
Injection vulnerabilities in MongoDB occur when unvalidated user input is executed as database queries, allowing attackers to manipulate data, extract sensitive information, or execute unauthorized commands.
Threat agents/Attack vectors
Threat Agents : Hackers, malicious insiders, automated bots
Attack Vectors : Injecting malicious JavaScript or MongoDB query operators ($where, $ne, $gt)
Exploiting improperly sanitized user input in database queries
Bypassing authentication or authorization mechanisms via injection
Impacts
Data breaches : Attackers extract or modify sensitive data
Authentication bypass : Unauthorized users gain system access
Service disruption : Malicious queries degrade database performance
Example attack scenarios
Query Manipulation : An attacker injects { "username": { "$ne": null } }, bypassing login checks.
JavaScript Injection : A vulnerable $where query executes arbitrary JavaScript code.
Mass Data Exposure : An attacker exploits an injection flaw to dump an entire collection.

- **C41. Countermeasure name:** Sanitize inputs and use parameterized queries
  - **State:** IMPLEMENTED

Component: OAuth2 Authorization Server

UT5. Threat name: Server application does not offer any mechanism to let the user exercise their Right to Correct [T-CCPA-SERVER-SIDE-RIGHT-TO-CORRECT]

- **State:** 🔈 Expose
- **Description:**

If a server application does not offer any mechanism to let the user exercise their Right to Correct the user is unable to correct any inaccurate personal information that the business has collected about them. This can lead to incorrect decisions being made about users, such as denying them access to services or products, or even exposing them to identity theft or fraud.

- **C42. Countermeasure name:** Develop a dashboard or reporting tool to track and monitor requests
  - **State:** RECOMMENDED

UT6. Threat name: Server application does not offer any mechanism to let the user exercise their Right to Delete [T-CCPA-SERVER-SIDE-RIGHT-TO-DELETE]

- **State:** 🔈 Expose
- **Description:**

If a server application does not offer any mechanism to let the user exercise their Right to Delete the user is unable to delete any personal information that the business has collected about them. This can be a threat to the user's privacy and security because personal information can be used for malicious purposes, such as identity theft or fraud.

- **C43. Countermeasure name:** Develop a dashboard or reporting tool to track and monitor requests
  - **State:** RECOMMENDED

UT7. Threat name: Server application does not offer any mechanism to let the user exercise their Right to Know [T-CCPA-SERVER-SIDE-RIGHT-TO-KNOW]

- **State:** 🔈 Expose
- **Description:**

If a client application does not offer any mechanism to let the user exercise their Right to Know the user is unable to know what personal information the business has collected about them and how it has been used. This can be a threat to the user's privacy and security because they may not be aware of how their personal information is being used or shared, which can lead to unwanted solicitations or even identity theft.

- **C44. Countermeasure name:** Develop a dashboard or reporting tool to track and monitor requests
  - **State:** RECOMMENDED

MT27. Threat name: Attackers exploit weak access policies [T-OAUTH2-AUTHORIZATION-SERVER-T-OAUTH2-AS-06]

- **State:** ✓ Mitigate
- **Description: General threat description**
Weak access policies in the OAuth2 Authorization Server allow attackers to gain unauthorized access or escalate privileges by exploiting overly permissive or misconfigured scopes.
Threat agents/Attack vectors
Agents: Malicious users, compromised clients, unauthorized apps.
Vectors: Abusing broad scopes, manipulating OAuth2 flows, or exploiting default policies.

Impacts
Unauthorized data access or privilege escalation.
Breaches, reputation damage, regulatory violations.
Example Attack Scenarios
A third-party app is granted admin-level access due to misconfigured policies.
An attacker manipulates scopes to access restricted data.
Default policies grant new apps excessive permissions.

- **C45. Countermeasure name:** Enforce principle of least privilege in scope management
  - **State:** IMPLEMENTED

MT28. Threat name: Attackers exploit open redirects [T-OAUTH2-AUTHORIZATION-SERVER-T-OAUTH2-AS-03]

- **State:** ✓ Mitigate
- **Description: General threat description**
  Open redirect vulnerabilities in the OAuth2 Authorization Server allow attackers to redirect users to malicious sites during the authorization flow, potentially stealing credentials or tokens.
  Threat agents/Attack vectors
  Agents: Malicious actors, phishing campaigns.
  Vectors: Manipulating redirect URIs to point to malicious domains.
  Impacts
  Credential theft or token interception.
  Phishing attacks and user trust erosion.
  Example Attack Scenarios
  An attacker tricks the server into redirecting users to a phishing site during login.
  A malicious app intercepts authorization tokens by exploiting an open redirect.
- **C46. Countermeasure name:** Validate redirect URIs against a strict whitelist
  - **State:** IMPLEMENTED

MT29. Threat name: Attackers intercept tokens over unencrypted communication channels [T-OAUTH2-AUTHORIZATION-SERVER-T-OAUTH2-AS-08]

- **State:** ✓ Mitigate
- **Description: General threat description**
  When tokens are transmitted over unencrypted channels (e.g., HTTP), attackers can intercept them using techniques like packet sniffing, enabling unauthorized access to protected resources.
  Threat agents/Attack vectors
  Agents: Network eavesdroppers, malicious intermediaries.
  Vectors: Intercepting tokens sent over insecure HTTP or during communication without encryption.
  Impacts
  Unauthorized resource access.
  Potential data breaches and service misuse.
  Example Attack Scenarios
  Tokens are intercepted via a man-in-the-middle (MITM) attack on an HTTP connection.
  An attacker captures tokens transmitted over an unsecured public Wi-Fi network.
- **C47. Countermeasure name:** Enforce HTTPS and modern TLS for secure communication
  - **State:** IMPLEMENTED

MT30. Threat name: Attackers bypass detection due to lack of logging and monitoring [T-OAUTH2-AUTHORIZATION-SERVER-T-OAUTH2-AS-07]

- **State:** ✓ Mitigate
- **Description:**

General threat description
Without proper logging and monitoring in the OAuth2 Authorization Server, attackers can conduct malicious activities undetected, including token theft, privilege escalation, and unauthorized access.
Threat agents/Attack vectors
Agents: Malicious users, insiders, automated attack bots.
Vectors: Exploiting the absence of logging for unauthorized operations or manipulating tokens without traceable activity.
Impacts
Delayed detection of security breaches.
Prolonged exploitation and greater damage.
Example Attack Scenarios
An attacker repeatedly tries token brute-forcing without triggering alerts.
Unauthorized scope escalation goes unnoticed due to lack of audit logs.

- **C48. Countermeasure name:** Log key actions and monitor for unusual activity
  - **State:** IMPLEMENTED

MT31. Threat name: Attackers inject malicious authorization codes [T-OAUTH2-AUTHORIZATION-SERVER-T-OAUTH2-AS-04]

- **State:** ✓ Mitigate
- **Description: General threat description**
  Attackers inject malicious authorization codes into the OAuth2 flow to hijack sessions, steal access tokens, or impersonate legitimate users or clients.
  Threat agents/Attack vectors
  Agents: Malicious actors, compromised clients, phishing attacks.
  Vectors: Exploiting weaknesses in code validation or tricking users into authorizing fake codes.
  Impacts
  Unauthorized access to user accounts or resources.
  Token theft, data breaches, and impersonation.
  Example Attack Scenarios
  An attacker uses a phishing site to inject a malicious code into the redirect URI.
  A compromised client sends manipulated authorization codes to the server to steal tokens.
- **C49. Countermeasure name:** Bind authorization codes to specific clients and enforce PKCE for public clients
  - **State:** IMPLEMENTED

MT32. Threat name: Attackers perform brute force on client credentials [T-OAUTH2-AUTHORIZATION-SERVER-T-OAUTH2-AS-02]

- **State:** ✓ Mitigate
- **Description:**

General threat description
Attackers may attempt to brute force client credentials (e.g., client IDs and secrets) to gain unauthorized access to the OAuth2 Authorization Server, enabling them to impersonate legitimate clients.
Threat agents/Attack vectors
Agents: Attackers, automated bots.
Vectors: Repeatedly guessing client credentials using brute force methods to obtain valid access.
Impacts
Unauthorized access to protected resources.
Compromised client applications, leading to data breaches.
Example Attack Scenarios

An attacker uses a dictionary attack to guess client credentials and gain access to the server.
A brute-force attack is launched against the OAuth2 client's credentials to issue tokens fraudulently.

- **C50. Countermeasure name:** Mandate high-entropy client secrets and protect against brute-force attacks
  - **State:** IMPLEMENTED

MT33. Threat name: Attackers perform token replay attacks [T-OAUTH2-AUTHORIZATION-SERVER-T-OAUTH2-AS-05]

- **State:** Mitigate
- **Description:** **General threat description**
  Attackers intercept and reuse valid access tokens to gain unauthorized access to protected resources, bypassing authentication mechanisms by replaying tokens without authorization.
  Threat agents/Attack vectors
  Agents: Malicious actors, network eavesdroppers.
  Vectors: Intercepting and reusing valid tokens in different sessions or on different systems.
  Impacts
  Unauthorized access to resources.
  Data leakage, service disruption, and potential privilege escalation.
  Example Attack Scenarios
  An attacker intercepts an access token in transit and uses it to access resources on behalf of a legitimate user.
  A valid token is stolen and reused in a different session to bypass authentication and access sensitive data.
- **C51. Countermeasure name:** Use short-lived access tokens with refresh token rotation and validate token binding
  - **State:** IMPLEMENTED

MT34. Threat name: Attackers exploit insufficient token validation [T-OAUTH2-AUTHORIZATION-SERVER-T-OAUTH2-AS-01]

- **State:** Mitigate
- **Description:** **General threat description**
  Weak or improper token validation allows attackers to bypass security measures and gain unauthorized access to resources by using tampered or expired tokens.
  Threat agents/Attack vectors
  Agents: Malicious actors, attackers with knowledge of token structures.
  Vectors: Exploiting weak validation mechanisms, such as missing signature verification or failing to check token expiration.
  Impacts
  Unauthorized access to sensitive data or services.
  Compromised application security and potential data breaches.
  Example Attack Scenarios
  An attacker modifies the payload of a JWT token and bypasses validation checks to gain unauthorized access.
  A token is accepted past its expiration date, allowing an attacker to reuse it for unauthorized actions.
- **C52. Countermeasure name:** Validate tokens by verifying signature, claims, and expiration
  - **State:** IMPLEMENTED

Component: Trained model

MT35. Threat name: An adversary may be able to reveal sensitive information available in the model or the data used to build it [T-ML-AI-PRIVACY]

- **State:** Mitigate
- **Description:**

General Threat Description: Machine learning (ML) and AI models can inadvertently encode and retain sensitive information from the data they are trained on. This poses a risk where an adversary could extract or infer sensitive data from the model itself, especially if the model architecture or the algorithm inherently retains data characteristics (e.g., k-nearest neighbors). Understanding the potential for data leakage through model outputs or behavior is crucial, as is selecting algorithms that minimize the risk of exposing sensitive data.
Threat Agents/Attack Vectors: Potential threat agents include external attackers seeking to exploit model vulnerabilities or insiders who leverage their access to extract data. Attack vectors might involve techniques like model inversion attacks, where outputs are used to reconstruct training data, or exploiting models that store exemplars of sensitive data.
Impacts: Revealing sensitive information through a model can lead to privacy breaches, violating user trust and legal compliance (such as GDPR). It can also have financial implications for the organization in terms of fines and remediation costs, and damage the organization's reputation significantly.
Example Attack Scenarios
1. An adversary uses a series of queries to a public AI service to perform a model inversion attack, reconstructing personal data of individuals whose information was used in the training set.
2. A malicious insider accesses a model using k-nearest neighbors that retains exemplars of sensitive medical records, and extracts these records to sell on the black market.

- **C53. Countermeasure name:** Carefully consider the model choice
  - **State:** IMPLEMENTED
- **C54. Countermeasure name:** Keep a history of queries to the model
  - **State:** IMPLEMENTED
- **C55. Countermeasure name:** Consider privacy-preserving techniques and strategies
  - **State:** IMPLEMENTED

MT36. Threat name: Lack of sufficient details about the model (algorithm), its architecture, or its data [T-ML-AI-INFORMATION-DEFICIT]

- **State:** Mitigate
- **Description:**

General Threat Description: Inadequate documentation and transparency about the construction and training data of a machine learning model can pose significant risks, particularly concerning compliance with laws and regulations. This lack of detail can hinder the ability to verify the integrity and fairness of the model, understand its decision-making process, and ensure that it does not perpetuate or introduce bias. It may also obstruct efforts to replicate or further develop the model responsibly. In fact, properly evaluating and/or fine-tuning opaque third-party models can be seen as a 'black art'! Also, for large models, such as LLMs, this issue may be amplified by the cost factor which may lead to cutting corners or intellectual shortcuts while building the models.
Threat Agents/Attack Vectors: This threat is often not due to external attackers but rather internal oversights or negligence. It may also stem from proprietary practices where organizations intentionally obscure details about AI models to protect intellectual property. However, this poses risks when users and regulators require transparency for legal and operational purposes.
Impacts: The primary impact is regulatory non-compliance, which can lead to fines, restrictions, or compulsory modifications to the deployment of the AI system. It can also result in reduced trust from users and the public, especially in sensitive applications such as those involving personal data or affecting individual rights and freedoms.
Example Attack Scenarios
1. A financial institution uses an AI model for credit scoring but does not disclose sufficient information about the data sources or the model's decision-making process. When the model unfairly denies credit to a minority group, the institution is unable to prove compliance with fair lending laws and faces legal action.
2. A healthcare provider employs a proprietary AI system for diagnosing patients but lacks detailed documentation on the training data, which unknowingly contains biased data against certain demographics. The lack of transparency prevents effective audit trails, leading to continued misdiagnoses and eventual regulatory scrutiny.

- **C56. Countermeasure name:** Keep sufficient details and documentation about the content used for training/fine-tuning models
  - **State:** IMPLEMENTED
- **C57. Countermeasure name:** Maintain sufficient technical documentation about the model building and usage

- **State:** IMPLEMENTED

MT37. Threat name: An adversary may take advantage of model sharing and/or transfer [T-ML-AI-SHARING-TRANSFER]

- **State:** ✓ Mitigate
- **Description:**

General Threat Description: The storage and transfer of machine learning models expose them to risks of tampering and compromise, particularly when models are shared or reused among different users or applications. This risk is heightened with the use of ready-to-use models that may not have undergone rigorous security checks. Models could be embedded with malicious components, such as Trojans, that activate under specific conditions to alter the behavior of the model or to facilitate data breaches.
Threat Agents/Attack Vectors: Threat agents include competitors, cybercriminals, or malicious insiders who can embed Trojans or backdoors in models before they are shared or transferred. These models can be distributed via unsecured networks, third-party repositories, or included in shared codebases. Attack vectors also include manipulating model files during transfer, such as through man-in-the-middle attacks, or substituting legitimate models with compromised versions on compromised platforms.
Impacts: The introduction of compromised AI models into systems can lead to unauthorized access, data breaches, incorrect model predictions, and manipulation of system behavior. These impacts can have severe consequences, including operational disruption, loss of sensitive information, compliance violations, and damage to an organization's credibility.
Example Attack Scenarios
1. A data scientist downloads a model from a public repository for image processing tasks. Unbeknownst to them, the model contains a Trojan that activates in specific conditions, leaking classified images to an external server.
2. During a model transfer between departments within a company, an attacker intercepts the model file and injects a backdoor. When deployed, the model allows the attacker to bypass normal authentication checks and access restricted parts of the network.

- **C58. Countermeasure name:** Consider possible trojanized versions when acquiring shared models
  - **State:** IMPLEMENTED
- **C59. Countermeasure name:** Protect data and IP (Intellectual Property) when sharing or shipping models
  - **State:** IMPLEMENTED

End of Technical Threat report