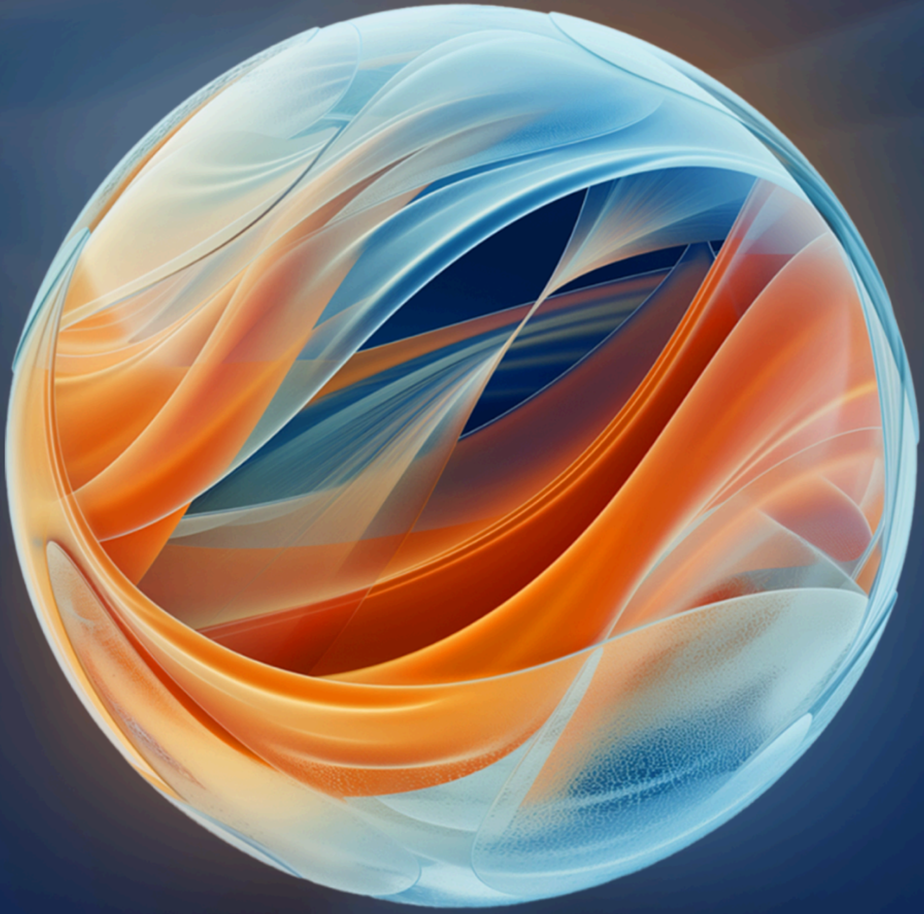
AI Organizational Responsibilities:

Core Security Responsibilities



AI Organizational Responsibilities
Working Group

cloud
CSA security
alliance®

The permanent and official location for the AI Organizational Responsibilities Working Group is https://cloudsecurityalliance.org/research/working-groups/ai-organizational-responsibilities

Acknowledgments

Lead Authors

Jerry Huang
Ken Huang

Contributors/Co-Chairs

Ken Huang
Nick Hamilton
Chris Kirschke
Sean Wright

Reviewers

Candy Alexander
Ilango Allikuzhi
Eray Altili
Aakash Alurkar
Romeo Ayalin
Renu Bedi
Saurav Bhattacharya
Sergei Chaschin
Hong Chen
John Chiu
Satchit Dokras
Rajiv Gunja
Hongtao Hao, PhD
Grace Huang
Onyeka Illoh
Krystal Jackson
Arvin Jakkamreddy Reddy
Simon Johnson
Gian Kapoor
Ben Kereopa-Yorke
Chris Kirschke
Madura Malwatte
Madhavi Najana
Rajith Narasimhaiah
Gabriel Nwajiaku
Govindaraj Palanisamy
Meghana Parwate
Paresh Patel
Rangel Rodrigues
Michael Roza
Lars Ruddigkeit
Davide Scatto
Maria Schwenger Mj
Bhuvaneswari Selvadurai
Himanshu Sharma
Akshay Shetty
Nishanth Singarapu
Abhinav Singh
Dr. Chantal Spleiss
Patricia Thaine
Eric Tierling
Ashish Vashishtha
Peter Ventura
Jiewen Wang
Wickey Wang
Udith Wickramasuriya
Sounil Yu

CSA Global Staff

Marina Bregkou
Sean Heide
Alex Kaluza
Claire Lehnert
Stephen Lumpe

Table of Contents

Executive Summary

This white paper is a working draft that focuses on the information security and cybersecurity aspects of organizational responsibilities in the development and deployment of Artificial Intelligence (AI) and Machine Learning (ML) systems. The paper synthesizes expert-recommended best practices within core security areas, including data protection mechanisms, model vulnerability management, Machine Learning Operations (MLOps) pipeline hardening, and governance policies for training and deploying AI responsibly.

Key points discussed in the white paper include:

- **Data Security and Privacy Protection**: The importance of data authenticity, anonymization, pseudonymization, data minimization, access control, and secure storage and transmission in AI training.
- **Model Security**: Covers various aspects of model security, including access controls, secure runtime environments, vulnerability and patch management, MLOps pipeline security, AI model governance, and secure model deployment.
- **Vulnerability Management**: Discusses the significance of AI/ML asset inventory, continuous vulnerability scanning, risk-based prioritization, remediation tracking, exception handling, and reporting metrics in managing vulnerabilities effectively.

The white paper analyzes each responsibility using quantifiable evaluation criteria, the Responsible, Accountable, Consulted, Informed (RACI) model for role definitions, high-level implementation strategies, continuous monitoring and reporting mechanisms, access control mapping, and adherence to foundational guardrails. These are based on industry best practices and standards such as NIST AI RMF, NIST SSDF, NIST 800-53, CSA CCM, and others.

By outlining recommendations across these key areas of security and compliance, this paper aims to guide enterprises in fulfilling their obligations for responsible and secure AI design, development, and deployment

Introduction

This white paper focuses on what we define as an enterprise's "core security responsibilities" around Artificial Intelligence (AI) and Machine Learning (ML), data security, model security, and vulnerability management. As organizations have duties to uphold secure and safe AI practices, this white paper and two others in this series provide a blueprint for enterprises to fulfill such organizational responsibilities. Specifically, this white paper synthesizes expert-recommended best practices within core security areas - data protection mechanisms, model vulnerability management, MLOps pipeline hardening, and governance policies for training and deploying AI responsibly. The other two white papers in this series discuss additional aspects of secure AI development and deployment for enterprises. By outlining recommendations across these key areas of security and compliance in three targeted white papers, this series aims to guide enterprises in fulfilling their obligations for responsible and secure AI design, development, and deployment.

AI Shared Responsibility Model

The AI Shared Responsibility Model outlines the division of tasks between AI platform providers, AI application owners, AI developers and AI usage, varying by service models (SaaS, PaaS, IaaS).

The secure operation of AI applications involves a collaborative effort among multiple stakeholders. In the context of AI, responsibilities are shared between three key parties: the AI service users, the AI application owners and developers, and AI platform providers.

When evaluating AI-enabled integration, it is crucial to comprehend the shared responsibility model and delineate the specific tasks handled by each party.

Key Layers in an AI-Enabled Application

1. **AI Platform:**
    - This layer provides the AI capabilities to applications. It involves building and safeguarding the infrastructure that hosts AI models, training data, and configuration settings.
    - Security considerations include protecting against malicious inputs and outputs generated by the AI model. AI safety systems should protect against potential harmful inputs and outputs like hate, jailbreaks, and so on.
    - AI Platform Layer has following tasks:
        - Model safety and security
        - Model tuning
        - Model accountability
        - Model design and implementation
        - Model training and governance
        - AI compute and data infrastructure

2. **AI Application Layer:**
    - The AI application layer interfaces with users, leveraging the AI capabilities. Its complexity can vary significantly. At their most basic level, standalone AI applications serve as a conduit to a collection of APIs, which process textual prompts from users and relay them to the underlying model for a response. More sophisticated AI applications are capable of enriching these prompts with additional context, utilizing elements such as a persistence layer, a semantic index, or plugins that provide access to a broader range of data sources. The most advanced AI applications are designed to integrate seamlessly with pre-existing applications and systems, enabling a multi-modal approach that encompasses text, audio, and visual inputs to produce diverse content outputs.
    - As an AI application owner, you ensure seamless user experiences and handle any additional features or services. To safeguard an AI application from harmful activities, it is essential to establish a robust application safety system. A Generative AI (GenAI) system should thoroughly examine the content utilized in the prompt dispatched to the AI model. Additionally, it must scrutinize the exchanges with any add-ons like plugins and functions, data connectors, and interactions with other AI applications, a process referred to as *AI orchestration*. For those developing AI applications on an Infrastructure-as-a-Service (IaaS) or Platform-as-a-Service (PaaS) services, integrating a dedicated AI content safety feature is advisable. Depending on specific requirements, additional features may be implemented to enhance protection.
    - AI Application has the following tasks:
        - AI plugins and data connections
        - Application design and implementation
        - Application infrastructure
        - AI safety system
3. **AI Usage:**
    - The AI usage layer outlines the application and consumption of AI functionalities. GenAI introduces an innovative user/computer interaction model, distinct from traditional interfaces like APIs, command prompts, and GUIs. This new interface is interactive and adaptable, molding the computer's capabilities to the user's intentions. Unlike earlier interfaces that required users to conform to the system's design and functions, the generative AI interface prioritizes user interaction. This allows the users' inputs to significantly shape the system's output, emphasizing the importance of safety mechanisms to safeguard individuals, data, and corporate resources.
    - Security considerations for AI usage are akin to those for any computer system, relying on robust measures for identity and access management, device security, monitoring, data governance, and administrative controls.
    - Given the significant impact user actions can have on system outputs, a greater focus on user conduct and responsibility is necessary. It is essential to revise policies for acceptable use and to inform users about the distinctions between conventional IT applications and those enhanced by AI. This education should cover AI-specific issues concerning security, privacy, and ethical standards. Moreover, it's important to raise awareness among users about the potential for AI-driven attacks, which may involve sophisticatedly fabricated text, audio, video, and other media designed to deceive.
    - AI usage layer has the following tasks:
        - User training and accountability
        - Acceptable usage policy and admin controls
        - Identity and Access Management (IAM) and device controls

- Data governance

Remember that this shared responsibility model helps demarcate roles and ensures a clear separation of duties, contributing to the safe and effective use of AI technologies. The distribution of workload responsibilities varies based on the type of AI integration based on service models.

1. **Software as a Service (SaaS):**
   - In SaaS-based AI integrations, the AI platform provider assumes responsibility for managing the underlying infrastructure, security controls, and compliance measures.
   - As a user, your primary focus lies in configuring and customizing the AI application to align with your specific requirements.
2. **Platform as a Service (PaaS):**
   - PaaS-based AI platforms offer a middle ground. While the provider manages the core AI capabilities, you retain some control over configurations and customization.
   - You are responsible for ensuring the safe use of the AI model, handling training data, and adjusting model behavior (e.g., weights and biases).
3. **Infrastructure as a Service (IaaS):**
   - In IaaS scenarios, you have greater control over the infrastructure. However, this also means taking on more responsibilities.
   - You manage the entire stack, including the AI model, training data, and infrastructure security.

Foundational Components of a Data-Centric AI System

The foundational components of a data-centric AI system encompass the entire lifecycle of data and model management. These components work together to create a secure and effective AI system that can process data and provide valuable insights or automated decisions.

- **Raw Data**: The initial unprocessed data collected from various sources.
- **Data preparation**: The process of cleaning and organizing raw data into a structured format.
- **Data sets**: Curated collections of data, ready for analysis and model training.
- **Data and AI governance**: Policies and procedures to ensure data quality and ethical AI usage.
- **Machine Learning algorithms**: The computational methods used to interpret data.
- **Evaluation**: Assessing the performance of machine learning models.
- **Machine Learning Models**: The output of algorithms trained on datasets.
- **Model management**: Overseeing the lifecycle of machine learning models.
- **Model deployment and inference**: Implementing models to make predictions or decisions.
- **Inference outcomes**: The results produced by deployed models.
- **Machine Learning Operations (MLOps)**: Practices for deploying and maintaining AI models.
- **Data and AI Platform security**: Measures to protect the system against threats.

Data Operations: Involves the acquisition and transformation of data, coupled with the assurance of data security and governance. The efficacy of ML models is contingent upon the integrity of data pipelines and a fortified DataOps framework.

Model Operations: Encompasses the creation of predictive ML models, procurement from model marketplaces, or the utilization of Large Language Models (LLMs) such as those provided by OpenAI or through Foundation Model APIs. Model development is an iterative process that necessitates a systematic approach to document and evaluate various experimental conditions and outcomes.

Model Deployment and Serving: Entails the secure construction of model containers, the isolated and protected deployment of models, and the implementation of automated scaling, rate limiting, and surveillance of active models. It also includes the provision of features and functions for high-availability, low-latency services in Retrieval Augmented Generation (RAG) applications, as well as the requisite features for other applications, including those that deploy models externally to the platform or require data features from the catalog.

Operations and Platform: Covers the management of platform vulnerabilities, updates, model segregation, and system controls, along with the enforcement of authorized model access within a secure architectural framework. Additionally, it involves the deployment of operational tools for Continuous Integration/Continuous Deployment (CI/CD), ensuring that the entire lifecycle adheres to established standards across separate execution environments—development, staging, and production—for secure ML operations (MLOps).

Table 1 aligns the operations with the core aspects of a data-centric AI system, highlighting their roles and interdependencies

Foundational Component	Description
Data Operations	Ingestion, transformation, security, and governance of data.
Model Operations	Building, acquiring, and experimenting with ML models.
Model Deployment and Serving	Secure deployment, serving, and monitoring of ML models.
Operations and Platform	Platform security, model isolation, and CI/CD for MLOps.

Table 1: Mapping Data-Centric AI System Components and Their Interconnected Roles

Table 2 provides a synthesized view of the potential security risks and threats at each stage of an AI/ML system, along with examples and recommended mitigations to address these concerns.

System Stage	System Components	Potential Security Risks	Threats	Mitigations
Data Operations	Raw Data, Data Prep, Data sets	Data loss: Unauthorized deletion or corruption of data. Data poisoning: Deliberate manipulation of data to compromise the model's integrity. Compliance challenges: Failure to meet regulatory requirements for data protection.	Compromise/poisoning of data: Attackers may inject false data or alter existing data.	Implement robust data governance frameworks. Deploy anomaly detection systems. Establish recovery protocols and regular data backups.
Model Operations	ML Algorithms, Model Management	Model theft: Stealing of proprietary models. Unauthorized access: Gaining access to models without permission.	Attacks via API access: Exploiting API vulnerabilities to access or manipulate models. Model stealing (extraction): Replicating a model for unauthorized use.	Strengthen access controls and authentication mechanisms. Secure API endpoints through encryption and rate limiting. Regularly update and patch systems.
Model Deployment and Serving	Model Serving, Inference Response	Unauthorized access: Accessing the model serving infrastructure without authorization. Data leakage: Exposing sensitive information through misconfigured systems.	Model tricking (evasion): Altering inputs to receive a specific output from the model. Training data recovery (inversion): Extracting private training data from the model.	Secure deployment practices, including containerization and network segmentation. Active monitoring and logging of model interactions. Implement rate limiting and anomaly detection.
Operations and Platform	ML Operations, Data and AI Platform Security	Inadequate vulnerability management: Not addressing known vulnerabilities in a timely manner. Model isolation issues: Failure to properly isolate models, leading to potential cross-contamination.	Attacking ML supply chain: Introducing vulnerabilities or backdoors in third-party components. Model contamination (poisoning): Corrupting training data to cause misclassification or system unavailability.	Continuous vulnerability management and patching. CI/CD processes for consistent deployment. Isolation controls and secure architecture design.

Table 2: AI/ML Security Risk Overview

We analyze each responsibility in the following dimensions.

1. Evaluation Criteria: When discussing AI responsibility, consider quantifiable metrics for assessing the security impact of AI systems. By quantifying these aspects, stakeholders can better understand the associated risks of AI technologies and how to address those risks. Organizations must frequently evaluate their AI systems to ensure security and reliability. They should assess measurable things like how well the system handles attacks (adversarial robustness), whether it leaks sensitive data, how often it makes mistakes (false-positive rates), and whether the training data is reliable (data integrity). Evaluating and monitoring these critical measures as part of the organization's security plan will help improve overall security posture of AI systems.

2. RACI Model: This model helps clarify who is Responsible, Accountable, Consulted, and Informed (RACI) regarding AI decision-making and oversight. Applying the RACI model delineates roles and responsibilities in AI governance. This allocation of responsibilities is essential for secure AI systems. It is important to understand that depending on an organization's size and business focus, the specific roles and teams delineated in this white paper are for reference only. The emphasis should be on clearly outlining the key responsibilities first. Organizations can then determine the appropriate roles to map to those responsibilities, and subsequently, the teams to fill those roles. There may be some overlapping responsibilities across teams. The RACI framework defined herein aims to provide initial role and team designations to aid organizations in developing their own tailored RACI models. However, implementation may vary across companies based on their unique organizational structures and priorities.

3. High-level Implementation Strategies: This section outlines strategies for seamlessly integrating cybersecurity considerations into the Software Development Lifecycle (SDLC). Organizations must prioritize the enforcement of CIA principles—ensuring the confidentiality, integrity, and availability of data and systems. Access control mechanisms should be implemented rigorously to manage user permissions and prevent unauthorized access. Robust auditing mechanisms must track system activity and promptly detect suspicious behavior. Impact assessments should evaluate potential cybersecurity risks, focusing on identifying vulnerabilities and mitigating threats to safeguard sensitive information in AI systems

4. Continuous Monitoring and Reporting: Continuous Monitoring and Reporting ensures the ongoing security, safety, and performance of AI systems. Critical components include real-time monitoring, alerts for poor model performance or security incidents, audit trails/logs, and regular reporting, followed by action to implement improvements and resolve issues. Continuous Monitoring and Reporting helps organizations maintain transparency, enhance performance and accountability, and build trust in AI systems.

5. Access Control: Access control is crucial for securing AI systems. This includes strong API authentication/authorization policies, managing model registries, controlling access to data repositories, overseeing continuous integration and deployment pipelines (CI/CD), handling secrets, and managing privileged access. By defining user roles and permissions for various parts of the AI pipeline, sensitive data can be safeguarded, and models can't be tampered with or accessed without proper authorization. Implementing strong identity and access management not only protects intellectual property but also ensures accountability throughout AI workflows.

6. Adherence to Foundational Governance, Risk and Compliance, Security, Safety, and Ethical Guardrails: Emphasize adherence to guardrails based on industry best practices and regulatory requirements such as the following:

- NIST SSDF for secure software development
- NIST Artificial Intelligence Risk Management Framework (AI RMF)
- ISO/IEC 42001:2023 AI Management System (AIMS)
- ISO/IEC 27001:2022 Information Security Management System (ISMS)
- ISO/IEC 27701:2019 Privacy Information Management System (PIMS)
- ISO 31700-1:2023 Consumer protection Privacy by design for consumer goods and services
- OWASP Top 10 for LLM Applications
- NIST SP 800-53 Rev. 5 Security and Privacy Controls for Information Systems and Organizations
- General Data Protection Regulation (GDPR) on data anonymization and pseudonymization and guidance
- Guidance for tokenization on cloud-based services

Assumptions

This document assumes an industry-neutral stance, providing guidelines and recommendations that can be applicable across various sectors without a specific bias towards a particular industry.

Intended Audience

The white paper is intended to cater to a diverse range of audiences, each with distinct objectives and interests.

1. Chief Information Security Officers (CISOs): This white paper is specifically designed to address the concerns and responsibilities of CISOs. It provides valuable insights into integrating core security principles within AI systems. Please note that the role of Chief AI Officer (CAIO) is emerging in many organizations, and it's anticipated that a majority of related responsibilities defined in this white paper may shift from CISO to CAIO in the near future.

2. AI researchers, engineers, data professionals, scientists, analysts and developers: The paper offers comprehensive guidelines and best practices for AI researchers and engineers, aiding them in developing ethical and trustworthy AI systems. It serves as a crucial resource for ensuring responsible AI development.

3. Business leaders and decision makers: For business leaders and decision-makers such as CIO, CPO, CDO, CRO, CEO and CTO the white paper offers essential information and awareness for cyber security strategies related to AI system development, deployment, and lifecycle management.

4. Policymakers and regulators: Policymakers and regulators will find this paper invaluable as it provides critical insights to help shape policy and regulatory frameworks concerning AI ethics, safety, and control. It acts as a guide for informed decision-making in the realm of AI governance.

5. Investors and shareholders: Investors and shareholders will appreciate this white paper as it showcases an organization's commitment to responsible AI practices. It highlights the governance mechanisms in place to ensure ethical AI development, which can be vital for investment decisions.

6. Customers and the general public: This white paper offers transparency to customers and the general public regarding an organization's values and principles when it comes to developing secure AI models.

Responsibility Role Definitions

The following tables provide a general guide, illustrating various roles commonly found within organizations integrating or operating AI technologies. It's essential to recognize that each organization may define these roles and their associated responsibilities differently, reflecting their unique operational needs, culture, and the specific demands of their AI initiatives. Thus, while the table offers a foundational understanding of potential roles within AI governance, technical support, development, and strategic management, it is intended for reference purposes only. Organizations are encouraged to adapt and tailor these roles to best suit their specific requirements, ensuring that the structure and responsibilities align with their strategic objectives and operational frameworks. New roles can be defined further as AI technology evolves.

Management and Strategy

Role Name	Role Description
Chief Data Officer (CDO)	Oversees enterprise data management, policy creation, data quality, and lifecycle.
Chief Technology Officer (CTO)	Leads technology strategy and oversees technological development.
Chief Information Security Officer (CISO)	Oversees information security strategy and operations.
Business Unit Leaders	Directs business units and aligns AI initiatives with business objectives.
Chief AI Officer (CAIO)	Responsible for the strategic implementation and management of AI technologies within the organization.

Management	Oversees and guides the overall strategy, ensuring alignment with organizational goals, including CEO, COO, CIO,CTO, CISO, CAIO, CFO, etc.
Chief Cloud Officer	Leads cloud strategy, ensuring cloud resources align with business and technological goals.
Chief Architect	Leads architecture strategy, ensuring designs technology architecture to align with enterprise standards, processes, procedures, and targets. He/she makes technology choices, supervises the quality and implementation of designs, developing high performance architects within the organization.

Governance and Compliance

Role Name	Role Description	Category Name
Data Governance Board	Sets policies and standards for data governance and usage.	Governance and Compliance
Data Protection Officer	Oversees data protection strategy and compliance with data protection laws and regulations.	Governance and Compliance
Chief Privacy Officer	Ensures compliance with privacy laws and regulations.	Governance and Compliance
Legal Team/Department	Provides legal guidance on AI deployment and usage. Communicates on legal/regulatory obligations. Ensures appropriate provisions on the master agreements with AI vendors.	Governance and Compliance
Compliance Team/Department	Ensures adherence to internal and external compliance requirements.	Governance and Compliance
Data Governance Officer	Manages data governance within the organization, ensuring compliance with policies, data privacy laws, and regulatory compliance requirements.	Governance and Compliance
Information Security Officer	Authorizing official, management official, or information system owner for ensuring that the appropriate operational security posture is maintained for an information system or program including ISSO, ISM, and ISS.	Governance and Compliance

Technical and Security

Role Name	Role Description
Security Operations Team	Implements and monitors security protocols to protect data and systems.
Network Security Teams	Protect networks against threats and vulnerabilities.
Cloud Security Team	Ensures the security of cloud-based resources and services.
Cybersecurity Team	Protects against cyber threats, vulnerabilities, and unauthorized access to organizational assets.
IT Ops Team	Supports and maintains IT infrastructure, operational and secure.
Network Security Officer	Oversees the security of the network, ensuring data protection and threat mitigation.
Hardware Security Team	Secures physical hardware from tampering and unauthorized access.
System Administrators	Manages and configures IT systems and servers for optimal performance and security.

Operations and Development

Role Name	Role Description
Data Custodian	Responsible for the safe custody, transport, data storage, and implementation of business rules. This is any organization or person who acquires, manipulates, stores, or moves the data on behalf of the data owner.
AI Development Team	Develops and implements AI models and solutions.
Quality Assurance Team	Tests and ensures the quality of AI applications and systems.
AI Operations Team	Manages AI system operations for performance and reliability.
Application Development Teams	Develops applications, integrating AI functionalities as needed.
AI/ML Testing Team	Specializes in testing AI/ML models for accuracy, performance, and reliability.

Development Operations (DevOps) Team	Enhances deployment efficiency, maintaining operational stability.
Development Security Operations (DevSecOps) Team	Implements security across the software development lifecycle (SDLC).
AI Maintenance Team	Ensures AI systems and models are updated, optimized, and functioning correctly post-deployment.
Project Management Team	Oversees AI projects from initiation to completion, ensuring they meet objectives and timelines.
Operational Staff	Supports day-to-day operations, ensuring smooth integration and functioning of AI technologies.
Data Science Teams	Gathers and prepares data for use in AI model training and analysis.
Container Management Team	Manages containerized applications, facilitating deployment and scalability.
IT Operations Team	Ensures IT infrastructure is operational, supporting AI and technology needs.
AI Development Manager	Leads AI development projects, guiding the team towards successful implementation.
Head of AI Operations	Directs operations related to AI, ensuring efficiency and effectiveness of AI solutions.

Normative References

The documents listed below are essential for applying and understanding this document.

- Generative AI safety: Theories and Practices
- OpenAI Preparedness Framework
- Applying the AIS Domain of the CCM to Generative AI
- EU AI Act
- Biden Executive Order on Safe, Secure, and Trustworthy Artificial Intelligence
- OWASP Top 10 for LLM Applications
- CSA Cloud Controls Matrix (CCM v4)
- MITRE ATLAS™ (Adversarial Threat Landscape for Artificial-Intelligence Systems)
- NIST Secure Software Development Framework(SSDF)
- NIST Artificial Intelligence Trustworthiness and Risk Management Framework-
- General Data Protection Regulation (GDPR)
- OWASP LLM AI Cybersecurity & Governance Checklist
- OWASP Machine Learning - Top 10
- WEF Briefing Papers
- Building the AI-Powered Organization

1. Incorporating Data Security & Privacy in AI Training

AI is shifting from complex model-centric approaches towards data-centric approaches. Rather than relying primarily on intricate models trained on small data sets, AI now leverages massive data sets and open-ended data streams. However, this data-centric paradigm also raises valid concerns around data privacy, security, bias and appropriate use that the AI community must address responsibly. Data is transforming AI, but we must ensure it is ethically sourced and stewarded.

The following sections discuss important categories that are relevant to ensuring the security and privacy of training data in AI organizations. These categories include Data Authenticity, Anonymization/Pseudonymization, Data Minimization, Access Control to Data, and Secure Storage & Transmission. Each category is analyzed thoroughly with quantifiable evaluation criteria, clearly defined responsibilities through the RACI model, high-level implementation strategies, continuous monitoring and reporting mechanisms, access control mapping, and adherence to foundational guardrails based on industry best practices. This comprehensive approach ensures a structured, accountable, and efficient framework for managing the vital assets that fuel AI advancements, while also aligning with ethical mandates and operational excellence.

1.1 Data Authenticity and Consent Management

Data Authenticity in AI refers to the assurance that the data used for training, testing, and deploying AI models is genuine, accurate, and reliable. It's about verifying that the data has not been tampered with or altered in a way that could mislead the AI algorithms or result in inaccurate, biased, or unreliable model outputs.

Ensuring data authenticity is crucial because AI models heavily rely on data quality and integrity. If the data is inauthentic or manipulated, the model may learn incorrect patterns, leading to poor performance and potentially harmful decisions based on these predictions. Data authenticity is particularly important in fields where decisions based on AI models have significant consequences across various industries, such as education, healthcare, financial services, retail, manufacturing, governmental services and cybersecurity.

Additionally, it is essential to obtain proper data consent and comply with regulations like the General Data Protection Regulation (GDPR) when collecting and processing personal data for AI applications. GDPR mandates that organizations obtain explicit consent from individuals before collecting and processing their personal data, and it also grants individuals the right to access, rectify, and erase their data.

- **Evaluation Criteria:** Measure the percentage of data audited for authenticity periodically, aiming for 100% verification over a set period. Additionally, monitor compliance with data consent and GDPR regulations.

- Whenever possible, individuals should have the right to correct data about them. This is required by the FTC in insurance, for example: FTC pursues AI regulation, bans biased algorithms.
- **RACI Model:** Data Management Team (Responsible), Chief Data Officer (Accountable), Legal and Compliance Departments (Consulted), Security Team (Informed).
- **High-Level Implementation Strategy:** Implement a policy for regular data authenticity audits, which may involve techniques like data provenance checks and anomaly detection. Additionally, establish processes for obtaining data consent, ensuring data privacy, and adhering to GDPR regulations.
- **Continuous Monitoring and Reporting:** Regular reporting on the percentage of data verified for authenticity, unauthorized data changes, and compliance with data consent and GDPR regulations.

By ensuring data authenticity, obtaining proper data consent, and complying with regulations like GDPR, organizations can build trustworthy AI models while respecting individuals' privacy and rights over their personal data.

1.2 Anonymization and Pseudonymization

Anonymization and pseudonymization protect personal data privacy in Trustworthy AI systems as follows:

- **Anonymization** permanently removes identifiers from data, which makes it impossible to re-identify individuals, helping comply with data protection laws and, in many cases such as under the GDPR, even exempting the anonymized data from being under the scope of data protection regulations.
- **Pseudonymization** replaces identifiers with system-generated identifiers or artificial pseudonyms. Individuals remain linked to their data, but their real identity is protected. Pseudonymization is a requirement under certain data protection regulations, like the GDPR and HIPAA.
- **Evaluation Criteria:** Target a reduction in identifiable personal data by 99% through anonymization and pseudonymization techniques. It is important to distinguish between direct identifiers (like full names, SSNs, and credit card numbers) and quasi-identifiers. Direct identifiers require stricter reduction measures due to the ability to use them to identify an individual with a high degree of certainty. Additionally, direct identifiers like credit card numbers can lead to theft, while the ones like SSNs can lead to identity theft. Quasi-identifiers (age, zip code, and gender), while important for privacy, may undergo less stringent reduction, ensuring a balance between data protection and usability. That being said, many quasi-identifiers can instigate bias in AI (e.g., age, location, gender, race, sexuality). Moreover, certain quasi-identifiers may fall under special categories of personal data (Art. 9 GDPR - Processing of special categories of personal data) which require special caution, including religious beliefs, political affiliations, sexual orientation, and ethnic origin.

  Quasi-identifiers can also potentially be combined to re-identify a person. While quasi-identifiers are crucial for the usability and functionality of AI systems, especially in fields like healthcare or marketing, they present a re-identification risk. When different data sets are combined, even if they have been anonymized or pseudonymized, these quasi-identifiers can potentially be aligned to re-identify individuals.

For example, a data set containing anonymized medical records could be combined with a publicly available data set, such as voter registration records. If both data sets include detailed demographic information, it might be possible to match records based on these quasi-identifiers, leading to re-identification of individuals in the anonymized data set.

To mitigate this risk, a balanced approach is required. It is vital to continually assess the risk of re-identification, especially as data processing techniques evolve and become more sophisticated. Employing advanced techniques like differential privacy, where statistical noise is added to the data to prevent re-identification, can further enhance privacy protection. Additionally, regular audits and compliance checks are essential to ensure that the data anonymization and pseudonymization processes align with evolving data protection laws and regulations.

- **RACI Model**: Data Custodian(Responsible), Data Protection Officers (Consulted), Chief Privacy Officer (Accountable), Legal Team (Consulted), IT Team (Informed), Security team (Consulted), Data Governance team (Consulted), and Data Scientist (Informed).
- **High-Level Implementation Strategy:** Implementing state-of-the-art anonymization and pseudonymization techniques involves utilizing advanced cryptographic methods such as differential privacy, homomorphic encryption, and secure multiparty computation to protect sensitive data while maintaining its utility for analysis. For example, companies can employ techniques like k-anonymity, l-diversity, and t-closeness to ensure that individual identities are concealed within data sets, while still allowing for meaningful analysis. Additionally, techniques, such as tokenization and data masking, can be employed to replace sensitive data with nonsensitive equivalents, further enhancing privacy protection.
- **Continuous Monitoring and Reporting**: Regular assessments of the effectiveness of these techniques.
- **Access Control Mapping:** Restrict access to redaction or anonymization rules and de-pseudonymization tools.
- **Foundational Guardrails**: Follow the guidelines of the General Data Protection Regulation (GDPR) on data anonymization and pseudonymization and guidance for Tokenization on Cloud-based services.

1.3 Data Minimization

Data Minimization refers to the practice of using only the necessary amount of data required to achieve a specific purpose or function. This practice is a requirement within many data protection regulations, such as the GDPR. It is also one of the techniques that could be utilized to prevent the re-identification of anonymized data. This technique limits the amount and type of data collected, stored, and used for ML purposes. This practice aids in protecting the privacy and security of the data subjects and improves the performance and the efficiency of the ML models. In ML, data minimization involves carefully selecting features and data points essential for model training and performance, while excluding irrelevant or excessive data. This is related to the explainability, fairness, transparency, and privacy foundational pillars of trustworthy AI.

- **Evaluation Criteria:** Aim for at least a good percentage decrease in non-essential data collected based on the organization's business objectives and responsibilities.

- **RACI Model:** Data Collection Teams (Responsible), Data Privacy Office (Consulted), Data Governance Board (Accountable), Compliance Teams (Consulted), Security Team (Informed), Data Scientist (Informed).
- **High-Level Implementation Strategy:** Develop strict data collection guidelines focusing on minimal data acquisition.
- **Continuous Monitoring and Reporting:** Track the volume of data collected and assess its necessity.
- **Access Control Mapping:** Implement controls on who can authorize additional data collection.
- **Foundational Guardrails:** Employ privacy-by-design principles from the GDPR.

1.4 Access Control to Data

Access Control for data in Machine Learning involves managing and restricting who can access and interact with the data used for training, testing, and deploying ML models. This process ensures that only authorized individuals or systems have the ability to view, modify, or use the data. Effective access control is crucial in ML environments to protect sensitive information, maintain data integrity, and comply with privacy regulations. It typically involves authentication mechanisms to verify user identities, authorization protocols to grant users specific access rights based on their roles, and auditing systems to track data access and usage. Typically in an organization, AI models are run on data aggregated from various data sources or systems. As such AI models should respect the access control definitions and policies of the underlying data source systems. This means the AI models should only have access to the specific data that they are authorized to process, based on the access control rules and permissions defined by the data custodians or system administrators. Maintaining proper access control ensures data privacy, security, and compliance with regulatory requirements across the AI infrastructure and prevents unauthorized access or misuse of sensitive or confidential data by the AI models.

- **Evaluation Criteria:** Achieve less than 0.5% unauthorized data access incidents yearly
- **RACI Model:** Security Teams (Responsible), Chief Information Security Officer (Accountable), Data Governance Bodies, Data Custodians, IT Teams (Consulted), Operational Teams (Informed).
- **High-Level Implementation Strategy:** Implement a layered security model as part of your high-level implementation strategy for access control. This model should integrate not just robust authentication and authorization protocols, but also advanced technologies such as multifactor authentication (MFA), role-based access control (RBAC), and the principle of least privilege (PoLP).
- **Continuous Monitoring and Reporting**: Monitor access logs and conduct audits. Use tools that can flag risk-based access to critical models, and generated models and data.
- **Access Control Mapping:** Regularly monitor and manage access permissions.
- **Foundational Guardrails:** Implement best practices per ISO/IEC 42001, ISO/IEC 27001, ISO/IEC 27701, NIST 800-53, and OWASP Top 10 A07:2021-Identification and Authentication Failures.

1.5 Secure Storage & Transmission

In Machine Learning, secure storage, and transmission are critical for protecting sensitive data. Secure storage involves encrypting data at rest to prevent unauthorized access, employing robust access controls, and conducting regular security audits. For secure transmission, data in transit is encrypted using protocols like Transport Layer Security (TLS) or field-level or envelope encryption. This ensures that data remains confidential and intact while being transferred between systems or networks. These practices enhance the security of the data and the ML models by preventing unauthorized access, use, and disclosure of the data, as well as malicious or accidental modification, deletion, or corruption of the data.

- **Evaluation Criteria:** Maintain encryption standards at 256-bit AES or higher for 100% of data in transit and at rest.
- **RACI Model:** Security Teams (Responsible), Chief Information Security Officer (Accountable), Compliance and Legal Teams (Consulted), Management (Informed).
- **High-Level Implementation Strategy:** Invest in advanced encryption technologies and automated deletion of AI data and models by policy.
- **Continuous Monitoring and Reporting:** Use tools for real-time security monitoring.
- **Access Control Mapping**: Integrate secure storage and transmission with access controls.
- **Foundational Guardrails:** Follow the National Institute of Standards and Technology (NIST) guidelines and privacy laws to protect data in transit and at rest.

2. Model Security

Model Security is a multi-faceted task that encompasses a wide array of components. These include Access Controls on Model APIs, Authentication and Authorization Frameworks, Rate Limiting, Model Lifecycle Management, Secure Model Runtime Environment, Hardware-Based Security Features, Network Security Controls, OS hardening, and Secure Configurations, Security in container and cloud environments. For each of these critical areas, we will explore evaluation criteria, assign responsibilities using the RACI model, outline High-Level Implementation Strategies, establish Continuous Monitoring and Reporting mechanisms, map Access Control, and refer to Foundational Guardrails from standards like NIST AI RMF, NIST SSDF, NIST 800-53, and CSA CCM.

2.1. Access Controls to Models

Access controls are pivotal in securing AI models and ensuring only authorized personnel and systems interact with sensitive data and functionalities. Within the realm of AI model governance, access control measures must be robust, adaptable, and aligned with organizational and industry security standards. From authentication and authorization frameworks to rate limiting and lifecycle management, the integrity of AI models hinges upon strong and flexible access control protocols. These protocols dictate who can access AI models, when they can do so, and under what circumstances. As organizations navigate the complexities of AI deployment, implementing comprehensive access control strategies becomes imperative to mitigate risks, safeguard intellectual property, and uphold regulatory compliance standards. Additionally, we emphasize integrating AI model access controls with the organization's existing security frameworks to enhance overall system resilience. This involves establishing granular access policies that dictate who can interact with AI models and under what circumstances. Moreover, implementing robust authentication mechanisms, such as multi-factor authentication and role-based access control, can further fortify the security posture of AI systems. Regular audits and monitoring of access logs are essential to detect and respond to any unauthorized access attempts promptly. By tightly integrating AI model access controls with existing security frameworks, organizations can bolster their defense against cyber threats and ensure the integrity and confidentiality of their AI systems and data.

2.1.1 Authentication and Authorization Frameworks

Authentication and authorization frameworks for Machine Learning models are integral for security, ensuring that access to ML models and related data is strictly controlled and managed. Authentication verifies the identity of users or systems, often using methods such as passwords, tokens, or biometric verification. At the same time, authorization determines their access level, defining who can view, edit, or use the models based on established roles and permissions. These frameworks are essential for safeguarding sensitive information, preserving data integrity, and adhering to privacy and security regulations, thereby preventing unauthorized access or modifications to the ML models and their data. In AI, a particular validation is the approved, appropriate use of AI data and models by any user and entity based on purpose and context. This aspect is embedded in the AI digital rights and is foundational to access and authorization.

- **Evaluation Criteria**: Achieve 100% coverage of authentication and authorization frameworks across AI models that are not accessed via APIs (covered in 1.2.1).
- **RACI Model**: Security Team (Responsible), Chief Information Security Officer (Accountable), Legal Team (Consulted), AI Development Teams (Informed).
- **High-Level Implementation Strategy:** Develop and implement comprehensive frameworks for secure AI model access.
- **Continuous Monitoring and Reporting:** Regularly audit authentication and authorization mechanisms.
- **Access Control Mapping:** Customize access based on model-specific requirements.
- **Foundational Guardrails:** Use NIST 800-207, NIST 800-53, NIST SP 800-63, and NIST AI RMF for risk management.

2.1.2. Model Interfaces Rate Limiting

Model Interfaces Rate Limiting in Machine Learning (ML) involves restricting the number of requests a user or system can make to an ML model within a given time frame. This practice is crucial for managing the load on the model, preventing abuse (such as Denial-of-Service (DoS) attacks), and ensuring equitable resource distribution among users. Rate limiting can be implemented at various interface levels where users interact with the ML model, such as APIs or web interfaces. Controlling the request rate helps maintain the model's performance, stability, and availability, ensuring it continues to operate efficiently and reliably, even under high demand or potential attack scenarios.

- **Evaluation Criteria:** Reduced Downtime due to Denial of Service (DoS) or Distributed Denial of Service (DDoS) attacks.
- **RACI Model:** Platform Support team (Responsible), Solution Owner (Accountable), Data Scientist (Consulted), Risk Management Team (Consulted), AI Model Users (Informed).
- **High-Level Implementation Strategy:** Enforce rate limiting to prevent overuse or abuse of AI model interfaces.
- **Continuous Monitoring and Reporting:** Track usage patterns and adjust rate limits accordingly.
- **Access Control Mapping**: Implement user-based rate limiting strategies.
- **Foundational Guardrails:** Follow OWASP LLM 04: Model Denial of Service.

2.1.3. Access Control in Model Lifecycle Management

Access Control in Model Lifecycle Management for Machine Learning (ML) involves managing and regulating access and interaction with ML models throughout their lifecycle – stages of development, deployment, and maintenance. This process ensures that only authorized personnel or systems can interact with the ML models at various stages, thereby protecting the models from unauthorized access or changes, which could lead to compromised model integrity or performance issues. Implementing robust access control is crucial for maintaining the security and efficacy of ML models, as it helps prevent potential data breaches, misuse of models, and ensures compliance with data privacy and security regulations. By carefully controlling access at each stage of the model's lifecycle, organizations can safeguard their ML assets while fostering a secure and efficient ML development environment. Access control in model lifecycle management improves transparency and accountability of the ML models and data, providing consistent and clear policies and procedures for data access, usage and by documenting and logging the data provenance and lineage. This area is connected to the privacy, transparency, and accountability fundamental pillars of trustworthy AI.

- **Evaluation Criteria:** Ensure access to AI models and data is restricted to authorized users and systems across all stages of the model lifecycle.
- **RACI Model:** AI Model Governance Team (Responsible), Chief Data Officer (Accountable), Security Team, Legal Team, Compliance Team (Consulted), Operational Staff (Informed)
- **High-Level Implementation Strategy**:
    - Classify data and models based on sensitivity levels.
    - Define distinct access control rules, mapped to user roles, for each phase of the model lifecycle.
    - Integrate access controls with existing IAM (Identity and Access Management) solutions.
    - Log and audit all access requests and data/model usage.
- **Continuous Monitoring and Reporting**:
    - Send alerts on unauthorized access attempts.
    - Perform user access reviews and re-certification.
    - Conduct audits of access and controls regularly.
    - Establish alert thresholds for suspicious activities.
- **Access Control Mapping**:
    - Development Stage: Restrict access to data scientists, and ML engineers
    - Testing Stage: Add access for the quality assurance team
    - Production Stage: Grant tightly controlled access to production systems
- **Foundational Guardrails:**
    - Align with standards like NIST 800-53, NIST AI RMF Framework
    - Validate controls against best practices frameworks like CSA CCM.

2.2. Secure Model Runtime Environment

Constructing resilient systems for secure AI model runtime environments necessitates converging robust hardware, network, and software security controls. With an overarching focus on safeguarding AI deployments against evolving threats, organizations meticulously engineer and fortify their runtime environments to uphold integrity, confidentiality, and availability. From hardware-based security features leveraging trusted execution environments to network security controls like firewalls and segmentation, each component is intricately woven into the fabric of defense-in-depth strategies. With an unwavering commitment to compliance with industry standards such as NIST AI RMF, NIST 800-53, and CSA CCM, teams collaborate across disciplines to orchestrate implementing, monitoring, and governing these critical security measures.

2.2.1. Hardware-Based Security Features

Hardware-Based Security Features for Machine Learning (ML) models encompass physical and architectural elements in computing hardware that bolster the security of ML applications. This includes Trusted Execution Environments (TEEs), and confidential computing for isolated and secure processing, Secure Enclaves for protecting sensitive code and data, Hardware Security Modules (HSMs) for secure cryptographic operations, Secure Boot mechanisms to ensure trusted software initialization, and Physical Anti-Tamper Mechanisms to prevent unauthorized physical access. These features are vital in providing a foundational layer of security, particularly crucial in high-stakes sectors like finance, healthcare, or defense, where ML models handle sensitive data and require robust protection against various threats, including tampering and unauthorized access.

- **Evaluation Criteria**: The corporate-defined percentage of AI systems where applicable, use hardware-based security features.
- **RACI Model**: Hardware Security Team (Responsible), Chief Technology Officer (Accountable), Procurement Department (Consulted), System Administrators (Informed).
- **High-Level Implementation Strategy**: Integrate trusted execution environments (such as NVIDIA's confidential computing approach), GPUs, TPUs, and other hardware security measures in AI systems.
- **Continuous Monitoring and Reporting**: Regularly check the integrity and functionality of hardware security.
- **Access Control Mapping**: Ensure hardware security settings are accessible only to authorized personnel.
- **Foundational Guardrails**: Implement NIST AI RMF and NIST 800-53 guidelines.

2.2.2. Network Security Controls

Network Security Controls for Machine Learning (ML) models are measures and protocols implemented to protect ML models and their associated data from network-based threats and vulnerabilities. Adopt a Zero Trust architecture and segregate AI systems from the broader network. It reduces the attack surface by isolating AI systems and ML models, making it harder for attackers to move laterally within the network. These controls are essential for safeguarding the data used in ML models during transmission, preventing unauthorized access, and ensuring the integrity and confidentiality of ML communications. Key network security controls include the use of next generation firewalls to monitor and control incoming and outgoing network traffic, encryption protocols like TLS for securing data in transit, intrusion detection and prevention systems (IDPS), Web Application Firewall (WAF) to identify and mitigate attacks, virtual private networks (VPNs) for creating secure communication channels, and secure API gateways to manage and authenticate API calls to ML models. These measures are crucial in ML environments where data and model security are paramount, especially when models are accessed or managed over networks, including the Internet or cloud environments.

- **Evaluation Criteria**: Achieve 100% compliance with network security policies across AI systems.
- **RACI Model:** Network Security Team (Responsible), Chief Information Security Officer (Accountable), IT Operations (Consulted), All Network Users (Informed).
- **High-Level Implementation Strategy:** Implement comprehensive network security measures such as firewalls and segmentation.
- **Continuous Monitoring and Reporting:** Regularly monitor network ingress and egress traffic and traffic within the infrastructure and enforce control compliance. Incorporate regular penetration testing and vulnerability assessments is an proactive approach that will help identify potential weaknesses in network security controls before they can be exploited, ensuring the robust protection of ML models and their data.

- **Access Control Mapping:** Tailor network access controls to specific roles and model requirements.
- **Foundational Guardrails:** CIS Controls V8 related to Network Security

2.2.3. OS-Level Hardening and Secure Configurations

OS-Level Hardening and Secure Configurations for Machine Learning (ML) models involve reinforcing the underlying operating system (OS) on which ML models and applications run, to mitigate risks and reduce vulnerabilities. This process is crucial for creating a secure environment for ML operations, as the OS forms the foundational layer for these applications. Key aspects include:

Regular Updates and Patch Management: Keeping the OS and its components up-to-date with the latest security patches and updates to protect against known vulnerabilities.

Minimal Installation: Removing or disabling unnecessary services, applications, and features in the OS that are not needed for ML operations, minimizing potential attack surfaces.

Configuring Security Settings: Adjusting OS settings to enhance security, such as enabling firewalls, configuring user permissions, and implementing security policies that dictate how the system is accessed and used.

User Access Control: Implementing strict user access controls, ensuring that only authorized users have access to the ML system, and applying the principle of least privilege, where users are granted only the access necessary to perform their tasks.

Monitoring and Auditing: Setting up monitoring and auditing tools to track activities and changes in the OS, which can help detect and respond to security incidents.

Secure Communication Protocols: Ensuring all communications to and from the ML system are encrypted and secure.

These measures help create a robust security posture for ML systems, protecting them from various threats that could compromise the integrity, confidentiality, and availability of ML models and their data.

- **Evaluation Criteria**: Ensure 100% of AI systems operate with hardened OS and secure configurations.
- **RACI Model**: System Administration Team (Responsible), Chief Information Security Officer (Accountable), Security Team (Consulted), End-Users (Informed).
- **High-Level Implementation Strategy**: Apply best practices in OS hardening and secure configuration settings.
- **Continuous Monitoring and Reporting**: Monitor compliance and vulnerability to security threats.
- **Access Control Mapping**: Restrict who can alter system configurations.
- **Foundational Guardrails**: Utilize NIST 800-53, CIS and DISA STIGs Benchmarks for secure configurations.

2.2.4. K8s and Container Security

Kubernetes (K8s) and Container Security for Machine Learning (ML) refers to the set of practices and tools designed to secure containerized ML applications and their deployment environments. Kubernetes, a container orchestration platform, and containerization technologies are widely used for deploying and managing ML models and workloads. Security in this context involves ensuring that containers are securely configured and reducing vulnerabilities under a good risk management framework, implementing robust Kubernetes cluster security (including network policies, access controls, and pod security), and securing the communication channels within the Kubernetes environment. This also encompasses managing container privileges, regularly scanning for vulnerabilities, and enforcing policies governing how containers are run and interact in ML workflows, safeguarding ML models and data against unauthorized access, breaches, and other security threats in a containerized deployment environment.

- **Evaluation Criteria:** Target reduced security breach rate in container environments within AI systems.
- **RACI Model:** Container Management Team (Responsible), CTO (Accountable), Security Team (Consulted), DevOps Team (Consulted), Application Development Teams (Informed).
- **High-Level Implementation Strategy:** Ensure secure deployment of AI applications in containerized environments.
- **Continuous Monitoring and Reporting:** Conduct regular security assessments of container orchestration tools.
- **Access Control Mapping:** Define and enforce stringent access policies for container environments.
- **Foundational Guardrails:** Follow OWASP Kubernetes Top Ten, NIST SSDF, CNCF Security Whitepaper, CIS Benchmarks, NIST 800-190, and NIST 800-53 best practices.

2.2.5. Cloud Environment Security

Cloud Environment Security for Machine Learning (ML) models encompasses the strategies and measures implemented to protect ML models and their associated data in cloud-based infrastructures. This involves securing data storage and processing within the cloud, managing access controls to cloud resources, encrypting data at rest and in transit, and ensuring the security of ML models deployed on cloud platforms. It also includes regular vulnerability assessments, compliance with cloud-specific security standards, and implementing best practices for identity and access management. This security is crucial to prevent unauthorized access, data breaches, and other cyber threats, ensuring the integrity and confidentiality of ML models and data in cloud environments' dynamic and distributed nature.

- **Evaluation Criteria:** Strive for 100% adherence to cloud security policies in AI deployments.
- **RACI Model:** Cloud Security Team (Responsible), Chief Information Security Officer (Accountable), IT Governance (Consulted), Cloud Service Users (Informed).
- **High-Level Implementation Strategy:** Implement robust cloud security measures for AI systems.
- **Continuous Monitoring and Reporting:** Use cloud-specific monitoring tools to detect and alert on threats.
- **Access Control Mapping:** Customize access controls for cloud-based AI applications.

- **Foundational Guardrails:** Employ CSA CCM. Adhere and prioritize cloud-native application protection platforms (CNAPPs).

2.3 Vulnerability and Patch Management

The following responsibilities and approaches should be integrated into the organization's broader security and development processes, ensuring that AI systems are developed, deployed, and maintained securely and effectively. Regular reviews and updates to these processes will help adapt to evolving threats and technologies as it identifies, prioritizes, and applies fixes for the software flaws and weaknesses that could expose the AI systems to attacks, errors, or failures. These following techniques are aspects of AI organizational responsibilities as they help ensure the security, reliability, and trustworthiness of AI systems

2.3.1 ML Code Integrity Protections

ML Code Integrity Protections refer to the measures and practices employed to ensure the security and integrity of the source code used in Machine Learning (ML) applications. This involves safeguarding the code against unauthorized modifications, ensuring its authenticity, and maintaining its quality throughout the development and deployment process. Key practices include implementing version control systems to track and manage changes, using code signing to verify the authenticity of the code, conducting regular code reviews and audits to detect vulnerabilities, and employing static and dynamic code analysis tools to identify potential security issues. These protections are vital in maintaining the trustworthiness of ML applications, preventing malicious code injection, and ensuring that the ML models perform as intended without being compromised.

- **Evaluation Criteria:**
    - Percentage of ML code covered by integrity protections
    - Frequency of integrity checks
    - Percentage of critical vulnerabilities addressed
- **Responsibility (RACI Model):** AI Development Team (Responsible), AI Development Manager (Accountable), Security Team, DevOps Team (Consulted), Compliance Team (Informed).
- **High-Level Implementation Strategy:**
    - Implement automated integrity checks for ML models.
    - Integrate integrity checks into CI/CD pipelines.
    - Establish regular vulnerability assessments for ML code.
    - Enforce secure coding practices specific to ML algorithms.
    - Implement runtime monitoring for anomalous behavior in ML models.
- **Continuous Monitoring and Reporting:**
    - Utilize monitoring tools to detect anomalies in ML model behavior.
    - Establish incident response procedures for detected anomalies.
    - Regularly report on integrity check results and any anomalies found.
- **Access Control Mapping:**
    - Grant access to ML code repositories based on roles and responsibilities.
    - Implement least privilege principles for accessing ML code.
    - Utilize multi factor authentication for sensitive ML code repositories.

- **Foundational Guardrails:**
    - Reference NIST AI RMF for guidelines on securing ML systems.
    - Implement the security controls recommended in NIST 800-53 that are relevant to ML systems.
    - Adhere to CSA CCM for cloud-specific security considerations.

2.3.2 Version Control Systems for ML Training and Deployment Code

Version Control Systems for ML Training and Deployment Code are essential tools in managing machine learning (ML) projects, enabling teams to track and manage changes to the codebase used for training and deploying ML models. These systems facilitate collaboration among developers and data scientists by maintaining a history of changes, allowing for easy tracking of modifications, and supporting the rollback to previous versions, if needed. They are crucial in handling various versions of ML models and their associated data sets, ensuring consistency and reproducibility in ML experiments and deployments. By using version control, teams can efficiently manage the lifecycle of ML models, from development and testing to deployment and maintenance, while ensuring the integrity and traceability of the code and models throughout the process.

- **Evaluation Criteria:**
    - Percentage of code under version control
    - Frequency of commits and updates
    - Compliance with version control policies
- **Responsibility (RACI Model):**
    - Responsible: Development Team
    - Accountable: Development Manager
    - Consulted: DevOps Team, QA Team
    - Informed: Security Team
- **High-Level Implementation Strategy:**
    - Implement a centralized version control system (e.g., Git)
    - Enforce branching and merging policies
    - Automate code reviews and checks before merging
    - Implement version tagging for release management
- **Continuous Monitoring and Reporting:**
    - Monitor commit activity and identify irregular patterns
    - Establish alerts for unauthorized changes or unusual activity
    - Regularly review version control logs for compliance
- **Access Control Mapping:**
    - Define access control lists for repositories based on roles
    - Implement two-factor authentication for repository access
    - Audit repository access regularly to ensure compliance
- **Foundational Guardrails:**
    - Follow NIST SSDF for secure software development guidelines.
    - Implement controls outlined in NIST 800-53 related to version control and change management.

2.3.3 Code Signing to Validate Approved Versions

Code Signing to Validate Approved Versions in the context of Machine Learning (ML) is a security practice where digital signatures are used to verify the authenticity and integrity of the software code used in ML models. This process involves attaching a cryptographic signature to the code, typically after it has been reviewed and approved for deployment. The signature acts as a seal that verifies that the code has not been altered or tampered with since it was signed. In ML workflows, code signing is important for ensuring that the code used for training, testing, and deploying ML models is the exact, authorized version and has not been maliciously modified. This practice helps maintain trust in the ML software supply chain, especially when models are distributed across different environments or shared among multiple teams or organizations.

- **Evaluation Criteria:**
    - Percentage of code signed with approved certificates
    - Compliance with code signing policies
    - Frequency of code signing checks
- **Responsibility (RACI Model):**
    - Responsible: Development Team
    - Accountable: Development Manager
    - Consulted: Security Team, Release Management Team
    - Informed: Compliance Team
- **High-Level Implementation Strategy:**
    - Implement code signing during the build process
    - Manage code signing certificates securely
    - Automate code signing checks before deployment
    - Implement timestamping for signed code to prevent replay attacks
- **Continuous Monitoring and Reporting:**
    - Monitor code signing activity and certificate usage
    - Implement alerts for unauthorized code signing attempts
    - Regularly review code signing logs for compliance
- **Access Control Mapping:**
    - Restrict access to code signing infrastructure to authorized personnel only
    - Implement role-based access controls for code signing certificates
    - Regularly review access logs for code signing infrastructure
- **Foundational Guardrails:**
    - Reference NIST 800-53 controls related to code signing and integrity checking.

2.3.4 Infrastructure as Code Approaches

Infrastructure as Code (IaC) Approaches for Machine Learning (ML) models involve managing and provisioning the computing infrastructure through machine-readable definition files, rather than through physical hardware configuration or interactive configuration tools. This practice enables the automated setup, configuration, and management of the infrastructure required for ML models, such as servers, storage, and networking resources, in a consistent and repeatable manner. IaC allows ML teams to quickly deploy and scale their models in diverse environments (like cloud, on-premises, or hybrid setups) with minimal manual intervention. This approach enhances the efficiency and reliability of infrastructure deployment for ML models and ensures that the infrastructure state is maintainable, version-controlled, and compliant with defined standards, leading to improved collaboration and reduced operational risks in ML projects.

- **Evaluation Criteria:**
  - Percentage of infrastructure managed as code
  - Compliance with IaC best practices
  - Frequency of infrastructure updates and reviews
- **Responsibility (RACI Model):**
  - Responsible: DevOps Team
  - Accountable: DevOps Manager
  - Consulted: Development Team, Security Team
  - Informed: Operations Team
- **High-Level Implementation Strategy:**
  - Utilize IaC tools such as Terraform or AWS CloudFormation
  - Implement version control for infrastructure code
  - Automate testing and validation of infrastructure changes
  - Implement infrastructure drift detection and remediation
- **Continuous Monitoring and Reporting:**
  - Monitor infrastructure changes and drift
  - Implement alerts for unauthorized or unexpected changes
  - Regularly review infrastructure code for compliance with standards
- **Access Control Mapping:**
  - Limit access to infrastructure code repositories based on roles
  - Implement role-based access controls for IaC tools
  - Audit access to infrastructure code repositories regularly
- **Foundational Guardrails:**
  - Follow NIST 800-53 guidelines for secure configuration and management of infrastructure.
  - Implement security controls outlined in CSA CCM for cloud infrastructure.

2.4 MLOps Pipeline Security

The key areas in MLOps pipeline security include source code scans for vulnerabilities, testing model robustness against attacks, validating pipeline integrity at each stage, and monitoring automation scripts.

2.4.1. Source Code Scans for Vulnerabilities

Source Code Scans for Vulnerabilities in ML code involve using automated tools to systematically examine the source code of Machine Learning (ML) applications for security vulnerabilities and coding flaws. This practice is crucial in the early detection and remediation of potential security weaknesses that could compromise the ML system. The scans typically check for common vulnerabilities like buffer overflows, injection flaws, insecure library use, and coding practices that may lead to unintended behavior or performance issues. By regularly scanning ML code, developers and data scientists can ensure that the codebase adheres to security best practices and standards, reducing the risk of exploits and breaches.

This proactive approach is essential in maintaining the integrity and trustworthiness of ML applications, particularly when they handle sensitive data or are used in critical systems.

- **Evaluation Criteria**: Assess effectiveness by the percentage of the source code used in model training and deployment scanned and the frequency of these scans.
- **RACI Model**:
    - Responsible: Model Development Team
    - Accountable: Chief Information Security Officer (CISO)
    - Consulted: Application Security Team
    - Informed: Development Operations (DevOps) Team
- **High-Level Implementation Strategy**: Implement automated tools for regular source code vulnerability scanning and integrate these tools into the development lifecycle.
- **Continuous Monitoring and Reporting**: Set up continuous monitoring systems for code scans and report findings in real-time.
- **Access Control Mapping**: Ensure that only authorized personnel can access and modify source code and scanning tools.
- **Foundational Guardrails**: Utilize NIST 800-53 standards and CSA CCM as guiding principles.

2.4.2. Testing Model Robustness Against Attacks

Testing Model Robustness Against Attacks in Machine Learning (ML) involves evaluating ML models to determine how well they can withstand and react to various adversarial attacks or manipulations. This testing is crucial for identifying potential vulnerabilities in ML models that could be exploited to produce incorrect outcomes, cause system malfunctions, or leak sensitive information. It typically includes probing the model with intentionally crafted inputs (adversarial examples) to assess its resilience to such attacks, analyzing the model's behavior under different threat scenarios, and verifying its ability to maintain performance and accuracy in the face of unexpected or malicious input. Robustness testing helps ensure the reliability and security of ML models, especially in applications where robust decision-making is critical, such as in autonomous vehicles, financial systems, or healthcare diagnostics.

- **Evaluation Criteria**: Measure the effectiveness by the percentage of models tested against attacks and the frequency of these tests.
- **RACI Model**:
    - Responsible: AI/ML Testing Team
    - Accountable: Head of AI/ML Development
    - Consulted: Security Analysts
    - Informed: AI/ML Development Team
- **High-Level Implementation Strategy**: Develop a robust testing framework for models, focusing on identifying and mitigating potential attack vectors.
- **Continuous Monitoring and Reporting**: Implement mechanisms for ongoing assessment of model robustness and update stakeholders regularly.
- **Access Control Mapping**: Control access to testing frameworks and models appropriately.
- **Foundational Guardrails**: Refer to best practices from NIST AI RMF, NIST AI 100-2 E2023 **Adversarial Machine Learning**: A Taxonomy and Terminology of Attacks and Mitigations, and other relevant standards.

2.4.3. Validating Pipeline Integrity at Each Stage

Validating Pipeline Integrity at Each Stage in Machine Learning (ML) refers to the process of ensuring each phase of the ML pipeline — from data collection and preprocessing to model training, evaluation, and deployment — operates correctly and securely. This involves conducting thorough checks and validations at every stage to safeguard against data corruption, unauthorized access, and other vulnerabilities that could compromise the pipeline's performance and the model's accuracy. Such validation includes verifying data quality and consistency, ensuring secure data handling practices, assessing the reliability and reproducibility of model training processes, and confirming that deployment mechanisms are secure and function as intended. This comprehensive approach to validation is critical for maintaining the overall integrity and efficacy of ML pipelines, especially in complex or high-stakes environments where the accuracy and reliability of ML models are paramount.

- **Evaluation Criteria**: The focus is on meticulously monitoring the integrity of the MLOps pipeline. This is achieved by examining the percentage of stages that undergo validation and assessing the depth and thoroughness of these validation processes. The goal is to ensure that every stage of the pipeline functions as intended and adheres to established standards and best practices.
- **RACI Model**:
    - Responsible: The DevOps Team is entrusted with the day-to-day task of validating each pipeline stage. They are the primary executors of the validation processes, ensuring that each stage of the MLOps pipeline is thoroughly checked and validated.
    - Accountable: The Head of Engineering holds the ultimate accountability for the overall integrity of the MLOps pipeline. This role involves overseeing the validation process and ensuring the pipeline meets the necessary standards and requirements.
    - Consulted: The Quality Assurance (QA) Team is consultative, providing expert advice and input into the validation processes. Their involvement is crucial in defining the validation standards and reviewing validation outcomes.
    - Informed: Project Managers are kept informed about the status and outcomes of the pipeline validation processes. This ensures they are aware of any potential issues or changes that might impact project timelines or deliverables.
- **High-Level Implementation Strategy**: A systematic approach is crucial for validating the integrity of each stage of the MLOps pipeline. This strategy encompasses establishing clear procedures and standards for data and process integrity. It involves defining specific validation tests and checks for each pipeline stage, ensuring that every element, from data ingestion to model deployment, functions correctly and securely.
- **Continuous Monitoring and Reporting**: An essential component of validating pipeline integrity is implementing a system for continuous validation and real-time reporting. This system should detect any discrepancies or anomalies as they occur, allowing immediate action to rectify issues. Continuous monitoring ensures that the pipeline remains secure and efficient at all times.
- **Access Control Mapping**: Strict access controls are vital for maintaining the integrity of each stage of the MLOps pipeline. This involves defining and enforcing who has access to various parts of the pipeline, under what conditions, and with what level of authority. Such controls are essential to prevent unauthorized access or modifications that could compromise the pipeline's integrity.
- **Foundational Guardrails**: To ensure best practices are followed, it's important to align the pipeline validation processes with established industry standards and guidelines, such as those outlined in the NIST Secure Software Development Framework (SSDF). Adhering to such

frameworks provides a benchmark for security and efficiency, guiding the development and maintenance of a robust and reliable MLOps pipeline.

2.4.4. Monitoring Automation Scripts

This task involves vigilant oversight of all scripts that automate various stages of the machine learning lifecycle, from data preprocessing to model deployment and management.

- **Evaluation Criteria**: The effectiveness of monitoring automation scripts is quantified by two main metrics: the percentage of automation scripts under continuous surveillance and the frequency of these monitoring activities. This evaluation helps ensure that all scripts function correctly and efficiently and potential issues are promptly identified and addressed.
- **RACI Model**:
    - Responsible: The IT Operations Team is primarily responsible for the day-to-day monitoring of automation scripts within the MLOps pipeline. Their role includes overseeing the scripts' execution, ensuring their performance and security, and identifying operational issues.
    - Accountable: The Chief Technology Officer (CTO) holds overall accountability for the management and security of the automation scripts. The CTO ensures monitoring strategies are effectively implemented and aligned with the organization's technological goals.
    - Consulted: The DevOps Team provides crucial input and expertise, particularly in script deployment and operational efficiencies. Their consultation is vital in enhancing the monitoring processes and tools used within the pipeline.
    - Informed: All stakeholders in the MLOps pipeline, including data scientists, ML engineers, and project managers, are kept informed about the status and performance of the automation scripts. This ensures a cohesive and transparent operation across all stages of the pipeline. To guarantee a cohesive and transparent operation throughout the MLOps pipeline, it is crucial that all stakeholders—including data scientists, machine learning engineers, and project managers—are thoroughly informed about the status and performance of automation scripts. This practice not only fosters a unified approach across all stages of the pipeline but also ensures that decision-making is based on up-to-date and accurate information, enhancing the overall security and efficiency of AI deployments.
- **High-Level Implementation Strategy**: Implementing a comprehensive monitoring system for all automation scripts is essential. This system should track the scripts' performance and efficiency to ensure their compliance with defined standards and practices.. It should integrate seamlessly into the MLOps pipeline, providing real-time insights into the behavior and output of the automation scripts.
- **Continuous Monitoring and Reporting**: Continuous surveillance is key in promptly identifying and addressing issues with the automation scripts. The monitoring system should be capable of generating real-time alerts and reports, providing timely information on script performance, errors, or security concerns. This continuous feedback loop is vital for maintaining the operational integrity of the MLOps pipeline.
- **Access Control Mapping**: Strict access controls are necessary to safeguard the automation scripts and the overall pipeline. This involves defining who can access, modify, or execute the

scripts. Access should be based on role-specific requirements, ensuring only authorized personnel can make changes, thereby reducing the risk of unauthorized or harmful modifications.
- **Foundational Guardrails**: Employing best practices from established frameworks like CSA CCM and relevant guidance provided by NIST.

2.5 AI Model Governance

AI model governance encompasses several key areas: model risk assessments, business approval procedures, model monitoring requirements, and new model verification processes.

2.5.1. Model Risk Assessments

Model Risk Assessments in Machine Learning (ML) involve systematically evaluating the potential risks associated with deploying and using ML models. This assessment aims to identify and quantify the possible adverse impacts of model inaccuracies, biases, or failures. Key focus areas include evaluating the model's accuracy and generalizability across different data sets and scenarios, assessing the potential for biased or unfair outcomes, and understanding the model's behavior in edge cases or under adversarial conditions. Model risk assessments also consider the consequences of model failure, particularly in critical applications such as healthcare, finance, or public safety. This process is essential for identifying and mitigating risks to ensure that ML models are deployed responsibly and safely by clearly understanding their limitations and potential impacts.

- **Evaluation Criteria**: Assess the percentage of models undergoing risk assessment and the comprehensiveness of these assessments.
- **RACI Model**:
    - Responsible: Risk Management Team, Data Governance Committee
    - Accountable: Chief Risk Officer (CRO)
    - Consulted: AI Ethics Board, Legal Counsel
    - Informed: Data Science Teams
- **High-Level Implementation Strategy**: Develop a framework for assessing risks associated with AI models, including bias, fairness, and data privacy.
- **Continuous Monitoring and Reporting**: Implement tools for ongoing risk monitoring and establish a protocol for reporting risks as part of the Software Supply Chain.
- **Access Control Mapping**: Ensure access to risk assessment tools and data is strictly controlled and monitored.
- **Foundational Guardrails**: Align with NIST AI RMF and NIST 800-53 for risk management practices.

2.5.2. Business Approval Procedures

This encompasses the formal processes and protocols an organization follows to approve ML models for deployment into production. These procedures ensure that any ML model aligns with the business objectives, complies with regulatory and ethical standards, and meets the required performance benchmarks. Typically, this involves a multi-step review process where various stakeholders, including data scientists, business analysts, risk management teams, and sometimes legal and compliance departments, evaluate the model's effectiveness, reliability, and potential business impact. Key aspects often assessed include the model's predictive accuracy, performance on validation datasets, potential bias or ethical issues, and compliance with data privacy laws. The objective of these procedures is to establish a controlled and informed approach to deploying ML models, minimizing business risks, and ensuring responsible use of AI technologies. One example is OpenAI's Preparedness Framework.

- **Evaluation Criteria**: Track the percentage of AI models approved for deployment and the adherence to approval guidelines.
- **RACI Model**:
  - Responsible: Project Management Team
  - Accountable: Chief AI Officer
  - Consulted: Business Unit Leaders
  - Informed: All AI Stakeholders
- **High-Level Implementation Strategy**: Establish clear procedures for approving AI models, involving relevant stakeholders in the decision-making process.
- **Continuous Monitoring and Reporting**: Maintain records of approval processes and decisions, with mechanisms for regular review.
- **Access Control Mapping**: Control access to approval documentation and decision-making tools.
- **Foundational Guardrails**: Follow best practices, such as NIST SSDF and CSA CCM.

2.5.3. Model Monitoring Requirements

This refers to the ongoing process of tracking and evaluating ML models' performance after being deployed into production. This monitoring is crucial to ensure that the models continue to operate as expected over time and under varying conditions. Key aspects of model monitoring include tracking the model's predictive accuracy, detecting any drift in the model's inputs or outputs (data drift or concept drift), monitoring for any signs of bias or unfairness in predictions, and keeping an eye on the overall health and performance of the ML system. Additionally, monitoring should include mechanisms for alerting relevant stakeholders about significant changes or anomalies in the model's performance. This continuous oversight helps promptly identify and address issues such as model degradation, shifts in underlying data patterns, or emerging biases, ensuring the ML models remain effective, fair, and aligned with their intended purpose.

- **Evaluation Criteria**: Evaluate models based on the frequency and depth of monitoring activities.
- **RACI Model**:
  - Responsible: AI Operations Team
  - Accountable: Head of AI Operations
  - Consulted: Quality Assurance Team
  - Informed: Business Analysts
- **High-Level Implementation Strategy**: Implement a comprehensive model monitoring system that tracks performance, accuracy, and compliance. Recent ML monitoring technologies include features like data and concept drift detection, model performance tracking, feature attribution analysis, bias detection, and real-time alerting. These capabilities are exemplified by offerings such as SageMaker Model Monitor, which captures real-time inference data and compares it to a

baseline; Google Cloud AI Platform Prediction Monitoring, which provides insights into model predictions and data drift; and various AI monitoring and explainability platforms that enable teams to monitor, explain, and analyze ML models in production, detecting issues like data drift, model drift, and bias.

- **Continuous Monitoring and Reporting**: Set up systems for continuous data collection and analysis, with alerts for performance dips or anomalies.
- **Access Control Mapping**: Restrict access to monitoring tools and sensitive data.
- **Foundational Guardrails**: Utilize guidelines from NIST 800-53 for monitoring protocols.

2.5.4. New Model Verification Processes

The processes involve systematic procedures to rigorously test and validate new ML models before their deployment into production. These processes are designed to ensure that the models meet predefined criteria for accuracy, reliability, and fairness, and are free from defects or biases that could lead to incorrect or unfair outcomes. Verification typically includes extensive testing against diverse data sets to evaluate the model's performance and generalizability, examination for potential biases or ethical issues, and assessment of the model's robustness against adversarial attacks or data anomalies. Additionally, the verification process often involves a review of the model's documentation and development practices to ensure compliance with industry standards and best practices. These verification processes aim to establish confidence in the new models' capabilities and readiness for deployment, ensuring they function as intended and deliver value in line with business objectives and ethical guidelines.

- **Evaluation Criteria**: Measure the thoroughness of the verification process by the percentage of models undergoing complete verification.
- **RACI Model**:
    - Responsible: AI Development Team
    - Accountable: Chief Data Officer or Chief Technology Officer
    - Consulted: IT Security Team
    - Informed: Executive Management
- **High-Level Implementation Strategy**: Develop a rigorous process for verifying new models, including testing for accuracy, bias, and security vulnerabilities.
- **Continuous Monitoring and Reporting**: Establish a protocol for ongoing assessment of new models post-deployment.
- **Access Control Mapping**: Ensure strict control over who can approve and deploy new models.
- **Foundational Guardrails**: Align with NIST AI RMF for verification best practices.

2.6 Secure Model Deployment

This involves a range of practices to ensure the deployment process is safe, controlled, and aligned with organizational standards. Key areas include deployment authorization procedures, canary releases, blue-green deployments, rollback capabilities, and decommissioning models.

2.6.1. Canary Releases

Canary Releases refers to a technique used to minimize the risk of introducing a new ML model into production by gradually rolling it out to a small subset of users before deploying it more broadly. This approach allows teams to test and monitor the model's performance in a real-world environment with actual data and user interactions but on a limited scale.

- **Evaluation Criteria**: Monitor the effectiveness of canary releases by the success rate and the detection of issues in these initial deployments.
- **RACI Model**:
  - Responsible: DevOps Team
  - Accountable: Head of AI Operations or Chief Technology Officer
  - Consulted: Quality Assurance (QA) Team
  - Informed: Product Management Team
- **High-Level Implementation Strategy**: Implement canary releases as a step in the deployment process to test models gradually in a real-world environment.
- **Continuous Monitoring and Reporting**: Set up real-time monitoring for canary releases to quickly identify and address issues.
- **Access Control Mapping**: Ensure that only designated team members can initiate and monitor canary releases.
- **Foundational Guardrails**: Follow deployment guidelines as per NIST SSDF and NIST 800-53.

2.6.2. Blue-Green Deployments

A blue-green deployment is a strategy in software deployment, including machine learning (ML) models, that reduces downtime and risk by running two identical production environments, known as "blue" and "green." This approach is particularly useful in ML where deploying new models can have significant impacts on application performance and user experience.

- **Evaluation Criteria**: Evaluate the deployment strategy by the smoothness of transitions and the downtime during deployments.
- **RACI Model**:
  - Responsible: IT Operations Team
  - Accountable: Chief Technology Officer (CTO)
  - Consulted: DevOps Team
  - Informed: End Users
- **High-Level Implementation Strategy**: Adopt blue-green deployment strategies to reduce downtime and risks associated with deploying new models.
- **Continuous Monitoring and Reporting**: Continuously monitor blue and green environments for performance and issue resolution.
- **Access Control Mapping**: Manage access controls to both environments, ensuring security and integrity.
- **Foundational Guardrails**: Utilize best practices from relevant NIST guidelines.

2.6.4. Rollback Capabilities

Rollback capabilities in the context of Machine Learning (ML) models refer to the process of reverting to a previous version of an ML model or checkpoints in a production environment when a newly deployed model exhibits unexpected behavior, poor performance, or causes other issues. This is a critical aspect of deployment strategy, ensuring system stability and performance can be maintained even when new models fail to meet expectations.

- **Evaluation Criteria**: Measure the effectiveness by the speed and success rate of rollbacks when required.
- **RACI Model**:
    - Responsible: Deployment Team (DevOps team)
    - Accountable: Head of AI Operations or Chief Technology Officer
    - Consulted: IT Support Team
    - Informed: Business Stakeholders
- **High-Level Implementation Strategy**: Ensure the deployment process includes efficient rollback capabilities to revert to previous versions if necessary.
- **Continuous Monitoring and Reporting**: Monitor deployments to rapidly detect issues requiring rollbacks.
- **Access Control Mapping**: Control access to rollback tools and procedures.
- **Foundational Guardrails**: Align with CSA CCM and other security frameworks.

2.6.5. Decommissioning Models

Decommissioning ML models refers to the process of safely and systematically removing machine learning models from active production environments. This process is essential when models become obsolete, are replaced by more advanced versions, or no longer meet the evolving business requirements or compliance standards. Decommissioning is a critical step in the lifecycle management of ML models to ensure that outdated models do not risk the system's integrity or efficiency.

- **Evaluation Criteria**: Assess the process by the percentage of decommissioned models handled correctly and the adherence to decommissioning protocols.
- **RACI Model**:
    - Responsible: AI Maintenance Team (DevOps)
    - Accountable: Data Governance Officer or Chief Technology Officer)
    - Consulted: Legal and Compliance Teams
    - Informed: AI Development Teams
- **High-Level Implementation Strategy**: Develop clear procedures for safely decommissioning outdated or redundant AI models, ensuring data is handled securely.
- **Continuous Monitoring and Reporting**: Implement monitoring to ensure decommissioning processes are followed correctly.
- **Access Control Mapping**: Restrict access to decommissioning tools and data.
- **Foundational Guardrails**: Follow decommissioning practices as per NIST AI RMF and other relevant guidelines.

3. Vulnerability Management

AI Vulnerability Management is a critical component in safeguarding AI and ML systems, ensuring they remain secure, functional, and compliant. This section discusses key items in this category.

3.1. AI/ML Asset Inventory

AI/ML Asset Inventory systematically records and updates all assets within the AI/ML landscape. This includes not just the models and datasets but also the APIs, algorithms, libraries, and any supporting software or tools used in creating, training, and deploying AI/ML components of the software supply chain. The inventory provides a clear view of the resources at play, which is crucial for identifying potential vulnerabilities and managing risks effectively. The Asset Inventory used can be in the format of a model card, data card, and model registry, depending on ML systems. Understanding what assets exist and how they are interconnected is crucial for identifying potential vulnerabilities.

- **Evaluation Criteria**: The efficacy of an AI/ML Asset Inventory is measured by its comprehensiveness and the regularity of its updates. A comprehensive inventory covers every aspect of the AI/ML environment, leaving no component unaccounted for. The frequency of updates is equally important, ensuring that the inventory reflects the current state of the AI/ML ecosystem, including any new models developed, datasets acquired, or changes in the software environment.
- **RACI Model**:
    - Responsible: IT Operations Team is tasked with the day-to-day management and updating of the inventory. (Or DevOps)
    - Accountable: The Chief Information Officer (CIO) or Chief Technology Officer) oversees the process, ensuring that the inventory is maintained accurately and is used effectively in vulnerability management.
    - Consulted: AI/ML Development Teams provide insights and information necessary for keeping the inventory up-to-date and relevant.
    - Informed: Executive Management is kept in the loop about the status and health of AI/ML assets, enabling informed decision-making at higher levels.
- **High-Level Implementation Strategy**: A regular schedule should be established for reviewing and updating the AI/ML Asset Inventory. This can be facilitated by automated tools that track changes in the AI/ML environment and alert the responsible team to update the inventory. The process should also involve periodic audits to ensure accuracy and completeness.
- **Continuous Monitoring and Reporting**: Implementing real-time monitoring systems helps quickly identify changes in the AI/ML assets. This could include new model deployments, updates to existing models, datasets changes, or software environment alterations. Continuous monitoring aids in maintaining an up-to-date inventory, which is vital for effective vulnerability management.
- **Access Control Mapping**: Restricting access to the AI/ML Asset Inventory is crucial to maintain its integrity and confidentiality. Access should be limited to authorized personnel, with different levels of access for different roles, depending on their need to interact with the inventory.

- **Foundational Guardrails**: Adherence to frameworks such as the NIST AI RMF (Risk Management Framework) ensures that the AI/ML Asset Inventory is managed to align with industry best practices and regulatory requirements. These frameworks provide guidelines on effectively cataloging and managing AI/ML assets, contributing to a robust vulnerability management strategy.

3.2. Continuous Vulnerability Scanning

This refers to the systematic and ongoing examination of all AI/ML assets to identify security weaknesses. This includes scanning models, datasets, associated infrastructure (outdated libraries or insecure APIs), and any other components in the AI/ML environment. The objective is to detect vulnerabilities that could be exploited, thereby preemptively addressing potential security issues.

- **Evaluation Criteria**: The effectiveness of this scanning process is measured by two primary metrics: the percentage of AI/ML assets scanned and the frequency of these scans. An optimal vulnerability scanning program ensures that no component is left unchecked and that scans are conducted regularly to catch new vulnerabilities that may arise due to updates or changes in the environment.
- **RACI Model**:
  - Responsible: The Security Operations Team carries out the scanning process, utilizing tools and technologies to conduct thorough assessments.
  - Accountable: The Chief Information Security Officer (CISO) oversees the entire process, ensuring that scans are conducted effectively and vulnerabilities are addressed promptly.
  - Consulted: AI/ML Teams provide insights into the specific requirements and configurations of the AI/ML assets, aiding in more targeted scanning.
  - Informed: IT Management is updated on the scanning results and any critical vulnerabilities affecting the broader IT infrastructure.
- **High-Level Implementation Strategy**: Implementing automated scanning tools is essential for efficient and effective vulnerability scanning. These tools should be configured to regularly scan all AI/ML assets and update as new threats emerge. Scheduling regular assessments ensures that the AI/ML environment remains secure over time.
- **Continuous Monitoring and Reporting**: Establishing an alert system is crucial for immediately identifying new vulnerabilities. This system should notify relevant teams about detected weaknesses, enabling quick response and remediation.
- **Access Control Mapping**: Access to the vulnerability scanning tools and results should be tightly controlled. Only authorized personnel should be able to conduct scans and access the detailed results, ensuring the security of sensitive information revealed during the scans.
- **Foundational Guardrails**: Adhering to established security standards such as NIST 800-53 ensures that the vulnerability scanning process aligns with industry best practices. These standards provide guidelines on effectively identifying and addressing vulnerabilities, enhancing the overall security posture of the AI/ML systems.

3.3. Risk-Based Prioritization

This involves assessing and ranking vulnerabilities found in AI/ML assets based on their potential impact and likelihood of exploitation. This process helps organizations focus their resources and efforts on mitigating the most critical vulnerabilities first, thereby efficiently reducing the overall risk to their AI/ML systems.

Evaluation Criteria: The effectiveness of this approach is gauged by the proportion of high-risk vulnerabilities that have been successfully addressed compared to the total number of identified vulnerabilities. A high percentage indicates effective prioritization and remediation of the most severe risks.

- **RACI Model**:
    - Responsible: The Risk Management Team is tasked with evaluating and prioritizing vulnerabilities based on their risk level.
    - Accountable: The Chief Information Security Officer (CISO) oversees the process, ensuring that the most critical vulnerabilities are identified and addressed promptly.
    - Consulted: The Compliance Team provides input, especially regarding regulatory and compliance aspects of the vulnerabilities.
    - Informed: AI/ML Development Teams are informed about the prioritization and status of vulnerabilities affecting their assets.

High-Level Implementation Strategy: Developing a comprehensive risk assessment framework is crucial. This framework should include criteria for evaluating the severity of vulnerabilities, such as the potential impact on confidentiality, integrity, availability, and the likelihood of exploitation.
Continuous Monitoring and Reporting: Implementing a system for continuously monitoring vulnerabilities and their risk levels is essential. Regular updates and reports on the risk status of vulnerabilities ensure that all stakeholders are aware of the current threat landscape and the progress of remediation efforts.
Access Control Mapping: Access to risk assessment tools and vulnerability data should be tightly controlled. Only authorized personnel should be able to assess, categorize, and prioritize vulnerabilities to maintain the integrity of the process.
Foundational Guardrails: Adherence to guidelines such as the NIST AI Risk Management Framework (RMF) ensures that the process aligns with industry best practices and provides a structured approach to managing AI/ML systems risks.

Risk-based prioritization is a vital component in AI vulnerability management, enabling organizations to allocate resources efficiently to mitigate the most pressing security risks in their AI/ML systems.

3.4. Remediation Tracking

This involves the continuous monitoring and management of the process to address and rectify identified vulnerabilities within AI/ML systems. It encompasses the tracking of actions taken to mitigate vulnerabilities, ensuring they are resolved promptly.

- **Evaluation Criteria**: The effectiveness of Remediation Tracking is assessed based on two primary metrics: the time taken to remediate vulnerabilities and the percentage of resolved issues. A shorter time to remediate and a higher percentage of resolved vulnerabilities indicate efficient and successful remediation efforts. Time taken to remediate vulnerabilities must be tracked against organizational vulnerability remediation SLAs based on above defined risk-based vulnerability prioritization.
- **RACI Model**:
    - Responsible: The IT Operations Team executes the actions required to remediate vulnerabilities.
    - Accountable: The Chief Information Security Officer (CISO) holds overall accountability for the effectiveness of the remediation process, ensuring that vulnerabilities are addressed promptly.
    - Consulted: AI/ML Development Teams provide input and assistance in understanding the specific requirements for remediating vulnerabilities related to AI/ML assets.
    - Informed: Executive Leadership is kept informed about the status of vulnerability remediation efforts and their potential impact on the organization.
- **High-Level Implementation Strategy**: Implementing robust tracking systems is essential for efficient Remediation Tracking. These systems should capture details of identified vulnerabilities, actions taken for remediation, responsible parties, and timelines for resolution.
- **Continuous Monitoring and Reporting**: Maintaining detailed records of remediation activities, including progress updates and completion status, is critical. Continuous monitoring ensures that vulnerabilities are actively tracked and managed until they are successfully resolved.
- **Access Control Mapping**: Access to documentation related to remediation activities should be securely controlled to prevent unauthorized access or tampering. This safeguards the integrity of the remediation process.
- **Foundational Guardrails**: Reference to established cybersecurity standards such as NIST 800-53 ensures that the Remediation Tracking process aligns with industry best practices and provides a structured approach to managing and documenting vulnerability remediation.

3.5. Exception Handling

Exception Handling is the process of effectively managing situations where deviations or exceptions occur from established security protocols and procedures. These exceptions may arise due to unique circumstances, operational needs, legacy systems, or other factors that require a deviation from standard security practices.

- **Evaluation Criteria**: Exception Handling effectiveness is assessed based on the number of exceptions handled and the resolution's overall effectiveness. A well-managed exception-handling process should minimize the number of exceptions and ensure that when they do occur, they are addressed appropriately with security concerns addressed using compensating controls.
- **RACI Model**:
    - Responsible: The Security Team manages, and addresses security exceptions as they arise.

  - Accountable: The Chief Information Security Officer (CISO) holds overall accountability for the effectiveness of the exception-handling process, ensuring that exceptions are managed in alignment with security policies and regulations.
  - Consulted: Legal and Compliance Teams provide guidance and advice to ensure that exceptions are managed within the bounds of legal and regulatory requirements.
  - Informed: Management is informed about the exceptions and their resolution, ensuring transparency and alignment with overall operations.
- **High-Level Implementation Strategy**: Establishing clear and documented protocols for handling security exceptions is essential. These protocols should define the process for identifying, assessing, and addressing exceptions while ensuring that security remains a top priority.
- **Continuous Monitoring and Reporting**: Exception handling should be well-documented, including the circumstances of the exception, the actions taken to address it, and any remediation measures put in place. Continuous monitoring and reporting help ensure that exceptions are managed effectively over time.
- **Access Control Mapping**: Access to the exception-handling process and related documentation should be restricted to authorized personnel only. This ensures that the handling of exceptions remains secure and compliant with established protocols.
- **Foundational Guardrails**: Ensuring alignment with industry standards such as the Cloud Security Alliance (CSA) Cloud Controls Matrix (CCM) helps establish best practices for handling exceptions and ensures that exceptions are managed in accordance with recognized guidelines.

Exception Handling is a critical component of AI Vulnerability Management, allowing organizations to respond effectively to unique circumstances while maintaining overall security and compliance standards.

3.6. Reporting Metrics

This refers to the specific measurements and key performance indicators (KPIs) used to assess and quantify the effectiveness of AI vulnerability management efforts. These metrics provide valuable insights into the state of security within AI/ML systems.

- **Evaluation Criteria**: The accuracy and timeliness of reporting are essential for assessing Reporting Metrics' effectiveness. Accurate and up-to-date reporting ensures that decision-makers have access to reliable information for making informed security decisions.
- **RACI Model**:
  - Responsible: The Reporting Team is responsible for gathering, analyzing, and presenting the vulnerability management metrics.
  - Accountable: The Chief Information Security Officer (CISO) is accountable for the overall effectiveness of the reporting process and ensuring that the metrics align with security objectives.
  - Consulted: AI/ML and IT Departments provide input and context to ensure the metrics accurately reflect the AI/ML environment's security posture.
  - Informed: Executive Management is informed about the results and insights derived from the reporting metrics, enabling strategic decision-making.

- **High-Level Implementation Strategy**: Developing standardized reporting procedures is crucial. These procedures should define how metrics are collected, analyzed, and presented to ensure consistency and accuracy.
- **Continuous Monitoring and Reporting**: Regular updates and distribution of reports are essential to keep all stakeholders informed about the current state of AI vulnerability management. This continuous reporting ensures that security issues are identified and addressed promptly.
- **Access Control Mapping**: Access to reporting tools and data should be controlled to prevent unauthorized access or tampering with metrics. Restricting access safeguards the integrity of the reporting process.
- **Foundational Guardrails**: Following best practices from recognized frameworks such as the National Institute of Standards and Technology (NIST) Secure Software Development Framework (SSDF) and the Cloud Security Alliance (CSA) Cloud Controls Matrix (CCM) helps establish industry-standard guidelines for developing and managing reporting metrics effectively.

Conclusion

This white paper has explored the core security responsibilities that organizations must uphold when developing and deploying AI and ML systems. By focusing on data security, model security, and vulnerability management, we have outlined a comprehensive framework for ensuring the security, privacy, and integrity of AI systems throughout their lifecycle.

In the realm of data security and privacy, we emphasized the importance of data authenticity, anonymization, pseudonymization, data minimization, access control, and secure storage and transmission. These measures are critical for protecting sensitive information and maintaining compliance with data protection regulations.

Regarding model security, we discussed the significance of access controls, secure runtime environments, vulnerability and patch management, MLOps pipeline security, AI model governance, and secure model deployment. By implementing robust security controls and governance processes, organizations can mitigate risks associated with AI models and ensure their reliable and trustworthy operation.

Vulnerability management is another crucial aspect of AI security. We highlighted the need for maintaining an AI/ML asset inventory, conducting continuous vulnerability scanning, prioritizing risks, tracking remediation efforts, handling exceptions, and establishing reporting metrics. These practices enable organizations to proactively identify and address vulnerabilities, minimizing the potential for security breaches and ensuring the ongoing security of AI systems.

Throughout the white paper, we analyzed each responsibility using quantifiable evaluation criteria, the RACI model for role definitions, high-level implementation strategies, continuous monitoring and reporting mechanisms, access control mapping, and adherence to foundational guardrails based on industry best practices and standards such as NIST AI RMF, NIST SSDF, NIST 800-53, CSA CCM, and others.

By adopting the recommendations and best practices outlined in this white paper, organizations can establish a strong foundation for secure and responsible AI development and deployment. However, it is essential to recognize that AI security is an ongoing process that requires continuous monitoring, adaptation, and improvement as technologies and threats evolve.

As organizations navigate the complexities of AI adoption, it is crucial to foster a culture of security and collaboration among all stakeholders, including management, technical teams, governance bodies, and end-users. By working together and adhering to the principles and practices discussed in this white paper, organizations can unlock the transformative potential of AI while ensuring the security, privacy, and trust of all stakeholders involved.

Acronyms

AI	Artificial Intelligence
AI RMF	Artificial Intelligence Risk Management Framework
AIMS	AI Management System
AIOps	Artificial Intelligence for IT Operations
API	Application Programming Interface
AWS	Amazon Web Services
CAIO	Chief AI Officer
CCM	Cloud Controls Matrix
CDO	Chief Data Officer
CEO	Chief Executive Officer
CFO	Chief Financial Officer
CI/CD	Continuous Integration/Continuous Deployment
CIO	Chief Information Officer
CIS	Center for Internet Security
CISO	Chief Information Security Officer
CNAPP	Cloud Native Application Protection Platform
COO	Chief Operating Officer
CPO	Chief Privacy Officer
CSA	Cloud Security Alliance
CTO	Chief Technology Officer
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
DAST	Dynamic Application Security Testing
DataOps	Data Operations
DDoS	Distributed Denial of Service
DevSecOps	Development, Security, and Operations
DISA	Defense Information Systems Agency
DoS	Denial of Service
ENISA	European Union Agency for Cybersecurity
GDPR	General Data Protection Regulation
HSM	Hardware Security Module
IaC	Infrastructure as Code
IaaS	Infrastructure as a Service
IAM	Identity and Access Management
IDPS	Intrusion Detection and Prevention System
IDS	Intrusion Detection System
IEC	International Electrotechnical Commission
IPS	Intrusion Prevention System
ISM	Information Security Manager
ISMS	Information Security Management System
ISO	International Organization for Standardization
ISS	Information Systems Security
ISSO	Information Systems Security Officer

K8s	Kubernetes
KPI	Key Performance Indicator
LLM	Large Language Model
MFA	Multifactor Authentication
MITRE ATT&CK	MITRE Adversarial Tactics, Techniques, and Common Knowledge
ML	Machine Learning
MLOps	Machine Learning Operations
NIST	National Institute of Standards and Technology
OS	Operating System
OWASP	Open Web Application Security Project
PaaS	Platform as a Service
PIMS	Privacy Information Management System
PoLP	Principle of Least Privilege
QA	Quality Assurance
RACI	Responsible, Accountable, Consulted, Informed
RBAC	Role-Based Access Control
SaaS	Software as a Service
SAST	Static Application Security Testing
SDLC	Software Development Life Cycle
SLA	Service Level Agreement
SSDF	Secure Software Development Framework
STIGs	Security Technical Implementation Guides
TEE	Trusted Execution Environment
TLS	Transport Layer Security
VPN	Virtual Private Network
WAF	Web Application Firewall