Implementation Guide for the AI Cyber Security Code of Practice

The Department for Science, Innovation and Technology (DSIT) commissioned John Sotiropoulos, Senior Security Architect at Kainos, to create this implementation guide. Each iteration was reviewed by DSIT and National Cyber Security Centre officials.

Introduction

The growing deployment and technological advancements of Artificial Intelligence ("AI") has further reiterated the need for tailored security requirements for AI systems. The UK Government's voluntary Code of Practice provides baseline cyber security provisions for various types of AI systems. It will be used as the basis for the development of a global standard in ETSI (Technical Specification ("TS") 104 223). This document will guide stakeholders across the AI supply chain on the Code's implementation by providing non-exhaustive scenarios as well as examples of practical solutions to meet these provisions. It is possible to meet the provisions in the UK Government's Code of Practice by using other solutions not set out in this document. This document will also be submitted and used in ETSI to develop the Technical Specification's supporting implementation guide.

1. Scope

This document serves as guidance to help stakeholders across the supply chain for AI systems, particularly Developers and System Operators, to meet the cyber security provisions outlined for AI systems in the UK Government's Code of Practice (and subsequently ETSI TS 104 223). These stakeholders could include a diverse range of entities, including large enterprises and government departments, independent developers, small and medium enterprises (SMEs), charities, local authorities and other non-profit organisations. The document will also be useful for stakeholders planning to purchase AI services. Additionally, this guide has been designed to support the future development of AI cyber security standards, including specifications that could inform future assurance and certification programmes. Where relevant, this document signposts supporting specifications and international frameworks.

References

The following referenced documents can further support the application of subject areas covered in this document. This is not an exhaustive list; other documents have also been referenced in this document to assist stakeholders.

- ETSI TR 104 222 - Securing Artificial Intelligence; Mitigation Strategy Report: Available at https://www.etsi.org/deliver/etsi_tr/104200_104299/104222/01.02.01_60/tr_104222v010201p.pdf
- ETSI GR SAI 002 - Securing Artificial Intelligence (SAI); Data Supply Chain Security. Available at: https://www.etsi.org/deliver/etsi_gr/SAI/001_099/002/01.01.01_60/gr_SAI002v010101p.pdf

- ETSI TR SAI-004 - Securing Artificial Intelligence (SAI); Traceability of AI Models: Available at https://www.etsi.org/deliver/etsi_tr/104000_104099/104032/01.01.01_60/tr_104032v010101p.pdf
- EU AI Act - Regulation (EU) 2024/1689: Available at: https://eur-lex.europa.eu/eli/reg/2024/1689/oj ISO/IEC 22989 – Artificial intelligence concepts and terminology: Available at https://www.iso.org/standard/74296.html
- ICO – Artificial Intelligence: Available At https://ico.org.uk/for-organisations/advice-and-services/audits/data-protection-audit-framework/toolkits/artificial-intelligence/
- NCSC Machine Learning Principles: Available at https://www.ncsc.gov.uk/collection/machine-learning-principles
- NCSC Guidelines for Secure AI System Development: Available at https://www.ncsc.gov.uk/collection/guidelines-secure-ai-system-development
- CISA Joint Cybersecurity Information - Deploying AI Systems Securely: Available at https://media.defense.gov/2024/Apr/15/2003439257/-1/-1/0/CSI-DEPLOYING-AI-SYSTEMS-SECURELY.PDF
- Cyber Security Agency of Singapore - Guidelines on Securing AI Systems: Available at https://www.csa.gov.sg/docs/default-source/publications/2024/guidelines-on-securing-ai-systems_2024-10-15.pdf
- Cyber Security Agency of Singapore - Companion Guide on Securing AI Systems: Available at https://www.csa.gov.sg/docs/default-source/publications/2024/companion-guide-on-securing-ai-systems_2024-10-15.pdf
- MITRE ATLAS Framework: Available at https://atlas.mitre.org/
- NIST Adversarial Machine Learning Taxonomy: Available at https://csrc.nist.gov/pubs/ai/100/2/e2023/final
- OWASP AI Exchange: Available at: https://owaspai.org/
- OWASP Top 10 for LLM Applications: Available at: https://genai.owasp.org/
- OWASP Machine Learning Security Top Ten: Available at https://owasp.org/www-project-machine-learning-security-top-10/

When developing the voluntary Code of Practice, we also consulted with the ICO to provide consistency with ICO guidance relevant to compliance with data protection law, where applicable. Various ICO guidance can also be found in the Code of Practice.

Definition of terms, symbols and abbreviations

For the purposes of this document, the terms given in the UK Government's Code of Practice apply. There are no symbols used in this document. A list of abbreviations is provided below.

Abbreviations

ADR: Architecture Decision Records
AI: Artificial Intelligence
API: Application Programming Interface
BOM: Bill of Materials
CEN/CLC: European Committee for Standardization and European Committee for Electrotechnical Standardization
CI/CD: Continuous Integration/Continuous Deployment.
CISA: Cyber Security and Infrastructure Agency

DPIA: Data Protection Impact Assessment
DPT: Data Protection Toolkit
ETSI: European Telecommunications Standards Institute
GDPR: General Data Protection Regulation
GRC: Governance Risk and Compliance
ICO: Information Commissioner's Office
ISO/IEC: International Organization for Standardization / International Electrotechnical Commission
LLM: Large Language Models
MFA: Multi-Factor Authentication
ML: Machine Learning
ML BOM: Machine Learning Bill of Materials
MLOps: Machine Learning Operations
MITRE: MITRE Corporation
NCSC: National Cyber Security Centre
NIST: National Institute of Standards and Technology
NLP: Natural Language Processing
OWASP: Open Web Application Security Project
RAG: Retrieval-Augmented Generation
RBAC: Role-Based Access Control
RL: Reinforcement Learning
RLFAI: Reinforcement Learning from AI
RLHF: Reinforcement Learning from Human Feedback
RSS: Really Simple Syndication
SaaS: Software as a Service
SBOM: Software Bill of Materials
SHA-256: Secure Hash Algorithm 256-bit
SLAs: Service Level Agreements
T&Cs: Terms and Conditions
WCAG: Web Content Accessibility Guidelines
WORM: Write Once, Read Many

Terms Used

Adversarial AI: Describes techniques and methods that exploit vulnerabilities in the way AI systems work, for example, by introducing malicious inputs to exploit their machine learning aspect and deceive the system into producing incorrect or unintended results. These techniques are commonly used in adversarial attacks but are not a distinct type of AI system.
Adversarial Attack: An attempt to manipulate an AI model by introducing specially crafted inputs to cause the model to produce errors or unintended outcomes.

Agentic Systems: AI systems capable of initiating and executing actions autonomously, often interacting with other systems or environments to achieve their goals.
Application Programming Interface (API): A set of tools and protocols that allow different software systems to communicate and interact.
Artificial Intelligence (AI): Systems designed to perform tasks typically requiring human intelligence, such as decision-making, language understanding and pattern recognition. These systems can operate with varying levels of autonomy and adapt to their environment or data to improve performance.
Bill of Materials (BOM): A comprehensive inventory of all components used in a system, such as software dependencies, configurations, and hardware.
Data Custodian: See definition in the Code of Practice
Data Poisoning: A type of adversarial attack where malicious data is introduced into training datasets to compromise the AI system's performance or behaviour.
Data Protection Impact Assessment (DPIA): A tool used in UK GDPR to assess and mitigate privacy risks associated with processing personal data in AI systems.
Embeddings: Vector representations of data (e.g., text, images) that capture their semantic meaning in a mathematical space, commonly used to improve the efficiency of search, clustering and similarity comparisons.
Evasion Attack: A type of adversarial attack where an adversary manipulates input data to cause the AI system to produce incorrect or unexpected outputs without altering the underlying model.
Excessive Agency: A situation where an AI system has the capability to make decisions or take actions beyond its intended scope, potentially leading to unintended consequences or misuse.
Explainability: The ability of an AI system to provide human-understandable insights into its decision-making process.
Feature Selection: The process of selecting a subset of relevant features (variables) for use in model training to improve performance, reduce complexity and prevent overfitting.
Generative AI: AI models that generate new content, such as text, images or audio, based on training data. Examples include image synthesis models and large language models like chatbots.
Governance Framework: Policies and procedures established to oversee the ethical, secure and compliant use of AI systems.
Guardrails: Predefined constraints or rules implemented to control and limit an AI system's outputs and behaviours, ensuring safety, reliability, and alignment with ethical or operational guidelines.
Hallucination (in AI): AI-generated content that appears factual but is incorrect or misleading. This is prevalent in LLMs, which may produce plausible sounding but inaccurate responses.
Inference Attack: A privacy attack where an adversary retrieves sensitive information about the training data, or users, by analysing the outputs of an AI model.
Large Language Model (LLM): A type of AI model trained on vast amounts of text data to understand and generate human-like language. Examples include chatbots and content generation tools.
Machine Learning (ML): A subset of AI where systems improve their performance on a task over time by learning from data rather than following explicit instructions.
Machine Learning Bill of Materials (ML BOM): A specialised BOM for AI systems that catalogues models, datasets, parameters and training configurations used in the development and deployment of machine learning solutions.
ML Ops (Machine Learning Operations): A set of practices and tools that streamline and standardise the deployment, monitoring and maintenance of machine learning models in production environments.
Model Extraction: An attack where an adversary recreates or approximates a proprietary AI model by querying it and analysing its outputs, potentially exposing trade secrets or intellectual property.
Model Inversion: A privacy attack where an adversary infers sensitive information about the training data by analysing the AI model's outputs.
Multimodal Models: AI models that process and integrate multiple types of data (e.g., text, images, audio) to perform tasks.

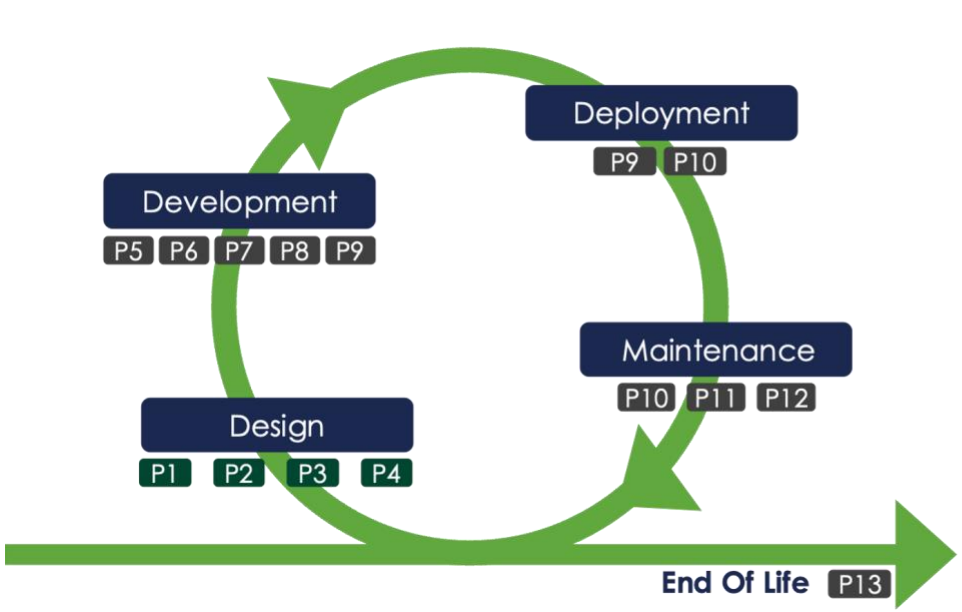
Natural Language Processing (NLP): A type of machine learning that understands, interprets, and generates human language in a way that is meaningful and useful.
Predictive (or Discriminative) AI: A type of machine learning designed to classify inputs or make predictions based on existing data. These models focus on identifying patterns and drawing distinctions, such as fraud detection or customer segmentation.
Prompt: An input provided to an AI model, often in the form of text, that directs or guides its response. Prompts can include questions, instructions, or context for the desired output.
Prompt Injection: An attacker exploits a vulnerability in AI models by using prompts that produce unintended or harmful outputs.
Retrieval-Augmented Generation (RAG): An AI approach that combines external knowledge retrieval (e.g., documents or databases) with prompts to language model generation to provide accurate and up-to-date responses.
Reinforcement Learning (RL): A machine learning approach where an agent learns by interacting with its environment and receiving feedback in the form of rewards or penalties.
Risk Assessment: The process of identifying, analysing and mitigating potential threats to the security or functionality of an AI system.
Sanitisation: The process of cleaning and validating data or inputs to remove errors, inconsistencies and malicious content, ensuring data integrity and security.
Software Bill of Materials (SBOM): A detailed list of all software components in a system, including open-source libraries, versions and licences to ensure transparency and security.
System Prompt: A predefined input or set of instructions provided to guide the behaviour of an AI model, often used to define its tone, rules, or operational context.
Threat Modelling: A process to identify and address potential security threats to a system during its design and development phases.
Training: The process of teaching an AI model to recognise patterns, make decisions, or generate outputs by exposing it to labelled data and adjusting its parameters to minimise errors.
Web Content Accessibility Guidelines. Guidelines, as part of, internationally recognised standards for making web content more accessible to people with impairments. They are developed and maintained by the **World Wide Web Consortium (W3C)** under its Web Accessibility Initiative (WAI)

2. How to use this document

Purpose

The intent of this document is to help implementers of the Code (and proposed TS 104 223) understand how each provision can be met. The Code also includes some examples to contextualise certain provisions; these have been incorporated into relevant sections of this document. Recommendations in the Code are expected to be followed by AI supply chain stakeholders unless they are not applicable because of the type of model(s) used for the AI system. Specifically, it may depend on whether a stakeholder has decided to develop their own model, use, or finetune a third-party model (either directly or remotely via an API). This document specifies where a particular provision has scope restrictions.

The image below highlights that the principles have been mapped to various phases of the AI lifecycle. Importantly, some of the principles and provisions are also relevant to other phases, which has been clarified in the relevant sections. An example is principle 9, which is included under "development", but it is also very important for the "deployment" of an AI system. The examples provided to address the below scenarios are not exhaustive or limitative; it is possible to meet the provisions in the Code by using other solutions, or variants of the examples provided.



P1	Raise awareness of AI security threats and risks
P2	Design your AI system for security as well as functionality and performance
P3	Evaluate the threats and manage the risks to your AI system
P4	Enable human responsibility for AI systems
P5	Identify, track and protect your assets
P6	Secure your infrastructure
P7	Secure your supply chain
P8	Document your data, models and prompts
P9	Conduct appropriate testing and evaluation
P10	Communication and processes associated with End-users and Affected Entities
P11	Maintain regular security updates, patches and mitigations
P12	Monitor your system's behaviour
P13	Ensure proper data and model disposal

Relationship to the UK Government's Code of Practice (and future TS 104 223)

The Code sets out a detailed list of provisions based on the above thirteen principles. This document can be used (when implemented) to inform the definition of test scenarios and the development of a test plan based on the Code.

3. Guidance on implementation

Section 4 provides examples for implementing the Code's provisions based on various scenarios (outlined below) of how a stakeholder might create and use an AI system.

- **Chatbot App**: An organisation using a publicly available LLM via the APIs offered by the external provider to develop a chatbot for internal and customer use. This may include: 1) A large enterprise uses a publicly available LLM through an API to create chatbots for internal and customer interactions, such as answering FAQs or automating routine customer service tasks. 2) A small retail business developing and using an AI-powered chatbot to handle online shopping queries, assisting customers with product recommendations and order tracking. 3) A hospital developing and using a chatbot to provide general health advice and appointment scheduling, ensuring compliance with data privacy requirements. 4) A local council develops and uses a chatbot to provide guidance on local planning applications and handle the applications.

- **ML Fraud Detection**: A mid-size software company selects an open-access classification model, which they train further with additional datasets to develop and host a fraud detection system. The system is designed to identify patterns of fraudulent financial transactions based solely on transactional data. It explicitly avoids linking decisions to inferred personal characteristics, behaviours, or any factors unrelated to the context of the financial transaction. The model's primary focus is on identifying fraud patterns and does not evaluate or classify individuals' social behaviour or implement social scoring. The scenario does not encompass situations that are detrimental or unfavourable treatment of certain natural persons or groups of persons that is unjustified or disproportionate to their social behaviour or its gravity, as stipulated in Article 5(1)(c) Prohibited Practices of the EU AI Act (Regulation (EU) 2024/1689).
- **LLM Provider**: A tech company develops a new multimodal LLM capable of understanding and generating text, audio and images, providing commercial API access to developers for diverse applications, such as virtual assistants and media generation.
- **Open-Access LLM.** A small organisation is developing an LLM for specific use cases. This may include 1) developing an LLM for legal and contract negotiation use cases planning to release it as open-access and monetise via support agreements. 2) A law firm using the open-access LLM to combine it with their confidential casework for legal research, enabling quick identification of relevant legal precedents and statutes. to 3) A rural development organisation developing and using an open access LLM to offer farmers localised advice on crop management and pest control strategies.

The information in this document will help stakeholders to protect end-users and affected entities from vulnerabilities that could result in confidentially, integrity, or availability attacks. This includes the various threat-related examples linked to AI systems below:

- Data Poisoning, Backdoors, Model Tampering, Evasion and Supply-Chain Attacks
- Privacy Attacks, such as Model Theft, Model Extraction, Model Inversion and Inference Attacks
- Information Disclosure of Personal and Special Category Data, Confidential Business Information or System Configuration details.
- Prompt Injections, Excessive Agency and Training Data Extraction and Model Denial of Service

These are the most common examples of threats, but threats will continue to evolve and new ones will emerge. For complete taxonomies, refer to OWASP AI Exchange, MITRE ATLAS and the NIST Adversarial Attacks Taxonomy.

AI models and systems can also be misused. Although this area is out of scope of this document, there is some crossover as AI Security underpins all aspects of AI Safety and safeguards Responsible AI. As a result, this guide will help reduce the risk of AI models and systems being misused by third parties. The following measures/controls set out in this document could help to mitigate the misuse of AI systems, for example to help produce misinformation or conduct cyber attacks:

- Human Oversight Mechanisms
- Access Control and Rate-Based Permissions

- Threat Modelling
- Risk Assessment
- Documentation and Monitoring of Prohibited Cases.
- Monitoring and Logging
- Rate Limiting

While the primary focus of this guide is AI security, certain aspects of Responsible AI such as copyright violations, bias, unethical or harmful use and legal or reputational risks are included in specific sections of this guide. In the context of AI, these areas often stem from or are exacerbated by poor AI security practices; safeguarding them not only mitigates the misuse of AI but also strengthens trust and compliance in AI systems.

Privacy and data protection related legislation cover a very small part of the Code because the provisions are set across all phases of the AI lifecycle. Therefore, personal data, (although a consistent theme throughout the principles), is specified only in particular circumstances. Organisations also need to consult official regulatory guidance for regulatory compliance where appropriate, including data protection guidance issued by the ICO (and/or other relevant data regulatory bodies). Additionally, stakeholders that adhere to the Code will still need to ensure their compliance with other regulatory compliance requirements.

For the purposes of this guide, 'regularly' denotes a frequency determined by the associated risks and operational requirements of the system. This can range from continuous, daily, or weekly actions for high-risk scenarios to quarterly or annual actions for lower risk scenarios.

This document has not included content on how stakeholders can verify conformity to each provision, as we recognise that standards bodies have a distinct process for creating a conformity assessment specification that sits alongside a standard and implementation guide. We also recognise that the assurance and certification sector has a key role to play in this area.

Other Related Standards

This guide aligns with international standardisation including:

i) Approved standards and reports:

- ETSI GR SAI 002 - Securing Artificial Intelligence (SAI); Data Supply Chain Security.
- ETSI GR SAI 007- Securing Artificial Intelligence (SAI); Explicability and transparency of AI processing
- ETSI TR 104 222 - Securing Artificial Intelligence (SAI); Mitigation Strategy Report
- ETSI TR 104 032 - Securing Artificial Intelligence (SAI); Traceability of AI Models
- ETSI TR 104 225 - Securing Artificial Intelligence TC (SAI); Privacy aspects of AI/ML systems
- ETSI TR 104 066 - Securing Artificial Intelligence; Security Testing of AI
- ISO/IEC 22989:2022, Information technology - Artificial intelligence - Artificial intelligence concepts and terminology
- ISO/IEC 42001:2023, Information technology - Artificial intelligence - Management system
- ISO/IEC 25059:2023, Software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Quality model for AI system)

ii) Standardisation work in progress:

- ETSI TS 104 050 - Securing Artificial Intelligence (SAI); AI Threat Ontology and definitions
- ISO/IEC DIS 12792, Transparency taxonomy of AI systems
- ISO/IEC JTC 1/SC 42/WG 4, AI system life cycle processes
- ISO/IEC DIS 27090, Guidance for addressing security threats to artificial intelligence systems
- CEN/CLC/JTC prENXXX (WI=JT021024), AI Risk Management
- CEN/CLC/JTC prEN XXX (WI=JT021008), AI trustworthiness framework

4. Examples for implementing the Code of Practice

Principle 1: Raise awareness of AI security threats and risks			
Provisions	**Related threats / risks**	**Example Measures/Controls**	**Reference/Resource**
1.1 Organisations' cyber security training programme **shall** include AI security content which **shall** be regularly reviewed and updated where necessary, such as if new substantial AI-related security threats emerge.	Staff may be unaware of unique AI vulnerabilities like data poisoning, adversarial attacks, or prompt injections, leaving the system exposed to sophisticated attacks. These attack types are still being understood and evolve in areas such as generative AI and so training must keep up to date as knowledge evolves.	***Establish an AI Security Awareness Training Programme*** *that covers basic AI concepts, threats, applicable regulations, etc. You should include guidance on how to monitor for threats and the escalation paths for reporting security concerns.* **1. Chatbot App**: Training on AI concepts, personal data and its regulatory implications, risks on confidential business information or system configuration, hallucinations, overreliance, and ethical use and safety; training should cover the at least the ICO and NCSC guidelines, and OWASP Top 10 for LLM applications. **2. ML Fraud Detection**: Provide training on AI concepts and use ICO and NCSC guidelines and the OWASP AI Exchange, to cover ML threats such as poisoning, evasion, model extraction, model inversion, inference, and supply-chain attacks. **3. LLM Platform**: provide training on AI concepts and threats including poisoning, prompt injections, safety and data protection and ICO, NCSC, guidelines; Cover OWASP AI Exchange, the OWASP Top 10 for LLM applications and recent reports on the risks of general-purpose systems. **4. Open-Access LLM Model:** Self-training on ICO Guidelines, OWASP AI Exchange The OWASP Top 10 for LLM applications and recent reports on the risks of general-purpose systems.	• ISO/IEC 22989 – Artificial intelligence concepts and terminology ICO Data Protection Audit Framework ICO: Generative AI- eight questions that developers and users need to ask NCSC Machine Learning Principles, Part 1, 1.1 Raise awareness of ML threats and risks. OWASP Top 10 for LLM applications OWASP AI Exchange
1.1.1 AI security training **shall** be tailored to the specific roles and responsibilities of staff members.	Without tailored AI security training, staff may lack the knowledge to address role-specific risks, leading to ineffective implementation of security measures, increased vulnerability to threats, and potential misuse or mismanagement of AI systems.	***Role-Specific AI Security Training:*** *Provide role-specific AI security training tailored to the responsibilities of each staff category.* **1. Chatbot App**: Train engineers on secure coding and AI-specific vulnerabilities (see 1.2.2); for CISOs. include governance frameworks, incident response strategies, and regulatory compliance, as found in ICO and NCSC guidance and the OWASP LLM Applications Cybersecurity and Governance Checklist. For Risk Officers cover relevant frameworks such as NIST RMF, AI threat modelling, and mitigation strategies; for IT Operations focus on implementing and maintaining security controls in production environments. **2. ML Fraud Detection**: Similar approach to the Chatbot App example. **3. LLM Platform**: Similar approach to the Chatbot App example. **4. Open-Access LLM Model:** Similar approach to the Chatbot App example.	• ICO: How should we assess security and data minimisation in AI? • CISA Joint Cybersecurity Information - Deploying AI Systems Securely • NCSC Guidelines for Secure AI System Development • NIST AI RMF • OWASP LLM Applications Cybersecurity and Governance Checklist
		Incorporate Training on AI Threat Modelling and Red Teaming. Provide developers and other technical staff with training on threat modelling techniques and red teaming techniques tailored for AI.	• Threat Modeling Cheat Sheet - OWASP • MITRE ATLAS

		1. Chatbot App: Training is provided to developers and Risk owners on Threat Modelling incorporating threats and mitigations from OWASP Top 10 for LLM applications. **2. ML Fraud Detection**: Follow the same approach as in the Chatbot App example but using MITRE ATLAS and OWASP AI Exchange **3. LLM Platform**: provide training on AI concepts and threats including poisoning, prompt injections, and data protection MITRE ATLAS and OWASP AI Exchange; and Generative Read-teaming (GRT) approaches including the AI Village Defcon 2024 report **4. Open-Access LLM Model:** The team should use the material in the previous example.	Generative AI: Red Teaming Challenge Transparency Report - AI Village Defcon 2024 OWASP AI Exchange OWASP Top 10 for LLM applications.
1.2 As part of an Organisation's wider staff training programme, they **shall** require all staff to maintain awareness of the latest security threats and vulnerabilities that are AI-related. Where available, this awareness **shall** include proposed mitigations.	AI systems face evolving threats, and staff who are not updated regularly on these vulnerabilities may unknowingly expose systems to risks, such as adversarial attacks or personal data leaks which will be in breach of data protection regulations.	***Maintain training awareness:*** *update training material regularly with new examples of AI threats (e.g., prompt injections, adversarial attacks) and mitigation techniques.* **1. Chatbot App:** Large organisations should conduct regular (at least annually) reviews of training material and the facility to register for changes to be updated. Smaller organisations can rely on logging new significant developments, e.g. a new version of the OWASP Top 10 for LLM Applications and include them in knowledge sharing sessions. **2. ML Fraud Detection:** Track and train staff on new adversarial attack patterns and data validation techniques or ICO guidelines as they emerge from updates in NIST, OWASP, and MITRE ATLAS taxonomies. **LLM Platform:** As in the ML Fraud Detection example. Additionally, update training with new research papers on AI vulnerabilities and share case studies on generative AI misuse and related mitigations. **Open-Access LLM Model:** Update training log with risks like data memorisation and unauthorised data use, using curated updates from ICO, OWASP, and others, including research, and utilising workshop sessions.	• MITRE ATLAS • OWASP Top 10 for LLM applications • OWASP AI Exchange NIST Adversarial ML Taxonomy
1.2.1 These updates **should** be communicated through multiple channels, such as security bulletins, newsletters, or internal knowledge-sharing platforms. This will ensure broad dissemination and understanding among the staff.	Failure to communicate new developments through diverse channels may result in uneven dissemination of critical security information, leaving some staff unaware of vulnerabilities, mitigations, or best practices, increasing the risk of oversight and security lapses.	***Disseminate Regular Security Updates and Bulletins.*** **1. Chatbot App**: This includes AI security bulletins, newsletters (ICO, NCSC, etc.), or messages on knowledge-sharing platforms and communication (messaging channels) to keep staff informed of the latest AI threats, vulnerabilities, and mitigations. Subscribe to the ICO, OWASP, and other curated newsletters and AI Feeds with a team member responsible in tracking changes. Everyone should contribute to team updates via team messaging channels. Attend conferences and events when possible and use free or low-cost resources such as RSS feeds or community forums to gain updates including new academic papers on emerging AI security vulnerabilities. Both the MITRE and OWSP slack channels are open to public and provide excellent information on AI Security news. **2. ML Fraud Detection**: As before but with an automation of curated data feeds of AI Security news and content being shared and knowledge sharing sessions **3. LLM Platform**: As before, participation in events and conferences disseminating learnings using a knowledge sharing process and platform **4. Open-Access LLM Model:** As with the chatbot app with the inclusion of automated feeds with new LLM-related research	• ICO Newsletter • NCSC News on AI • OWASP Top 10 for LLM Apps Newsletter • MITRE Slack Channel • OWASP Slack Invite

1.2.2 Organisations **shall** provide developers with training in secure coding and system design techniques specific to AI development, with a focus on preventing and mitigating security vulnerabilities in AI algorithms, models, and associated software.	Without specialized training in secure coding and AI system design, developers may inadvertently introduce vulnerabilities into AI algorithms, models, or supporting software, increasing the risk of exploits, data breaches, or system failures.	**Provide secure coding training** for engineers related to AI threats and incorporating guidelines from OWASP, NCSC, the ETSI Mitigation Strategy report. **1. Chatbot App:** Train engineers on secure coding, including implementing input validation to mitigate prompt injections. Smaller organisations can implement this control with coding standards pointing to guidelines and using code reviews and developer mentoring as training. **2. ML Fraud Detection:** Train developers on adversarial risks and OWASP AI Exchange. **3. LLM Platform:** Similar to the Fraud Detection example, but with additional system design techniques to address LLM specific safety risks (e.g., model jailbreaking). **4. Open-Access LLM Model:** Focus on secure coding techniques compliance with LLM specific threats and ICO guidelines for handling sensitive training data. Use the small organisation approach which was described in the Chatbot App section to address resource constraints.	• ETSI TR 104 222 – Securing Artificial Intelligence; Mitigation Strategy Report • NIST Secure Software Development Framework for Generative AI and for Dual Use Foundation Models Virtual Workshop • OWASP Top 10 for LLM applications • OWASP AI Exchange • OWASP Secure Coding Practices-Quick Reference Guide • NCSC Secure Development and Deployment Guidance. • NCSC Guidelines for Secure AI System Development

Principle 2: Design Your AI System for Security as well as Functionality and Performance			
Provisions	**Related threats/risks**	**Example Measures/Controls**	**Reference/Resource**
2.1 As part of deciding whether to create an AI system, a System Operator and/or Developer **shall** conduct a thorough assessment that includes determining and documenting the business requirements and/or problem they are seeking to address,	Without assessing whether an AI system is required to meet the business requirements, systems may be unnecessary or poorly suited for their environment, leading to lack of compliance, unnecessary	***Conduct Business Alignment Review:*** *Review and document business requirements for the AI system to ensure that design choices align with the organisation's needs and objectives* **1. Chatbot App:** If the chatbot is for simple summarisation or sentiment analysis, evaluate whether a task specific algorithm may be more appropriate than an LLM, which carries additional complexity and risk including regulatory risks. **2. ML Fraud Detection**: When decision-making transparency is a requirement, use a simpler model (e.g. Gradient Boosting Model) with better explainability instead of a more complex black-box Deep Learning model. **3. LLM Platform**: Review the business assessment to ensure the advanced multi-modal capabilities and complexity are required and potential risks have been considered in the business assessment.	• ICO: Do we need to consult the ICO? • ICO Data Protection Audit Framework • ICO: What is the impact of Article 22 of the UK GDPR on fairness? • OWASP Top 10 for LLM applications

along with potential AI security risks and mitigation strategies	complexity, increased attack surface, unexpected behaviour, and security vulnerabilities.	**4. Open-Access LLM Model:** When creating a specialized LLM, evaluate complexity, data protection and security risks before deciding the route to follow. Fine-tuning a pre-built LLM is faster and simpler but may involve sharing sensitive data with the provider, raising privacy concerns and risks like unauthorized access or compliance issues (e.g., UK GDPR). Building a model from scratch offers greater control but can be complex and it requires robust internal controls, such as encrypted storage and access management, to safeguard training data. Both approaches demand careful evaluation to prevent data breaches and ensure regulatory compliance.	
		Perform Risk Assessment: *Conduct and document an AI-specific risk assessment covering data classifications, logging risks of personal data and their mitigations in DPIAs. Cover expected data volume, types of integration, the model's complexity, architecture, and number of parameters. For more information on these risk factors see the NCSC Principles for Machine Learning and NCSC Guidelines for Secure AI system development.* **1. Chatbot App:** Focus the assessment on use of internal data and their classification, safety and abuse, reputational and legal risks if the chatbot provides misleading or inappropriate content. **2. ML Fraud Detection**: Use a standardised assessment template, such as the ICO's AI Data Protection Risk Toolkit**,** to record AI risk variables and risk scores. Include in other relevant factors such as interpretability and regulatory impact, then aggregate these scores to prioritise model security. Consult finance and regulatory compliance experts to understand sector-specific compliance requirements including e.g. PCI DSS, and FCA standards to ensure guidance. Ensure solution is compliant with the provisions of EU AI Act, 5(1)(c) in particular. **3. LLM Platform**: In addition to using ICO's AI Data Protection Risk Toolkit, factor in regulations, legislations and guidelines in targeted markets and cover copyright violation risks as well as emerging new risks identified in recent reports on the risks of general-purpose systems. **4. Open-Access LLM:** Follow the advice in the LLM Platform example, but include the risks related to misuse of open-source and open-access components, including malicious code, data leaks, and ethical/legal liabilities of public outputs (e.g., bias or misinformation). Review your responsibilities and legal liabilities related to licensing compliance, safeguarding sensitive data during model use or fine-tuning, and implementing safeguards to prevent misuse (e.g., phishing or misinformation campaigns). Leverage tools like the ICO's AI Data Protection Risk Toolkit, OWASP AI Security Guidelines, and MITRE ATLAS to structure assessments. your responsibilities	• EU AI Act Explorer • ICO Data Protection Audit Framework • ICO AI Data Protection Risk Toolkit • MITRE ATLAS • NCSC Risk management • International Scientific Report on the Safety of Advanced AI: Interim Report • NCSC Principles for Machine Learning • OWASP Top 10 for LLM applications
		Integrate Risk Management and Governance Frameworks: *Embed AI system assessments within both a formal Risk Management Framework (RMF) and the AI governance structure to ensure thorough risk evaluation, consistent mitigation, and monitoring before key organisational decisions.* **1. Chatbot App:** Implement NIST AI RMF and use it to assess the application as part of the framework's structured approach which includes defining purpose and risk objectives, risk	• NIST AI RMF • NIST AI RMF Playbook • NIST AI RMF Crosswalk Documents

		categorisation, risk assessment, control selection, implementation, monitoring, and review. Extend existing organisational governance with check lists and guidelines on how to review proposed AI solutions and criteria to escalate reviews to a full risk assessment for high-impact systems or models involving security, legal, compliance, and business units. **2. ML Fraud Detection**: Review NIST AI RMF to see whether it can be helpful in standardising your risk AI assessments and how to interface it with other GRC processes you have in your organisation. **3. LLM Platform**: See Chatbot App example. **4. Open-Access LLM Model:** This is not applicable to this example due to the size of the team. Instead, the team documents in their Wiki page how they perform risk assessments.	• ICO Accountability and Governance Implications for AI • European AI Alliance: Implementing AI Governance: from Framework to Practice • OWASP AI Security Centre of Excellence Guide
2.1.1 Where the Data Custodian is part of a Developers organisation, they **shall** be included in internal discussions when determining the requirements and data needs of an AI system.	Failure to include the Data Custodian in discussions about AI system requirements and data needs may result in non-compliance with data governance policies, inappropriate data usage, or insufficient safeguards for sensitive data, increasing the risk of data breaches or regulatory violations.	***Ensure collaboration with the Data Custodian:*** *during the design and development phases to define data requirements to identify regulatory compliance requirements. Ensure that Data Custodians are able to balance additional risks to data that come from the AI system with intended mitigations and the business need.* **1. Chatbot App:** Include data governance checklists as part of design discussions. Schedule workshops with Data Custodians, developers, and security staff to ensure ongoing alignment on data needs, compliance requirements, and data access implications. **2. ML Fraud Detection**: Include Data Custodian reviews and feature signoffs when using personal data; provide explanations of how data will be used, risks such as memorisation, extraction and inference, and options to safeguard use. **3. LLM Platform**: Align with internal governance processes to involve Data Custodians in defining data usage and ensuring, compliance. **4. Open-Access LLM Model:** Define who is the data custodian in your team and have them work with the rest of the team to perform and document DPIAs that are reviewed as part of the design process. Review the ICO guidelines to ensure the acting Data Custodian is performing their role in compliant manner.	• ICO's AI Data Protection Toolkit • NIST AI RMF • NIST AI RMF Playbook
2.2 Developers and System Operators **shall** ensure that AI systems are designed and implemented to withstand adversarial AI attacks, unexpected inputs and AI system failure.	Organisations may not always be successful in preventing breaches and so defence in depth requires assuming and handling some level of compromise which if undocumented will be in breach of data privacy regulation.	***Apply Secure by Design Principles****: Integrate security into the AI system's design phase by conducting threat modelling. Threat modelling covers both traditional cyber threats and AI-specific ones that might be introduced by the design choices. Incorporate standardized security controls in the system design controls to mitigate risks. Document each standardized control used in the design phase, and ensure it is integrated with specific test cases to verify its effectiveness during system testing. Ensure monitoring controls as well as incident response and recovery from failures are addressed in threat mitigation and they are documented in the system's design.* **1. Chatbot App:** Use threat modelling to identify risks specific to the chatbot app; apply controls for general application security and ones relevant to OWASP Top 10 for LLM applications, such as implementing input validation to prevent prompt injection attacks to the LLM it uses as well as preventing sensitive data exposure.	• CISA Joint Cybersecurity Information - Deploying AI Systems Securely • CSA Companion Guide on Securing AI Systems, Section 2.2.1 Planning and Design. • MITRE ATLAS • MITRE ATT&CK • NCSC Secure Design Principles

		2. ML Fraud Detection: In addition to application security controls, incorporate controls in your design for relevant predictive adversarial AI attacks such as poisoning, evasion, model extraction, and other privacy attacks. Use OWASP AI exchange as your threats and controls reference. **3. LLM Platform**: Use both MITRE ATT&CK and ATLAS to perform threat modelling of both model development and operation, addressing threats from adversarial AI attacks, especially ones related to LLMs such as poisoning and safety measures to prevent jailbreaking, data extraction, and unsafe use. Review customer-facing APIs and include them in threat modelling with usage scenarios. **4. Open-Access LLM Model:** Use a similar approach in the LLM Provider example, focusing on training but also how others might use the model and the safeguards it needs to have in place to protect them. Follow a lightweight approach to Threat Modelling as part of your design and use either MITRE ATLAS or OWASP AI Exchange as your threats and controls library.	• NCSC Principles for Machine Learning NCSC Guidelines for Secure AI system development. • OWASP AI Exchange
2.3 To support the process of preparing data, security auditing and incident response for an AI system, Developers **shall** document and create an audit trail in relation to the AI system. This **shall** include the operation, and life cycle management of models, datasets and prompts incorporated into the system.	A lack of audit trails can lead to untraceable changes or unauthorised adjustments, complicating incident response, forensic investigations, and regulatory compliance.	***Automated Audit Trails for ML Operations (MLOPs) and System changes****: Implement automated logging for all critical operations related to model training, dataset changes, prompts and parameter adjustments. For critical systems with compliance requirements, use WORM (write-once, read-many) storage to store logs, ensuring they remain tamper-proof and accessible for audits.* **1. Chatbot App:** Use a version control system to system prompt and prompt all changes with related documentation. If the model provider's API is used for fine-tuning, ensure the training and testing datasets are logged. For RAG workflows, track the embeddings generated, log metadata about retrieved data (e.g., query terms, document IDs), and maintain versioned snapshots of smaller reference datasets used in retrieval to ensure traceability and reproducibility. 2. **ML Fraud Detection:** Use an ML Ops platform or MLOps functionality in cloud platforms to enforce versioning by tracking of all changes for models, prompts, and other experimentation. Compliment MLOps with data life cycle management tools to implement similar versioning for datasets, storing details as to what data was used to train or test the system. 3. **LLM Platform:** Use the same approach described in the ML Fraud Detection example. 4. **Open-Access LLM Model:** Use open-source tools for tracking and automating workflows, such as versioning datasets, model configurations, and changes through APIs as part of your regular workflows.	• CSA Companion Guide on Securing AI Systems, Section 2.2.1 Planning and Design. • ml-ops.org : MLOps Principles • NCSC Principles for Machine Learning, Part 1, Secure Design. • NCSC Guidelines for Secure AI System Development, Secure Design. • OWASP AI Exchange
2.4 If a Developer or System Operator uses an external component they **shall** conduct an AI security risk assessment and due diligence process in line with their existing	Third-party components introduce risks through possible vulnerabilities in the external vendor's security practices which may not be to	***Security Due-Diligence for External Components****: Mandate a risk assessment process before a component (including external models) can be used, covering provenance, known risks, and when personal data is used a DPIA. Safeguard provenance by mandating in internal standards that components can only be sourced by trusted and approved sources, documenting source, version, licencing, history, and other related artifacts (e.g. Model Card for models); use checksums to verify integrity*	• ETSI GR SAI 002 - Data Supply Chain Security • NIST - Cybersecurity Supply Chain Risk Management

software development processes, that assesses AI specific risks.	the same standard as your own. This includes operating systems and libraries, container images, programming packages as well as models and datasets. Large models contain general purpose functionality, you will need to work to ensure that specific risks you care about are mitigated or otherwise managed.	**1. Chatbot App:** Review documentation including known vulnerabilities and run automated vulnerability scans against application and platform packages; consult published documentation and benchmarks or run your own against the LLM model used by the chatbot. Include embedding models in your diligence if you are using them to generate embeddings as part of RAG, instead of APIs. **2. ML Fraud Detection**: As in the Chatbot app example, but with additional diligence for the third-party base model. Consult model documentation and use tools to scan for vulnerabilities such as serialization attacks. Store approved models and components in an internal repository, ensuring they are the only ones used in production. Set up alerts for changes or security notifications for the external components. **3. LLM Platform**: Similar to Fraud Detection example but include auxiliar models that you may be using. For instance, smaller models to provide embeddings API for RAG use cases and RL models for RLHF and RLFAI in your finetuning. External datasets are of critical importance and need to be evaluated for copyright, privacy, bias, and ethical risks. **4. Open-Access LLM Model:** Automate scans for components and models you use (foundation for fine tuning, auxiliary for testing RAG, RL for RLHF and RLFAI scenarios) and ensure they are from trusted sources. Review external datasets sourced only from reputable sources and use automated tools to detect bias, personal data, and copyright issues.	• NCSC Supply Chain Security Guidance • OWASP AI Exchange • OWASP Top 10 for LLM applications - LLM03 Supply-Chain Vulnerabilities
2.5 Data Custodians **shall** ensure that the intended usage of the system is appropriate to the sensitivity of the data it was trained on as well as the controls intended to ensure the security of the data.	Misalignment between the intended usage of the AI system and the sensitivity of the data it was trained on can result in inappropriate data exposure, inadequate security controls, and regulatory non-compliance, leading to potential data breaches and misuse of personal data or other confidential information.	***Ensure Data Custodian Assurance:*** *Require Data Custodians to review system's intended usage and the data security controls to ensure compliance and balancing these risks with business needs.* **1. Chatbot App:** Review chatbot use cases with Data Custodian, access and usage policies and ensure they are aligned with the DPIA. **2. ML Fraud Detection**: As in the Chatbot app, but including data used for training, data memorisation, inversion and inference risks, and the implications of UK GDPR Article 22 for automated fraud detection. **3. LLM Platform**: Similar to the Fraud Detection example without the need for Article 22 but integrating Data Custodian review to governance with a multi-disciplinary board including legal and data protection experts to sign off. **4. Open-Access LLM Model:** Ensure the acting Data Custodian reviews with the rest of the team the design and plan and ensures it's compliant and aligned with the DPIA and that you have sufficient controls to mitigate personal data leakage through training data extraction and prompt injection attacks.	• ICO: UK GDPR Guidance and Resources • ICO: What is the impact of Article 22 of the UK GDPR on fairness? • NCSC: Protecting bulk personal data • GDPR Compliance Guidelines by EU Commission ISO/IEC 27001: Information Security Management Systems
2.5.1 Organisations **should** ensure that employees are encouraged to proactively report and identify any potential security risks in AI systems and ensure	A lack of proactive reporting and identification of security risks in AI systems can lead to undetected vulnerabilities, increasing the	***Support proactive reporting of security risks.*** *Establish a clear, accessible process for employees to report potential security risks in AI systems, encourage a culture of proactive risk identification by providing training, communication channels, transparent handling, and recognition for reporting issues.* **1. Chatbot App:** Develop an incident reporting template specifically for chatbot-related risks (e.g., sensitive data leakage or inappropriate responses). Use collaborative tools (e.g., messaging channels) to establish a dedicated risk-reporting channel.	• ICO: Reporting Processes • ICO: Breach identification, assessment and logging

appropriate safeguards are in place	likelihood of security breaches, data leaks, or misuse of AI, with potentially significant operational, financial, and reputational consequences.	**2. ML Fraud Detection:** Train employees to identify potential risks such as biases in fraud detection or false positives/negatives. Provide an anonymous reporting mechanism for concerns and include follow-ups on how identified risks are addressed. **3. LLM Platform:** Create a centralised risk registry for employees to log concerns about API misuse, data exposure, or unexpected system outputs. Provide regular updates on how identified risks are managed and mitigated. **4. Open-Access LLM Model**: Provide a checklist for team members and external contributors to log risks as tickets in team's work management board. Review reported risks as part of the work and ensure resolution steps are documented and shared.	• NCSC - Developing a positive cyber security culture • NCSC Responding to a cyber incident – a guide for CEOs
2.6 Where the AI system will be interacting with other systems or data sources, (be they internal or external), Developers and System Operators **shall** ensure that the permissions granted to the AI system on other systems are only provided as required for functionality and are risk assessed.	There is huge potential and interest in "agentic systems", where an AI system can decide and conduct its own actions, typically through integrations with other systems. However, as the actions an AI system may take are not fully predictable, and may be coerced by an attacker, extreme care must be taken when provisioning accounts or other access that the AI system will use. Failure to do this robustly might introduce the potential for unauthorised access, data exfiltration, and privilege escalation.	***Least-privilege access to data and systems accessed by AI System.*** *Mandate a risk assessment process before a component can be used covering provenance, known risks, and evaluations. Ensure that the assessment covers all possible model states, not just the designed or expected ones.* **1. Chatbot App:** If app uses external services to enrich LLM input or drive systems (for instance a booking system), implement data minimisation and granular least-privilege access policies for integration endpoints. Examine what would happen if the proposed integrations are used in the wrong order or for unintended purposes. Review against Excessive Agency as defined in OWASP Top 10 for LLM Applications and consider listing all the proposed integrations and their permissions and asking a security specialist what harm they could cause the organization if given those permissions and integrations. **2. ML Fraud Detection**: Review and test the data inputs used (including data preprocessing and enrichment) and ensure only the required data is used. Evaluate side-effects if model outputs (predictions) are used to drive downstream services e.g. automated processes. **3. LLM Platform**: If the system allows adding extra features, such as plugins or connections to other tools (e.g., a calendar app, an API, or an email service), test how these features work together and check for risks. For example, ensure that the system doesn't produce harmful or incorrect results when someone uses these extra features, like accessing confidential data, executing system commands, or sending misleading emails. **4. Open-Access LLM Model:** The small organisation follows a similar approach to the LLM Provider but tailored to its workflows and level of resources.	• ICO: Assessing security and data minimisation in AI. • MITRE ATLAS: Privilege Escalation • NCSC – Using a cloud platform securely - Apply access control • NCSC Zero trust architecture design principles • NIST AI RMF • OWASP Top 10 for LLM Applications: Excessive Agency
2.7 If a Developer or System Operator chooses to work with an external provider, they **shall** undertake a due diligence	Collaborating with external providers without assessing their adherence to CoP can lead to increased	***Security Review of External Providers:*** *Verify the external provider's implementation of the Code of Practice with the external provider and their overall regulatory compliance.* **1. Chatbot App:** Ask your cloud, LLM, and other providers to provide evidence of CoP compliance	DSIT: AI Cybersecurity Code of Practice

assessment and **should** ensure that the provider is adhering to this Code of Practice.	vulnerabilities, such as lack of regulatory compliance, insecure systems, or inadequate response protocols, which could compromise the entire system's security.	**2. ML Fraud Detection**: Ask your cloud or other service providers to provide evidence of CoP adherence. **3. LLM Platform**: Follow the same approach as the ML Fraud Detection example. **4. Open-Access LLM Model:** Ask the cloud and other service providers for evidence of CoP adherence; In the absence of specific CoP adherence documentation, review provider documentation to ascertain adherence and document your findings in a wiki page.	
Principle 3: Evaluate the threats and manage the risks to your AI system			
Provisions	**Related threats/risks**	**Example Measures/Controls**	**Reference/Resource**
3.1 Developers and System Operators **shall** analyse threats and manage security risks to their systems. Threat modelling should include regular reviews and updates and address AI-specific attacks, such as data poisoning, model inversion, and membership inference.	AI systems face unique threats, such as data poisoning, model inversion, and membership inference attacks, which traditional threat models may not account for. New threats will emerge that will need to be incorporated in threat modelling and risk management.	***Perform Threat Modelling including AI threats:*** *Apply threat modelling that captures potential impacts on stakeholders including both AI and traditional cyberattacks. Document each identified threat in detail, outlining the likelihood and severity of potential impacts to the AI model and the broader system and list mitigations using standardised OWASP or MITRE controls.Both OWASP AI Exchange or MITRE ATLAS provide threat taxonomies and related mitigations which both types of attacks and you can use them in threat modelling. If your AI system processes or was built on personal data ICO's guidance on AI and security is useful to consult for regulatory compliance.* **1. Chatbot App:** Map threats and data flows for the app focusing on traditional and LLM threats. Include un-used functionality of the chosen models or components. For instance, if a multi-modal model is being used just for language, then model the risks if someone were to conduct attacks or abuse through giving it images. **2. ML Fraud Detection**: Include both the inference system and the development environment to cover poisoning, model tampering, serialisation attacks in addition to run-time attacks such as evasion, extraction, inversion, and inference. **3. LLM Platform**: Follow the approach described in Fraud Detection, but with focus on generative AI risks such as overreliance, jailbreaking the LLM, and their safety, ethical, social, and regulatory consequences. **4. Open-Access LLM Model:** The small organisation performs the same type of modelling as in the previous LLM Platform example.	• ICO: Assessing security and data minimisation in AI. • NIST AI RMF – AI RMF Core - Map • NCSC Risk Management – Threat Modelling • OWASP: Threat Modelling Process • MITRE ATLAS • OWASP Top 10 for LLM applications • OWASP AI Exchange
3.1.1 The threat modelling and risk management process **shall** be conducted to address any security risks that arise when a new setting or configuration option is	Failure to conduct threat modelling and risk management when implementing or updating settings or configurations during the AI lifecycle can lead to	***Conduct Threat Modelling for Configuration Changes*** ***Description****: Perform threat modelling whenever settings or configurations are implemented or updated to identify and mitigate security risks throughout the AI lifecycle.* **1. Chatbot App:** When enabling or modifying user feedback options, assess potential risks, such as injection attacks through input fields, and implement mitigations like input validation and sanitisation.	• NIST – Guide to Data-Centric System Threat Modelling • NCSC – Introduction to Logging for Security Purposes

implemented or updated at any stage of the AI lifecycle.	unmitigated security vulnerabilities, such as configuration errors or unanticipated attack vectors, increasing the risk of exploitation and system compromise.	**2. ML Fraud Detection**: When feature selection or detection thresholds change, or new geolocation risk tables are deployed, evaluate risks like adversarial evasion attacks. Develop safeguards such as monitoring for unusual patterns and conducting stress tests on configurations. **3. LLM Platform**: When changing user authentication settings, evaluate threats for unauthorised access, and implement mitigations like rate limiting, lockouts, and enhanced logging. **4. Open-Access LLM Model**: When modifying the model's configuration to allow community contributions or plugins, assess and mitigate risks such as the introduction of malicious code or unintended functionalities.	• OWASP Top 10 for APIs - API4:2019 Lack of Resources & Rate Limiting • OWASP Threat Modelling in Practice
3.1.2 Developers **shall** manage the security risks associated with AI models that provide superfluous functionalities, where increased functionality leads to increased risk. For example, where a multi-modal model is being used but only single modality is used for system function.	Allowing AI models to retain superfluous functionalities that are not required for the system's purpose can introduce unnecessary security risks, such as expanded attack surfaces, increased vulnerability to exploitation, and potential misuse of unused features, compromising the overall security of the system.	***Restrict Superfluous Functionalities:*** *Limit AI model functionalities to those essential for the system's purpose to reduce the attack surface and minimise security risks associated with unused features.* **1. Chatbot App:** If the chatbot is implemented for text-based customer support, use guardrails or API blocking to disable or restrict any access to unused multimodal capabilities, such as speech-to-text or text-to-speech features to prevent unintended interactions or vulnerabilities. **2. ML Fraud Detection**: If the system only analyses transactional data, remove or disable unnecessary model features, such as image processing or location-based predictions to minimise risks and complexity. Only enable advanced features, such as the use of RL algorithm to explore a complex space, if the security implications are fully understood. **3. LLM Platform**: Provide different APIs for text from ones including advanced capabilities like multimodal input (e.g., image processing) for application only uses text. Provide specific voices in text to voice scenarios instead of allowing voice cloning. **4**. **Open-Access LLM Model**: Since the focus is for legal advice and contract negotiation, implement safety measures to disable general purpose or other use. Provide lightweight documentation to users about the rationale and security benefits of these restrictions.	• NCSC Secure Design Principles • OWASP AI Top 10 API Security Risks - 2023 • OWASP Top 10 for LLM Applications: Excessive Agency • Building Guardrails for Large Language Models
		Integrate Threat Modelling with AI Governance: *Require completed threat models for governance approval at critical stages of the AI lifecycle, ensuring documented risk understanding and mitigation before deployment providing support and guidance and ensuring cross-discipline input (ethics, privacy, legal, etc) to threat modelling.* **1. Chatbot App**: Establish governance policies that mandate a formal review of the threat model, including stakeholder and impact assessments, prior to each major system deployment. Provide a standardised threat modelling template using standard notation e.g. STRIDE or PASTA with AI threats found in MITRE ATLAS or OWASP AI Exchange. **2. ML Fraud Detection:** Introduce the need for a threat model as part of deployment approval. **3. LLM Platform:** As in the Chat Bot App, with the addition of a formal multi-disciplinary threat model review before approval. **4. Open-Access LLM Model:** This is not applicable to small organisations; instead, facilitate threat model by using reusable standardised templates in free diagrammatic tools.	• OWASP Threat Modelling Playbook • NIST AI RMF – AI RMF Core • NCSC Risk Management – Cybersecurity Risk Management Framework

3.1.3 System Operators **shall** apply controls to risks identified through the analysis based on a range of considerations, including the cost of implementation in line with their corporate risk tolerance.	When risk tolerance is not clearly defined in the context of AI-specific risks such as data poisoning or model misuse, this could result into Inadequately prioritized controls leading to breaches, operational disruptions, or unethical decision-making.	***Develop a Prioritisation Framework for AI Risk Controls:*** *Use an AI-specific risk-scoring system to prioritise mitigations and controls based on the impact of threats, likelihood of occurrence, and in alignment with organizational risk tolerance. This should account for regulatory risk, including data protection, and cover AI-specific vulnerabilities, such as adversarial manipulation, model drift, and bias.* **1. Chatbot App:** Indirect Prompt Injections and Bias are rated as High when the app is used for recruitment but Low when used for summarisation of internal documentation. **2. ML Fraud Detection**: Poisoning Backdoor and Evasion attacks are prioritised as High to avoid costly fraudulent transactions. **3. LLM Platform**: AI risks identified for general-purpose models are elevated as High including copyright violations which may result into legal liability **4. Open-Access LLM Model:** Follow a similar approach to the LMM Platform example.	• NIST AI RMF Playbook • NIST AI RMF, Use Cases, Autonomous Vehicle Risk Management Profile for Traffic Sign Recognition
3.2 Where AI security threats are identified that cannot be resolved by Developers, this **shall** be communicated to System Operators so they can threat model their systems. System Operators **shall** communicate this information to End-users, so they are made aware of these threats. This communication **should** include detailed descriptions of the risks, potential impacts, and recommended actions to address or monitor these threats.	Without clear communication on unresolved risks, System Operators and End-users may lack awareness, limiting their ability to apply safeguards effectively.	***Document and Communicate Identified Unresolved Risks:*** *Ensure clear documentation and timely communication of any unresolved threats to all relevant stakeholders.* **1. Chatbot App:** If the app performs document summarisation, the application maybe vulnerable to indirect prompt injection. Ensure system operators are aware so that they can introduce mitigations such PDF checks and reviews. **2. ML Fraud Detection**: Notify system operators to develop real-time monitoring of inputs for suspicious patterns to cover residual evasion attack risks **3. LLM Platform**: Document public API documentation with known risks and how to mitigate them. **4. Open-Access LLM Model:** Inform model users about potential risks like model misuse for generating biased or harmful outputs, and document recommended safeguards such as usage guidelines or implementing content moderation mechanisms.	• NIST AI RMF Playbook • NIST AI RMF, Use Cases, Autonomous Vehicle Risk Management Profile for Traffic Sign Recognition
3.3 Where an external entity has responsibility for AI security risks identified within an organisations infrastructure, System Operators **should** attain assurance that these	Reliance on third parties without adequate verification could expose the AI system to unmanaged vulnerabilities.	***Conduct AI-Specific Security Assessments for Third Parties:*** *Ensure third-party components and vendors undergo security assessments that specifically address AI-related risks and adherence with the AI Code of Practice.* **1. Chatbot App:** Request from the model provider to provide assurances addressing specific AI risks, such as model safety and secure data handling. **2. ML Fraud Detection**: Require similar assurances similar to the Chatbot App example but from external data providers to ensure data used for training is handled responsibly and securely	• ISO 9001 - What does it mean in the supply chain? • NCSC Supply chain security guidance • NCSC Cloud Security Guidance – Choosing a cloud provider

parties are able to address such risks.		**3. LLM Platform**: See the ML Fraud Detection Example. **4. Open-Access LLM Model:** Implement a lightweight approach by focusing on key external dependencies. For example, request a simple self-assessment checklist from any external providers (e.g., cloud hosting, pre-trained models, or APIs) to ensure they address basic AI risks such as data privacy, model integrity, and adherence to security standards. Limit reliance on complex external integrations to reduce potential vulnerabilities.	• World Economic Forum: Adopting AI Responsibly: Guidelines for Procurement of AI Solutions by the Private Sector: Insight Report
3.4 Developers and System Operators **should** continuously monitor and review their system infrastructure according to risk appetite. It is important to recognise that a higher level of risk will remain in AI systems despite the application of controls to mitigate against them.	Residual risk can be exploited by malicious actors, especially as evolving threats introduce new vulnerabilities or amplify existing ones, leading to potential breaches, disruptions, or compromised AI integrity.	***Establish Continuous AI Risk Monitoring Controls:*** *Implement a regular review processes of AI developments to determine whether emerging vulnerabilities, improved mitigation techniques, or advancements in AI models necessitates updates to the risk assessment controls.* **1. Chatbot App:** Threat intelligence feeds report that prompt injection attacks, including advanced techniques like using emoticons to bypass safety measures, have suddenly gained popularity in the hacking community. As websites and applications face widespread probing for these vulnerabilities, the organization mitigates the risk by developing and deploying in-house guardrails to detect and block such attacks. **2. ML Fraud Detection**: An updated version of the third-party model used for fraud-detection includes additional adversarial training to withstand evasions, leading to its selection, fine tuning and deploying the new version. **3. LLM Platform**: A new government report highlights risks associated with audio generation, such as the potential for voice cloning to bypass voice authentication systems. In response, the platform develops new safeguards to prevent misuse of its audio generation capabilities. **4. Open-Access LLM Model:** Review of a new research paper demonstrates a novel attack vector to jailbreak model, necessitating the development of additional safety features.	• MITRE AI Risk Database • NIST SP 800-137 • Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations • NCSC Early Warning • NCSC – Threat Intelligence
Principle 4: Enable human responsibility for AI systems			
Provisions	**Related threats/risks**	**Example Measures/Controls**	**Reference/Resource**
4.1 When designing an AI system, Developers and/or System Operators **should** incorporate and maintain capabilities to enable human oversight.	Without built-in human oversight, AI systems will generate incorrect outputs or decisions that are difficult to interpret, verify, or override, increasing risks of data protection compliance, unintended consequences,	***Implement Mechanisms for Human Oversight:*** *Control: Implement features that allow human operators to easily interpret, verify, and act on AI outputs, including manual release and overrides. Ensure that the design meet obligations around automated decisions in the UK GDPR Article 22 and encourages meaningful human decision-making rather than passive acceptance of AI recommendations.* **1. Chatbot App:** For new features that allow the chatbot to take autonomous actions, such as scheduling appointments or order office supplies, an override control has been applied to cancel appointments or orders, because of its low to moderate impact. By contrast a feature to provide personalised packages to customers, has high reputational and legal risks, as a result manual release control has been implemented with an operator review-and-approve interface.	• ICO Audit Framework toolkit on AI - Human review • ICO Guidance on AI and Data Protection - What is the impact of Article 22 of the UK GDPR on fairness? • CSA Companion Guide on Securing AI Systems, Section 2.2.4 Operations and Maintenance.

	misuse, or harmful impacts	**2. ML Fraud Detection**: Explanation techniques such as SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-agnostic Explanations) are implemented, to enable operators to understand the key factors behind each decision and intervene, if necessary, in compliance with UK GDPR Article 22, which grants individuals the right not to be subject to decisions based solely on automated processing that produce legal effects concerning them or that significantly affect them. **3. LLM Platform**: Incorporate a feature that allows human moderators to review and approve AI-generated content before publication, ensuring outputs align with ethical guidelines and community standards. **4. Open-Access LLM Model:** Provide users with tools to flag and report inappropriate or harmful AI-generated content, facilitating human oversight and continuous improvement of the model's outputs.	• NISTIR 8312: Four Principles of Explainable Artificial Intelligence • NCSC Machine Learning Principles. NCSC Guidelines for Secure AI system development, Secure Operation and Maintenance.
		Measure and Validate Accuracy of Human Oversight Decisions: *Regularly test and measure the accuracy of human oversight decisions, validating that operators can correctly interpret and act on AI outputs and identifying areas for improvement. Assess not just individual performance but how the system supports human understanding and engagement to foster effective sociotechnical communication between the operator and AI.* **1. Chatbot App:** For a hiring version of the chatbot app, regularly review a sample of candidates who were automatically flagged as unsuitable by the AI. Assess whether human operators correctly validated or overrode these decisions. Identify patterns of misinterpretation or bias in operator actions, and use these insights to refine training, decision guidelines, or the AI's recommendation criteria. **2. ML Fraud Detection**: Test whether human reviewers accurately validate flagged transactions, ensuring they can effectively distinguish between true fraud cases and false positives. **3. LLM Platform**: Evaluate how well operators identify and correct biased or inappropriate content generated by the platform, using flagged examples to improve content moderation guidelines and model outputs. **4. Open-Access LLM Model:** This is a nice to have for a small organisation and in this case the company might conduct experiments to evaluate model responses and feedback from operators on how to improve as in the above.	• ICO Audit Framework toolkit on AI - Human review • ICO: Explaining Decisions made with AI • NISTIR 8312: Four Principles of Explainable Artificial Intelligence
4.2 Developers **should** design systems to make it easy for humans to assess outputs that they are responsible for in said system (such as by ensuring that models outputs are explainable or interpretable).	Without clarity and ease of use, users may not perform oversight effectively leading to failures and harm.	***Develop User-Friendly Human Responsibility UI:*** *Implement UIs that display outputs, decision-making rationales, and logs clearly to make it easy for human operators to assess outputs and understand their accountability. Ensure systems are designed to encourage rigorous assessment by humans and not condition them to simply click an approve button.* **1. Chatbot App:** As the chatbot is extended to cover new cases, studies of the existing usage are analysed and consolidated in an UX library for consistent implementation of oversight and human responsibility features **2. ML Fraud Detection**: Include a dashboard that shows flagged transactions with explanations of the model's decision factors, providing a clear interface for review.	ICO: Explaining Decisions made with AI NISTIR 8312: Four Principles of Explainable ArtificialIntelligence • Multicalibration for Confidence Scoring in LLMs

		3. LLM Platform: Review explainability review and provide a web interface for API requests to include confidence scores and reasoning summaries. **4. Open-Access LLM Model:** Offer an API with an option to receive back responses to legal queries with references referencing specific legal principles or past cases that influenced the output.	From Understanding to Utilization: A Survey on Explainability for Large Language Models
4.3 Where human oversight is a risk control, Developers and/or System Operators **shall** design, develop, verify, and maintain technical measures to reduce the risk through such oversight.	Ineffective oversight technical measures may compromise the risk reduction effort by overburdening or failing to adequately support human reviewers.	***Implement Validation and Enforcement of Oversight controls:*** *Design and implement technical measures that provide guardrails to assist human reviewers in understanding, interpreting, and acting on AI outputs.* **1. Chatbot App:** For a chatbot that can make automated triaging decisions, provide human reviewers with a summary of the chatbot's reasoning for triage decisions (e.g., key user inputs) and allow them to adjust or override decisions before escalation. **2. ML Fraud Detection**: Provide human reviewers with flagged transactions prioritised by risk level and accompanied by explainable insights (e.g., key features influencing the fraud score), to ensure informed and efficient decision-making. **3. LLM Platform**: Integrate API-based guardrails that automatically flag AI-generated content containing sensitive information or potential biases, providing a score as an API field for human reviewers to identify outputs requiring attention. **4. Open-Access LLM Model:** Train the model with additional safety features to apply AI security measures such as adversarial robustness checks, and warning mechanisms for potentially misleading or harmful outputs. These measures may include applying confidence thresholds, logging flagged outputs, and ensuring model responses align with compliance policies.	• ICO Audit Framework toolkit on AI - Human review • Building Guardrails for Large Language Models
4.4 Developers **should** verify that the security controls specified by the Data Custodian have been built into the system.	Without validation of Data Custodian controls, the system may lack necessary data protection and governance measures, potentially leading to security vulnerabilities or regulatory non-compliance.	***Conduct Validation of Custodian:*** *Verify that all controls specified by the Data Custodian have been implemented correctly, with testing to validate effectiveness and alignment with data protection requirements and guidance.* **1. Chatbot App:** Ensure that data retention policies adhere to the Data Custodian's specifications by implementing automated data purging mechanisms and conducting regular audits to confirm compliance. **2. ML Fraud Detection**: Validate that data anonymisation controls specified by the Data Custodian are active and effective in protecting customer privacy. **3. LLM Platform**: Confirm that access controls and encryption protocols for the LLM platform's API endpoints meet the Data Custodian's requirements by performing penetration testing and security assessments. **4. Open-Access LLM Model:** Verify that the model's training data complies with the Data Custodian's guidelines on data sourcing and consent by reviewing data collection processes and conducting compliance checks.	• ICO: Audits • ICO: Audits, Artificial Intelligence Audits
4.5 Developers and System Operators **should** make End-users aware of prohibited use cases of the AI system.	Without clear communication on prohibited uses, end-users may unintentionally	***Document and Train Users on Prohibited Use Cases:*** *Clearly define and document prohibited use cases for the AI system, ensuring end-users understand limitations and restrictions. Use threat modelling to identify and inform users of all known harmful states and unmitigated risks.*	• ICO Accountability and Governance Implications for AI

	misuse the AI system, leading to legal, ethical, or operational risks.	**1. Chatbot App:** In document summarisation use of the app, guide the user at the beginning of each conversation and online documentation not to upload classified internal documents, explaining the risks of data memorisation and leaks**.** **2. ML Fraud Detection**: Threat modelling has identified that AI could be abused to monitor a person's spending habits without consent. Document and communicate to operators that using the system for non-compliance-related surveillance is prohibited. Provide training on ethical boundaries and enforce compliance audits to ensure proper usage. **3. LLM Platform**: Document prohibited use cases in use policies and T&Cs for APIs. **4. Open-Access LLM Model:** Threat modelling highlighted that the open-access LLM could be fine-tuned or deployed to generate misinformation or manipulate public opinion. Clearly communicate in the licensing terms and documentation that using the model for misinformation campaigns or malicious automation (e.g., phishing scams) is strictly prohibited.	
		Monitor for Prohibited Use Cases: *Implement controls to actively monitor, detect, and prevent prohibited use cases.* **1. Chatbot App:** Implement guardrails to detect and block use of personal data supported with additional automated tests and periodic audits of log. **2. ML Fraud Detection**: Implement monitoring of patterns of unauthorised access or analysis triggering escalation alerts **3. LLM Platform**: Use finetuning to add safety measures blocking prohibited use cases, API guardrails to detect and prevent misuse, activity monitoring, and output watermarking to detect misuse in prohibited cases. **4. Open-Access LLM Model:** Finetune to implement safety measures to detect and block prohibited uses, and watermarking model output to identify misuses in phishing attacks.	• ETSI TR 104 032- Securing Artificial Intelligence (SAI); Traceability of AI Models – 5.3 Watermarking • NIST AI RMF Playbook • OWASP Top 10 for LLM applications • OWASP AI Exchange • Building Guardrails for Large Language Models • OWASP - LLM and Generative AI Security Solutions Landscape

Principle 5: Identify, track, and protect your assets			
Provisions	**Related threats/risks**	**Example Measures/Controls**	**Reference/Resource**
5.1 Developers, Data Custodians and System Operators **shall** maintain a comprehensive inventory of their assets (including their	Without a clear understanding of AI assets and their dependencies, organisations may not be able to provide protection and risk exposing	***Establish an AI Asset Inventory:*** *Create and maintain a centralised inventory that records all AI assets, including datasets, models, software dependencies, hardware resources, and system configurations.* **1. Chatbot App:** Document all chatbot versions customised for different use cases, training datasets, software libraries, and APIs used. Include components used for RAG and/or embeddings generation.	• NCSC Machine Learning Principles, 2.3 Manage the full life cycle of models and datasets • NCSC Guidelines for Secure AI System

interdependencies / connectivity).	their AI systems to unauthorised access, data leakage, and vulnerability to external attacks.	2. **ML Fraud Detection**: Use a detailed register of all AI models in use, listing model version, source, datasets used for training, and any updates or retraining conducted. This includes the software libraries used in the development or running of the models. As an optional safeguard use scripts and scanning tools to generate an MLBOM as machine readable inventory list **3. LLM Platform**: Use the same approach as in the Fraud Detection example. **4. Open-Access LLM Model:** Use the same approach as in the Fraud Detection example.	Development, Secure Development. CSA Companion Guide on Securing AI Systems, Section 2.2.2 Development, Item 2.3
5.2 As part of broader software security practices, Developers, Data Custodians and System Operators **shall** have processes and tools to track, authenticate, manage version control, and secure their assets due to the increased complexities of AI specific assets.	Without secure tracking, version control, and authentication, AI systems may be vulnerable to unauthorised changes, data integrity issues, and version conflicts, leading to compromised reliability and security. This is exacerbated by the rapid changes in Gen.AI tool and models.	***Implement AI Asset Tracking*** *Use version control and ML Ops systems to manage and track changes to AI assets including models, datasets, and related software components ensuring transparency and rollback capability.* **1. Chatbot App:** Use Git to track changes to conversation flows and training datasets, ensuring traceability when new features or intents are added. Use an internal package repository mirror is used for components. **2. ML Fraud Detection**: Version control can track training data updates, capturing modifications to ensure traceability and data integrity whereas a model registry is used similarly for models. An internal package repository mirror is used for components. **3. LLM Platform**: Implement the same tracking described in the Fraud Detection example, with the addition of the vendor saving model weights in a protected and isolated storage using a separate dedicated enclave as recommended by the CISA Joint Cybersecurity Information - Deploying AI Systems Securely **4. Open-Access LLM Model:** Implement a model registry and scripts to log versioned training datasets, model weights, and configuration files. Use hash-based verification to detect unauthorized modifications to publicly shared assets. Protect package dependency files (e.g. requirements.txt) and keep a copy of packages for each release.	See References in *"Establish an AI Asset Inventory"*
		Authenticate, Authorize and Log Access to Assets: *Enforce strict access controls and authentication protocols to limit asset modifications to authorised personnel only logging any access.* **1. Chatbot App:** Access to system prompt, prompt templates, datasets and embeddings used in RAG are restricted to data scientists using RBAC and MFA with updates only allowed via CI pipelines. **2. ML Fraud Detection**: Restrict access to training datasets and models using RBAC and via APIs and CI pipelines with appropriate approvals for critical assets. All access events to sensitive datasets are logged, and alerts are triggered for access attempts from unauthorized users or unusual access patterns. Trigger alerts when models or data are updated outside the process. **3. LLM Platform**: See Fraud Detection Example for this control. **4. Open-Access LLM Model:** See Fraud Detection Example for this control.	

5.3 System Operators **shall** develop and tailor their disaster recovery plans to account for specific attacks aimed at AI systems.	Without tailored disaster recovery, AI systems may be unprepared for incidents such as data poisoning, or large language model (LLM) weaponisation, leaving the organisation vulnerable to prolonged disruption leading to data leakage, DoS or compromised model performance. Maintaining a reliable known good state can be challenging, particularly with continuous learning models or systems with frequent updates, increasing the risk of losing data integrity	***Incorporate AI-Specific Threat Scenarios into Recovery Plans:*** *Update disaster recovery plans to address AI-specific risks, including adversarial attacks, data poisoning, and model drift and ensure readiness for prompt recovery.* **1. Chatbot App:** Include a recovery plan to address adversarial attacks, such as prompt injections to compromise chatbot responses, by maintaining fallback mechanisms to disable automated responses temporarily while restoring integrity. **2. ML Fraud Detection**: Develop recovery plans for data poisoning incidents, including manual backup processes to maintain operations until a tested model checkpoint with reliable detection and adversarial robustness is restored. **3. LLM Platform**: Similar approach to Fraud Detection; consider ensuring the system remains operational during attacks by using backup infrastructure ('high availability ') or placeholders to quickly deploy tested model versions. **4. Open-Access LLM Model:** Decide on how to roll back to a previous healthy version of the model or data if poisoning or other exploitable AI vulnerabilities are identified. Automate the process of restoring backups to speed up recovery.	• CISA, JCDC, Government and Industry Partners Conduct AI Tabletop Exercise
5.3.1 System Operators **should** ensure that a known good state can be restored.	Inability to restore a known good state can lead to prolonged system downtime, data loss, or operational disruptions after failures or attacks	***Establish and Maintain a Known Good State:*** *Regularly backup AI models, data, and system configurations, maintaining a known good state that can be restored after a disruption. Good state may contain additional internal state to help a model reach optimal performance after model warm-up. This may not be always possible, and a new model will have to be trained addressing the attack. Nevertheless, backups will help accelerate developing a mitigation.* **1. Chatbot App:** Maintain backups of system prompts, RAG data and embeddings, prompt templates, LLM API access details, and other configuration details. **2. ML Fraud Detection**: Maintain backups of the latest clean dataset and model versions, allowing quick restoration if the current version is compromised with a backdoor attack or is discovered to have been trained on poisoned data. **3. LLM Platform**: Retain secure versions of models, weight files, fine-tuning checkpoints, and system configurations to facilitate recovery and training datasets to	

		quickly roll back if vulnerabilities such as poisoning attacks or unauthorised modifications are detected **4. Open-Access LLM Model:** Retain secure copies of model versions and training datasets to quickly roll back and release a new version if vulnerabilities such as poisoning attacks or unauthorised modifications are reported.	
		Develop Recovery plans for advanced scenarios: *Some incidents will include abuse and weaponisation of AI systems, especially Generative AI, to stage misinformation or other attacks abusing an organisation's system. Recovering from such as attacks will require multi-disciplinary expertise and response strategies to minimise impact and harm.* **1. Chatbot App:** Design a process that addresses adversarial take-over of the chatbot to spread misinformation or exploit unused multi-modal capabilities to create deep fakes. This is used after authorities notify you, the service has been used for the last six months for misinformation campaigns. **2. ML Fraud Detection**: Not applicable in predictive ML. **3. LLM Platform**: As in the Chatbot App scenario. **4. Open-Access LLM Model:** Consider using watermark verification to help authorities and customers using the model to deal with advanced attacks.	• OWASP Guide for Preparing and Responding to Deepfake Events • ETSI TR 104 032- Securing Artificial Intelligence (SAI); Traceability of AI Models – 5.3 Watermarking
5.4 Developers, System Operators, Data Custodians and End-users **shall** protect sensitive data, such as training or test data, against unauthorised access.	Unauthorised access to sensitive data, such as training datasets, training algorithms, hyper parameters and model parameters can lead to privacy violations, data breaches, or compromised model integrity, increasing regulatory and reputational risks.	***Implement Data Encryption at Rest and in Transit:*** *Encrypt sensitive data at rest and in transit to protect against unauthorized access, ensuring data confidentiality and security.* **1. Chatbot App:** Encrypt all user chat logs stored in cloud storage using server-side encryption with customer-managed keys. Use HTTPS with TLS 1.3 to secure communications between the chatbot and the end user, preventing data interception. **2. ML Fraud Detection**: Encrypt transaction data stored in the database (at rest) using AES-256 encryption. Use TLS 1.3 to encrypt data transmitted between the fraud detection model and retailer APIs, ensuring compliance with PCI DSS standards. Regularly review encryption settings to ensure they meet evolving regulatory requirements. **3. LLM Platform**: Encrypt model artifacts and training datasets stored in the development environment using industry-standard encryption algorithms. Use encrypted channels (e.g., SFTP or TLS) to transmit training data to remote servers during deployment. Periodically validate the encryption configurations compliance with the ISO 27001 standard. **4. Open-Access LLM Model:** Store training datasets encrypted and use HTTPS or SFTP for all data transfers to prevent unauthorised access during transfers.	• ICO: A guide to data security • ISO/IEC 27001:2022 Information security, cybersecurity and privacy protection — Information security management systems — Requirements • NCSC – Cloud Security Guidance – Using a cloud platform securely • OWASP Application Security Verification Standard (ASVS) OWASP AI Exchange
		Implement Strong Data Access Controls*: Restrict access to sensitive data to authorised personnel only, using multi-factor authentication and role-based access controls. Programmatic updates via pipelines have strict RBAC and approvals in place to prevent abuse.*	See References in "*Implement Data Encryption at Rest and in Transit"*

		1. Chatbot App: Apply RBAC with strict least-privilege access to datasets used in RAG including vectors and embeddings. **2. ML Fraud Detection**: Enforce access controls at multiple levels by restricting access to training datasets containing sensitive customer information at storage level and limiting access to the training environments. Ensure only essential team members and approved pipeline roles have permissions, with strict enforcement through RBAC and strong authentication across all layers. **3. LLM Platform**: Same as in the ML Fraud Detection example. **4. Open-Access LLM Model:** Same as in the ML Fraud Detection example.	
		Data Access and Usage Monitoring: *Implement automated monitoring tools to monitor access and usage of sensitive data including proprietary model weights, generating alerts for any unusual patterns or unauthorized attempts.* **1. Chatbot App:** Set up automated alerts for access to protected RAG data, ensuring that unauthorised access is immediately flagged. **2. ML Fraud Detection**: Set up automated alerts for access to training /testing datasets and model weights and configuration files. **3. LLM Platform**: Similar implementation as in the Fraud Detection. **4. Open-Access LLM Model:** Alert for unusually high failed attempts to access and utilise cloud or platform controls to alert for high volume downloads indicating potential data exfiltration.	See References in "*Implement Data Encryption at Rest and in Transit"*
5.4.1 Developers, Data Custodians and System Operators **shall** apply checks and sanitisation to data and inputs when designing the model based on their access to said data and inputs and where those data and inputs are stored. This **shall** be repeated when model revisions are made in response to user feedback or continuous learning.	Lack of data and input sanitization can introduce biases, errors, or malicious data, leading to compromised outputs, security risks, as well as potential legal and reputational harm.	***Apply Data Sanitisation and Validation:*** *Ensure that all data—both training datasets and runtime inputs—are sanitised and validated to prevent data poisoning, malicious inputs, and biases. During training, verify that datasets are clean, unbiased, and aligned with model objectives.* **1. Chatbot App:** Apply data sanitisation in incoming user prompts to reduce the risk of prompt injections. **2. ML Fraud Detection**: Sanitise training data to remove any incorrect or malicious entries that could skew model outputs. **3. LLM Platform:** Apply input sanitisation as part of guardrails; implement automated scripts to scan and sanitize large-scale text corpora for biases, offensive language, or duplicate entries during data preprocessing. **4. Open-Access LLM Model:** Consider use of content filtering and other relevant data preprocessing mechanisms when training the model to check inputs and ensure data quality.	• OWASP AI Exchange • Training Dataset Validation to Protect Machine Learning Models from Data Poisoning

5.4.2 Where training data or model weights could be confidential, Developers **shall** put proportionate protections in place.	Failure to adequately protect sensitive training data or model weights can lead to unauthorized access, intellectual property theft, or exposure of confidential information, increasing the risk of data breaches and regulatory non-compliance.	*See controls and examples in 5.4*	See References in "*Data Access and Usage Monitoring"*
Principle 6: Secure your infrastructure			
Provisions	**Related threats/risks**	**Example Measures/Controls**	**Reference/Resource**
6.1 Developers and System Operators **shall** evaluate their organisation's access control frameworks and identify appropriate measures to secure APIs, models, data, and training and processing pipelines.	Inadequate access control can expose sensitive data, models, and pipelines to unauthorized access, increasing the risk of data leaks, model tampering, or unauthorized modifications.	***Establish Role-Based Access Controls (RBAC):*** *Implement role-based access controls to limit access to AI models, data, and pipelines based on user roles and responsibilities, enforcing the principle of least privilege. This includes research environments. Protect API endpoints and data pipelines by implementing access controls, encryption, and authentication mechanisms to prevent unauthorised access.* **1. Chatbot App:** Restrict access to system prompts, configuration data, access to the underlying LLM API and any data used for RAG. Restrict access to embeddings generation and via pipelines to relevant group of data engineers. **2. ML Fraud Detection**: Restrict access to training data to data scientists and model tuning permissions to ML engineers, ensuring minimal access rights across roles. This should be for both interactive and API-based access. **3. LLM Platform**: Follow the same implementation approach as in the Fraud Detection example of this control. **4. Open-Access LLM Model:** Restrict access to training data to and model tuning permissions to developers, ensuring minimal access rights across roles.	• NCSC Machine Learning Principles, Part 3: Secure deployment • NCSC Guidelines for Secure AI System Development, Secure your Infrastructure. • CSA Companion Guide on Securing AI Systems, Section 2.2.3 Deployment, Item 3.1 • OWASP AI Top 10 API Security Risks - 2023
6.2 If a Developer offers an API to external customers or collaborators, they **shall** apply appropriate controls that mitigate attacks on the AI system via the API. For example,	Externally exposed APIs increase the risk of model extraction attacks, rapid data poisoning, and abuse, potentially compromising the	***Implement API Rate Limiting****: Enforce rate limits on API requests to prevent attackers from overwhelming the system, reverse engineering the model, or rapidly injecting malicious inputs.* **1. Chatbot App:** Apply rate limit to web endpoints or own APIs to prevent using the app's API to indirectly attack the supporting model. **2. ML Fraud Detection**: Apply rate limit to web endpoints and APIs. **3. LLM Platform**: Apply a rate limit of X requests per minute per user to prevent bulk extraction of model responses.	• OWASP Top 10 for APIs - API4:2019 Lack of Resources & Rate Limiting

placing limits on model access rate to limit an attacker's ability to reverse engineer or overwhelm defences to rapidly poison a model.	integrity of the AI system.	**4. Open-Access LLM Model:** Not applicable	
		Use Behavioural Analysis for API Security: *Implement behavioural analysis tools to detect abnormal API usage that could indicate malicious intent, such as model extraction or poisoning attempts.* **1. Chatbot App:** Use behavioural analysis to flag repeated API calls with unusual input patterns that could indicate an attempt to exploit or overload the chatbot's system. **2. ML Fraud Detection:** Monitor API usage for anomalies such as an unusual volume of high-risk transaction queries that may suggest adversarial testing or system probing. 3. **LLM Platform:** Use behavioural analysis with a dual LLM, moderation APIs, or anomaly detection code, to identify sudden spikes in repetitive queries that may signal reverse engineering attempts. **4. Open-Access LLM Model:** Provide guidelines to customers on implementing behavioural monitoring of model use.	• OWASP AI Top 10 API Security Risks - 2023 • NCSC – Logging and Protective Monitoring • LLM Monitoring and Observability — A Summary of Techniques and Approaches for Responsible AI
		Deploy API Gateway with Security Features: *Use an API gateway with security features like throttling, dynamic rate limiting, and any other attack detection features, authentication, and logging to manage and monitor access to external-facing APIs.* **1. Chatbot App:** Use an API Gateway to manage access if the apps API is publicly accessible. **2. ML Fraud Detection:** Use an API gateway to manage access to inference API, logging each request, applying rate limiting, and requiring OAuth for user authentication. 3. **LLM Platform:** Similar implementation to the Fraud Detection example but adding all APIs offered to customers. **4. Open-Access LLM Model:** Not applicable.	• OWASP AI Top 10 API Security Risks - 2023 • Wikipedia: API Management Overview and links
6.3 Developers **shall** also create dedicated environments for development and model tuning activities. The dedicated environments **shall** be backed by technical controls to ensure separation and principle of least privilege. In the context of AI, this is particularly necessary because training data **shall** only be present in the training and	Without separation and environment-specific controls, production-grade sensitive data and models may be exposed to unauthorised access via development or research workflows, leading to data leaks, tampering, unauthorised deployments, or compliance violations.	***Set Up Dedicated Development and Production Environments:*** *Establish separate environments for development, testing, and production, ensuring data and models are only accessible where necessary. Restrict sensitive training data to the development environment, isolating it from production* **1. Chatbot App:** maintain separated environments for the app with restricted access to system prompts, and API config using different LLM API endpoints for each environment. Use synthetic RAG data for development and testing. **2. ML Fraud Detection**: Use a model registry to track and manage versions, ensuring only authorized personnel access and modify them. Leverage anonymized transaction data or synthetic datasets in development to protect sensitive customer information. Implement ML pipelines to automate the promotion of tested models to production with approvals, ensuring compliance, security, and quality standards. **3. LLM Platform**: Similar approach as in the Fraud Detection example of this control. **4. Open-Access LLM Model:** Use lightweight containerization to isolate development and production environments. Restrict sensitive training data to local development only and deploy approved models via automated scripts to maintain separation and minimize manual intervention risks.	• Alan Turing Institute: What is synthetic data and how can it advance research and development • CISA Joint Cybersecurity Information - Deploying AI Systems Securely • DSTL: Machine learning with limited data • NCSC Secure your development Guide • NIST SP 800-218 • Secure Software Development Framework (SSDF)

development environments where this training data is not based on publicly available data.			
6.4 Developers and System Operators **shall** implement and publish a clear and accessible vulnerability disclosure policy.	Without a defined vulnerability disclosure policy, security vulnerabilities may go unreported, exposing the organisation to delayed or missed opportunities to patch critical issues.	***Develop and Publish a Vulnerability Disclosure Policy:*** *Create a vulnerability disclosure policy that details how vulnerabilities can be reported, including timelines for acknowledgment and resolution.* **1. Chatbot App:** Develop a policy using the NCSC Vulnerability Disclosure Toolkit **2. ML Fraud Detection**: develop a disclosure policy to comply with the ISO/IEC 29147:2018 Vulnerability disclosure standard. **3. LLM Platform**: Similar approach as in the Fraud Detection example of this control. **4. Open-Access LLM Model:** publish a policy on the organisation's website, including a contact email for reporting AI vulnerabilities and a commitment to respond within 48 hours.	• NCSC Vulnerability Disclosure Toolkit • ISO/IEC 29147:2018 Vulnerability disclosure standard.
6.5 Developers and System Operators **shall** create, test, and maintain an AI system incident management plan and an AI system recovery plan.	Without a dedicated incident and recovery plan, AI systems may be slow to recover from disruptions, resulting in prolonged downtime or compromised model performance.	***Develop an AI-Specific Incident Management Plan:*** *Create an incident management plan tailored to AI-specific threats, such as discovery of data poisoning, model drift, and adversarial attacks with recovery steps to a known good state, including procedures for validating model integrity.* **1. Chatbot App:** Develop an incident plan to address adversarial inputs causing inappropriate responses and jailbreaking, including isolating malicious queries, and implementing input validation updates. if the third-party LLM API becomes unavailable or produces unreliable responses (e.g., due to downtime or compromised functionality), the recovery plan includes switching to a pre-integrated backup LLM provider or a local fallback model to maintain core functionalities. If no immediate replacement is available, the app transitions to a basic response mode using pre-scripted messages or rule-based logic. **2. ML Fraud Detection**: For a fraud detection model, a 20% spike in fraud signals triggers an incident investigation. The team isolated recent training datasets identified anomalies like skewed entries and rolled back to a prior model version. Containment included notifying teams and deploying enhanced validation checks. **3. LLM Platform**: Create a comprehensive plan of handling jailbreaking, misuse, and model overload including temporarily restricting access, deploying a backup model, and conducting usage audits. **4. Open-Access LLM Model:** Implement a plan to handle community-reported data poisoning incidents by halting any contributions, validating datasets, and rolling back to a previous model checkpoint from its model registry, validating its outputs through automated tests before redeployment	• CSA Incident Response Checklist • NCSC Incident Management • NCSC Vulnerability Disclosure Toolkit
6.6 Developers and System Operators **should** ensure that,	Without clear understanding of cloud service	***Define and Validate Security Clauses in Cloud Contracts:*** *Ensure cloud service agreements explicitly outline security responsibilities, compliance standards, and support provisions, including data protection, access controls, incident*	• CSA - AI Organizational Responsibilities -

where they are using cloud service operators to help to deliver the capability, their contractual agreements support compliance with the above requirements.	agreements, organizations may not know what security measures they can expect or demand from the cloud provider, potentially leaving AI assets exposed to gaps in data protection or compliance failures.	*response, and audit capabilities. Provide detailed documentation to stakeholders to bridge knowledge gaps about the cloud provider's obligations.* **1. Chatbot App:** The developers of the chatbot app switch to a new LLM model provider hosted on the cloud, as it references encryption, data isolation and retention, limited customer data logging and a ban on model retraining with customer data in the T&Cs and cloud contract. This follows the refusal of the previous model provider to cover these requirements in the contract. **2. ML Fraud Detection**: Review and validate cloud provider documentation. **3. LLM Platform**: Similar approach as in the Fraud Detection example of this control. **4. Open-Access LLM Model:** Similar approach as in the Fraud Detection example of this control.	Governance, Risk Management, Compliance and Cultural Aspects • ICO guidance for cloud providers obligations under NIS Regulations 2018 - Security requirements • NCSA Cloud security shared responsibility model • NCSC Cloud Security Principles
Principle 7: Secure your supply chain			
Provisions	**Related threats/risks**	**Example Measures/Controls**	**Reference/Resource**
7.1 Developers and System Operators **shall** follow secure software supply chain processes for their AI model and system development.	Without secure software supply chain processes, AI systems are vulnerable to risks like supply chain attacks, insertion of malicious components, and dependency issues, including models and datasets, which can compromise AI integrity and security. Lack of accountability across the supply-chain for jurisdiction-specific regulations can lead to data breaches and lack of regulatory compliance.	***Safeguard Provenance and Transparency:*** *Mandate in internal standards that components can only be sourced by trusted and approved sources, documenting source, version, licencing, history, and other related artifacts (e.g. Model Card for models); use checksums to verify integrity. SBOMS can help automated creation of signed attestations of all components and their metadata. This can help safeguard transparency and provenance with non-repudiation and component tampering checks. SBOMs cover libraries and system components but not models or datasets. AI and ML BOMs are an emerging field. Monitor progress in standards such as Cyclone DX MLBOM an introduce similar scans when tools emerge. In the meanwhile, rely on ML Ops to enforce model and dataset provenance and use file checksum to verify integrity and provenance.* **1. Chatbot App:** Publish guidelines and developer standards and use SBOMs to document the controls used to build the application applying periodic tests and audits to ensure compliance. Cover models used for embeddings in the process. **2. ML Fraud Detection**: Same implementation to Chatbot App, but compliment SBOMs with an in-house created ML-BOM documenting the model package dependencies and model cards offering a machine-readable record which is validated with a script against approved standards. **3. LLM Platform**: Similar implementation to the Fraud Detection example but for the multimodal model, but adding datasets and auxiliary (RL, embeddings generation, and so on) **4. Open-Access LLM Model:** Agree on the criteria and workflow to manage external components, datasets, and models (foundation if used, RL, embeddings and so on). Use	• ETSI GR SAI 002 - Data Supply Chain Security • CISA - SBOM • MITRE – System of Trust Framework • NIST - Cybersecurity Supply Chain Risk Management • NCSC Supply Chain Security Guidance • NCSC Machine Learning Principles, 2.1 Secure your supply chain • OWASP CycloneDX – ML BOM • OWASP Top 10 for LLM applications - LLM03 Supply-Chain Vulnerabilities Supply-chain Levels for Software Artifacts (SLSA)

		scanning tools and scripts to create signed SBOMs and ML BOMs for machine readable lists.	
		Apply Vulnerability Management: *Implement a vulnerability management process that includes regular (e.g. daily) scanning of third-party components for known vulnerabilities and timely patching to mitigate risks.* **1. Chatbot App:** Use automated vulnerability scanning tools to identify vulnerabilities in third-party libraries or components. Introduce a patch management process where critical vulnerabilities are patched within 48 hours, moderate vulnerabilities within 14 days, and low-risk vulnerabilities within 30 days. **2. ML Fraud Detection**: Use the same approach as in the Chatbot App but with additional model scanning tools and tests for model serialisation attacks for base third party models. **3. LLM Platform**: Similar approach as in the Fraud Detection example of this control. **4. Open-Access LLM Model:** Similar implementation with the Fraud Detection example but use open-source or commercial LLM scanners if you are finetune foundational model; Agree your remediation timeframes for fixing defects with patching focusing on Critical and Highs and triaging Medium and Lows.	• NCSC – Vulnerability Management • OWASP Vulnerability Management Guide • OWASP - LLM and Generative AI Security Solutions Landscape
		Adopt Secure Supply Chain Frameworks: *Follow your organisation's preferred secure supply chain guidance or standard for all stages of AI development, from sourcing and integrating components to testing and deployment.* **1. Chatbot App:** Follow NCSC supply chain security guidance to assess the provider's security policies and ensure compliance with encryption, access control, and other security standards. **2. ML Fraud Detection**: Follow the same advice as in the Chatbot App example; Additionally, ensure training datasets sourced from external vendors comply with data protection legislation, provenance checks, contractual obligations, and avoiding unlawfully collected training datasets. **3. LLM Platform**: Compliment the implementation described in the Fraud Detection example with the additional adoption of ISO 28000:2022. **4. Open-Access LLM Model:** This is not applicable to small organisations, who can consult the standards and NCSC guidance on supply-chain security.	• NCSC supply chain security guidance • NIST C-SCRM • ISO 28000:2022
7.2 System Operators that choose to use or adapt any models, or components, which are not well-documented or secured **shall** be able to justify their decision to use such models or components	Utilizing poorly documented or untrusted AI components because of some of their unique features introduces potential vulnerabilities, increasing risks of data leaks or	***Document Justification for Untrusted Components:*** *When using poorly documented or untrusted components, document the justification, including alternative evaluations, and the absence of better suppliers.* **1. Chatbot App:** Use a chatbot API with superior multilingual capabilities despite limited documentation, justifying the decision with performance benchmarks. **2. ML Fraud Detection**: In the fraud detection scenario, the vendor identifies a fraud-detection model on a public model hub that outperforms all other models in false positives but lack any model cards or other documentation and the developer cannot provide more details. False positive reduction is a key market differentiator for the	• NIST AI RMF • NCSC Vulnerability Management

through documentation (for example if there was no other supplier for said component).	unexpected behaviour.	vendor who document the decision to migrate to the new components detailing comparative evaluations with other models. **3. LLM Platform**: When adopting an LLM fine-tuning library with undocumented optimization techniques, document its unique benefits and conduct tests to evaluate risks against other alternatives. **4. Open-Access LLM Model:** When using a content filtering library with limited documentation to remove harmful outputs, justify the decision with its accuracy benchmarks and document it in the project's wiki.	
7.2.1 In this case, Developers and System Operators **shall** have mitigating controls and undertake a risk assessment linked to such models or components.	See 7.2	***Evaluate and Mitigate Risks for Untrusted Components:*** *Perform a risk assessment to identify potential risks associated with unsecured AI components, specifying, and implementing mitigating controls to minimize vulnerabilities.* **1. Chatbot App:** Conduct input validation and adversarial testing on a poorly documented chatbot API to identify vulnerabilities. Mitigation includes rate limiting, anomaly detection for malicious input patterns, and monitoring for unexpected behaviour. **2. ML Fraud Detection**: Evaluate the new model with serialisation attack scans and adversarial robustness testing; this forms part of its risk assessment that also includes legal and regulatory compliance checks. Mitigation controls include real-time enhanced monitoring with anomaly detection, staggered deployment on low-risk cases in controlled environments for a period, contractual agreement with former members of the model development team to help testing. **3. LLM Platform**: Perform a risk assessment on the fine-tuning library, testing for optimization errors, compatibility issues, and security vulnerabilities. Mitigations include using sandboxed environments for testing and implementing logging to detect anomalies during fine-tuning. **4. Open-Access LLM Model:** Assess risks of the content filtering library, such as failure to flag harmful content or biases in filtering. Mitigation measures include integrating secondary filtering layers, testing for gaps, and monitoring flagged content for refinement.	• NIST SP 800-161 Rev. 1 Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations
7.2.2 System Operators **shall** share this documentation with End-users in an accessible way.	See 7.2	***Share Documentation with End-Users:*** *Share documentation of non-compliant components with end-users, detailing risks, and justifications for transparency.* **1. Chatbot App:** Provide end-users with a summary of the chatbot API's undocumented features, potential risks, and implemented safeguards, along with instructions for reporting anomalies. Ensure accessibility by providing for example, a downloadable accessible PDF File. **2. ML Fraud Detection**: Users included in the staggered deployment are informed with a WCAG 2.1 compliant document on the model choice, reasons, and mitigations and they are invited to report any suspicious system behaviour.	• See References in "*Evaluate and Mitigate Risks for Untrusted Components*" • GOV.UK - Guidance and tools for digital accessibility

		3. LLM Platform: Include documentation in the - compliant with accessibility formats - user guide detailing the fine-tuning library's benefits, risks, and mitigation strategies, and direct end-users to support channels for reporting unexpected outcomes. **4. Open-Access LLM Model:** Document the use of the library in the readme file including the additional monitoring controls customers can use; reference the wiki pages detailing risk assessments and evaluation and how to report issues. Test and leverage repository's accessibility features or generate an accessible PDF.	
7.3 Developers and System Operators **shall** re-run evaluations on released models that they intend on using.	Without reusable evaluations, periodic re-evaluation can be difficult, resulting into performance degradation, bias, inaccuracies, or security vulnerabilities go unnoticed in new releases.	***Create and Maintain Appropriate and Reusable Model Evaluation Suites:*** *Create appropriate model evaluation suites to assess performance, accuracy, security, and potential drift when required.* **1. Chatbot App:** Consult LLM external provider evaluations or create and run your own, focusing on your use-cases running them for major model upgrades. **2. ML Fraud Detection**: Changes in anti-money laundering legislation expand the definition of suspicious transactions to include previously normal patterns (e.g. smaller amounts). The company uses the model evaluation suite to detect any drift. Track performance metrics over time to help setting for triggering re-evaluation or re-training. **3. LLM Platform**: Run the evaluation suite to assess a new model release and identify areas that need mitigations. **4. Open-Access LLM Model:** Runs evaluation suites for major releases and changes to identify areas that need mitigations.	• DecodingTrust: A Comprehensive Assessment of Trustworthiness in GPT Models • AI Safety Institute: Inspect - An open-source framework for large language model evaluations • NIST: AI Test, Evaluation, Validation, and Verification (TEVV) • NIST Dioptra test platform
7.4 System Operators **shall** communicate the intention to update models to End-users in an accessible way prior to models being updated.	Lack of transparency around model updates can make it challenging for users to use a new model version, especially in evaluating data protection compliance. This creates availability and operational issues for end-users relying on certain model feature or behaviour.	***Provide Advance Notice of Model Updates:*** *Notify end-users of model updates at least one month in advance, including any potential impacts on model performance or functionality and offer previews to test changes for at least a month. Ensure notices are accessible.* **1. Chatbot App:** Notify users about updates to the chatbot's third party LLM, highlighting any changes in supported languages or response behaviour, and provide access to a testing sandbox. Notices complies with WCAG 2.1 guidelines. **2. ML Fraud Detection**: Send end-users a notice explaining upcoming changes in the algorithm and how it may impact detection and provides a test instance to allow them run evaluations before switching offering an accessible downloadable PDF. **3. LLM Platform**: Inform end-users – using accessible notices like the ones the previous two examples- about planned finetuning updates, detailing expected improvements or deprecated features, and offer a staging environment and API version for evaluation before deployment. **4. Open-Access LLM Model:** Publish a public notice on the repository's changelog and mailing list detailing upcoming changes, potential impacts, and instructions for testing the new pre-release version. The notice takes advantage of the repositories built-in accessibility features including WCAG 2.1 compliance.	• GOV.UK - Guidance and tools for digital accessibility
Principle 8: Document Your Data, Models, and Prompts			

Provisions	Related threats/risks	Example Measures/Controls	• Reference/Resource
8.1 Developers **shall** document and maintain a clear audit trail of their system design and post-deployment maintenance plans. Developers **should** make the documentation available to the downstream System Operators and Data Custodians.	Without thorough documentation and audit trails, AI system operations may lack transparency, limiting traceability and increasing risks of security gaps, regulatory non-compliance, and operational inefficiencies.	***Develop Comprehensive System Design and Maintenance Documentation****: Document the system design and post-deployment maintenance plans with audit trail of design decisions, architectural diagrams, and maintenance schedules. Ensure the documentation is accessible to downstream System Operators and Data Custodians, highlighting responsibilities, version control, and any dependencies.* **1. Chatbot App:** Document post-deployment monitoring approaches, detailing how user feedback is collected and incorporated into iterative updates. Share these protocols with Data Custodians to ensure they comply with data retention and usage policies. **2. ML Fraud Detection**: Maintain an audit trail of ADRs (Architecture Decision Records) design choices, such as feature selection, model architecture, and testing results. Provide detailed maintenance schedules, including retraining cycles and updates to detection rules. **3. LLM Platform**: Use the approach described in the Fraud Detection example of this control. Additionally, include an audit trail of all model fine-tuning activities and dependencies. **4. Open-Access LLM Model:** Automate log generation from your repository of all an audit trail of all model training/fine-tuning activities and their dependencies, releases, and associated release notes	• ICO: Data Protection Audit Framework Toolkits, Artificial Intelligence • NCSC Machine Learning Principles, 2.3 Manage the full life cycle of models and datasets. NCSC Guidelines for Secure AI System Development, Document your data, models, and prompts. • CSA Companion Guide on Securing AI Systems, Section 2.2.2 Development, Item 2.3
8.1.1 Developers **should** ensure that the document includes security-relevant information, such as the sources of training data (including fine-tuning data and human or other operational feedback), intended scope and limitations, guardrails, retention time, suggested review frequency and potential failure modes.	Lack of detailed security-relevant documentation can lead to unmitigated risks from unverified data sources, misuse of the AI system, and challenges in addressing failure modes.	***Include Relevant Security- Information in System Documentation*** *Document AI systems including intended scope, limitations, known failure modes, prompts, guardrails training data sources, data retention policies, review schedules. Use* ***Model Cards*** *to capture transparent summaries of a model's capabilities, ethical considerations, performance metrics, and limitations, consider using* ***ML Bills of Materials (MLBOMs)*** *to document in machine-readable way the model and its dependencies, with component versions, and licensing.* **1. Chatbot App:** Document prompts and data sources used for RAG, intended use cases, any safeguards to prevent misuse, data retention periods, and failure scenarios. **2. ML Fraud Detection**: Apply the implementation described in the Chatbot App to training/test data. In addition, create and maintain model cards to document the models trained. **3. LLM Platform**: Similar implementation as in the Fraud Detection example. **4. Open-Access LLM Model**: Creates and publish a model card. Use a standard such as OWASP CycloneDX to generate and publish a signed MLBOM documenting model dependencies.	• NCSC Machine Learning Principles, 2.3 Manage the full life cycle of models and datasets. • OWASP CycloneDX – ML BOM
8.1.2 Developers **shall** release cryptographic hashes for model components that are made available to	Absence of cryptographic hashes for model components increases the risk of	***Include Cryptographic Hashes for Models:*** *Release cryptographic hashes for all models made available to stakeholders, enabling verification of component authenticity.* **1. Chatbot App:** Not applicable.	• NIST: Cryptographic Standards and Guidelines • OWASP Cryptographic Storage Cheat Sheet

other stakeholders to allow them to verify the authenticity of the components.	tampering, unauthorized modifications, and integrity issues, compromising trust and security.	**2. ML Fraud Detection**: Provide cryptographic hashes for trained fraud detection models shared with clients, enabling them to confirm the integrity of the received models **3. LLM Platform**: Maintain hashes for each version internally and provide them in direct integration scenarios, e.g. a hosting by a cloud platform acting as a reseller. 4. **Open-Access LLM Model**: Generate and provide unique digital fingerprints (e.g. SHA-256 hashes) for each model version both on model hubs but also your website and Git repository, allowing developers to verify the model's integrity.	
8.2 Where training data has been sourced from publicly available sources, there is a risk that this data might have been poisoned. As discovery of poisoned data is likely to occur after training (if at all), Developers **shall** document how they obtained the public training data, where it came from and how that data is used in the model.	Without comprehensive documentation of training data sources and collection timestamps, organizations may be unable to determine whether their AI models have been affected by data poisoning, fraud, or insider abuse and whether they are compliant to data protection legislation. If data poisoning attacks are revealed to have occurred on public websites, producers or users of models will only know if they are affected if they have robust documentation of what data the model was trained on.	***Document the process of sourcing public training data**, detailing how data was collected, processed, and used in the model (e.g., pretraining, fine-tuning). Include poisoning mitigation measures such as data validation and anomaly detection. Maintain an audit trail to trace datasets if poisoning is suspected.* **1. Chatbot App:** Document any public data used in RAG scenarios. **2. ML Fraud Detection**: Track sources of training data for fraud detection models, logging the origin and preprocessing steps to ensure traceability in the event of anomalies. **3. LLM Platform**: Maintain an audit trail for all training and fine-tuning datasets sourced from public repositories, including documentation, evaluations, and poisoning mitigation measures like validation checks **4. Open-Access LLM Model:** For a community-contributed dataset, enforce logging metadata and manual filtering notes as part of the Pull Request template to enhance transparency and mitigate risks of poisoning.	• ICO: Data Protection Audit Framework Toolkits, Artificial Intelligence • ETSI GR SAI 002 - Data Supply Chain Security • CSA Companion Guide on Securing AI Systems, Section 2.2.2 Development, Item 2.3 • NCSC Machine Learning Principles, 2.3 Manage the full life cycle of models and datasets. NCSC Guidelines for Secure AI System Development, Document your data, models, and prompts.
8.2.1 The documentation of training data **should** include at a minimum the source of the data, such as the URL of	Failure to record detailed metadata, such as data sources and timestamps, can hinder efforts to trace and respond to	***Document metadata for all publicly sourced training data**: Include the exact source (e.g., URLs) and date/time of collection. Ensure this metadata is stored in an accessible format to facilitate traceability in case of reported data poisoning.* **1. Chatbot App:** Maintain a log of URLs and timestamps for all data used in a RAG scenario.	See References in "*Document the process of sourcing public training data"*

the scraped page, and the date/time the data was obtained. This will allow Developers to identify whether a reported data poisoning attack was in their data sets.	data poisoning incidents.	**2. ML Fraud Detection:** Record metadata for public datasets used in feature engineering and training, ensuring URLs and collection dates are stored securely. Include supplier contact details and documentation link for commercial datasets. **3. LLM Platform:** Capture detailed metadata for publicly sourced datasets, including data scrapers used, timestamp logs, and annotations for identified poisoning risks. **4. Open-Access LLM Model**: Provide contributors with a template to submit dataset metadata, requiring URLs and timestamps to be included for every contribution, making required part of the Pull Request template.	
8.3 Developers **should** ensure that they have an audit log of changes to system prompts or other model configuration (including prompts) that affect the underlying working of the systems. Developers **may** make this available to any System Operators and End-Users that have access to the model.	Without an audit log for configuration changes, tracking and understanding modifications to prompts or model configurations become difficult, increasing risks of unintended system behaviour or accountability issues.	***Maintain an Audit Log of Prompt and Model Configuration Changes:*** *Implement an audit log that captures all changes to system prompts and configuration settings, recording details such as change date, user ID, and a description of modifications. Provide access to System Operators and Data Custodians.* **1. Chatbot App:** maintain logs of all prompt changes and LLM API endpoint configurations with the required metadata. Configure alerts to notify administrators when critical prompt templates implementing guardrails are modified. **2. ML Fraud Detection**: Log changes to model parameters and weights and implement alerts when model parameters and weights are modified outside a pipeline deployment. **3. LLM Platform**: Similar approach as in the Fraud Detection example of this control.. **4. Open-Access LLM Model:** Maintain an open audit log that records changes to prompts, model weights, and configurations, specifying contributor IDs and timestamps. Provide stakeholders with read-only access for transparency and enable alerts for critical changes.	See References in "*Document the process of sourcing public training data"*
Principle 9: Conduct appropriate testing and evaluation			
Provisions	**Related threats/risks**	**Example Measures/Controls**	• **Reference/Resource**
9.1 Developers **shall** ensure that all models, applications, and systems that are released have been tested as part of a security assessment process.	Without rigorous security assessments by qualified experts understanding AI and classic risks, AI systems may be vulnerable to exploits, unauthorized access, or data breaches, leading to potential data loss, manipulation, or	***Implement Security Assessment Processes for All Releases****: Establish a mandatory security assessment process for all models, applications, and systems prior to release, covering areas like access control, data integrity, and adversarial AI attacks. Scope testing based on the threats identified in threat models.* **1. Chatbot App:** Test against the OWASP Top 10 for LLM Apps, focusing on the indirect injection risks from PDF uploads identified in the threat model. Add tests to cover general application and platform vulnerabilities ensuring the application has good security posture. **2. ML Fraud Detection:** Test against evasion attacks highlighted in the threat model as well as the security posture of the application, APIs, and platform using the OWASP Top 10 for Apps and APIs. **3. LLM Platform:** Test adversarial robustness, extraction attacks, output sensitivity, ethical violations and bias, data memorisation, and jail breaking.	• ETSI – Security Artificial Intelligence; Security Testing of AI • NCSC Machine Learning Principles, 1.4 Analyse vulnerabilities against inherent ML threats • NCSC Guidelines for Secure AI System Development, Secure Deployment • CSA Companion Guide on Securing AI Systems, Part 2.2.3.

	misuse of the AI model.	**4. Open-Access LLM Model:** Follow the same approach as the LLM Provider. Publish a summary of findings alongside the model to ensure transparency and improve community trust.	Deployment, Item 3 (Release AI systems responsibly) and Annex A – Technical Testing and System Validation • OWASP Top 10 for LLM Applications OWASP AI Exchange • OWASP Top 10 2021 OWASP Top Ten for APIs
		Use Offensive Security Assessments with Penetration Testing and Red Teaming: Use periodic penetration testing to evaluate the AI system's resilience and red teaming for models. **1. Chatbot App:** Expand the annual penetration test to include LLM-specific threats identified in the threat model, such as indirect prompt injections through PDF uploads. **2. ML Fraud Detection**: Include evasion attacks in penetration testing, focusing on adversarial examples designed to bypass detection algorithms. **3. LLM Platform**: Conduct red teaming exercises to evaluate the model's robustness against jailbreaking attempts, bias exploitation, and data memorization risks, leveraging multi-disciplinary teams for comprehensive assessments. **4. Open-Access LLM Model:** For a small organisation with limited resources, combine lightweight internal red teaming and invite experts from the open-source community to test for vulnerabilities, such as toxic content generation or prompt injections.	• Generative AI: Red Teaming Challenge Transparency Report - AI Village Defcon 2024 • NCSC: Penetration Testing • Red-Teaming for Generative AI: Silver Bullet or Security Theater?
9.2 System Operators **shall** conduct testing prior to the system being deployed with support from Developers.	Without pre-deployment testing, systems may fail to meet operational and security requirements, leading to compromised performance, reduced reliability, or security vulnerabilities once deployed.	***Implement Comprehensive Pre-Deployment Testing:*** *Compliment development-time testing with running benchmarking emulation suites covering functional, performance, and security tests to confirm that the system meets intended requirements before deployment. These can either be in-house or independent third-party benchmarks. Integrate these tests into the development pipeline to ensure issues are identified and resolved before release.* **1. Chatbot App:** In the chatbot scenario, unit, integration, and acceptance tests are complimented with tests before deployment on performance under load, functional response accuracy, and resilience against common Top 10 for LLM attack patterns. Tests are automated at CI level. **2. ML Fraud Detection**: In the fraud detection scenario, a similar approach is taken but with emphasis on predictive AI adversarial attacks (e.g. evasion) run against the model. **3. LLM Platform**: For LLM releases, evaluation suits and red team exercises help ensure the model is of acceptable quality and security before deployed. Automate adversarial robustness checks and ethical evaluation frameworks in pipelines to detect biases and regressions in output trustworthiness. **4. Open-Access LLM Model:** Apply the previous community-driven approach to early access models. Integrate tests into automated workflows.	• ETSI – Security Artificial Intelligence; Security Testing of AI • NCSC Machine Learning Principles, 1.4 Analyse vulnerabilities against inherent ML threats • NCSC Guidelines for Secure AI System Development, Secure Deployment • CSA Companion Guide on Securing AI Systems, Part 2.2.3. Deployment, Item 3 (Release AI systems responsibly) and Annex A – Technical Testing and System Validation

			OWASP Top 10 for LLM applications OWASP AI Exchange
9.2.1 For security testing, System Operators and Developers **should** use independent security testers with technical skills relevant to their AI systems.	Without independent security testing, systems may overlook critical vulnerabilities, especially those requiring specialized expertise, increasing the risk of undetected flaws and potential exploitation.	***Engage Independent Security Testers for Pre-Deployment Testing:*** *Use independent security testers with AI expertise to validate security measures, providing an unbiased review of system security.* **1. Chatbot App:** Use external accredited Pen Testers to test the application and API including OWASP Top 10 for LLMs in the scope. **2. ML Fraud Detection**: In addition to the steps described for the Chatbot App, include relevant OWASP AI Exchange items in the scope. **3. LLM Platform**: Add for platform and API security to the steps we described for the Chatbot app and the Fraud Detection examples; employ reputable red team testers to run red team exercises against the model. **4. Open-Access LLM Model:** Leverage the community to crowdsource red team experts for external testing; contract external experts if resources allow to compliment exercises with more rigour.	• NCSC CHECK Penetration Testing • OWASP Top 10 for LLM applications OWASP AI Exchange • OWASP Top Ten for APIs • OWASP Top 10 2021
9.3 Developers **should** ensure that the findings from the testing and evaluation are shared with System Operators, to inform their own testing and evaluation.	Without access to previous testing findings, System Operators may lack critical insights into known vulnerabilities or limitations, and mitigations that may not be seen as adequate by operators leading to potential gaps in security coverage.	***Establish a Process for Sharing Testing Results:*** *Create a standardized process for sharing all relevant testing results and evaluation findings with System Operators of testing reports and ensure timely access for all relevant stakeholders.* **1. Chatbot App: Share:** Share testing results on identified vulnerabilities with System Operators and planned mitigations. **2. ML Fraud Detection**: Share testing findings with System Operators, highlighting a lower rate of rejecting evasion attacks when transactions involve vendors already used by the user legitimately. **3. LLM Platform**: Provide detailed reports on testing outcomes, including risks like data memorization or output bias, and offer guidelines for secure deployment practices **4. Open-Access LLM Model.** Publish a summary of security and robustness findings in community forums, highlighting vulnerabilities like poisoning risks and encouraging feedback for improvements. Provide more detailed reports to customers on request with detailed risks and proposed mitigations.	• ISO/IEC 29119: Software Testing Standards – Communication and Reporting
9.4 Developers **should** evaluate model outputs to ensure they do not allow System Operators or End-users to reverse engineer non-public aspects of the model or the training data.	Without adequate output evaluation, Operators or End-users may reverse engineer model internals or correlate outputs with sequences of predesigned inputs, enabling them to	***Conduct Security Reviews for Output Sensitivity:*** *Engage with AI experts to evaluate model outputs and identify any information that could reveal internal model structures, ensuring sensitive aspects remain non-public.* **1. Chatbot App:** Test chatbot responses for unintended patterns that could reveal prompt templates or system configurations, adjusting output constraints to maintain operational privacy. **2. ML Fraud Detection**: Analyse outputs for patterns revealing specific feature importance (e.g., high fraud scores tied to rare combinations), replacing explicit results with categorical risk levels to obscure decision logic.	• ETSI – Security Artificial Intelligence; Security Testing of AI • ETSI – Security Artificial Intelligence; Privacy OWASP AI Exchange

	reconstruct the model or training data. This can erode commercial advantage, allow the creation of shadow models, or facilitate attacks, even affecting "open source" models if only parts, such as architecture or weights, are released.	**3. LLM Platform**: Test responses to ensure the model does not disclose memorized training data or hint at weight configurations, implementing filters to mask sensitive details. **4. Open-Access LLM Model:** Follow the approach described in the LLM Platform example while harnessing the community and crowd sourcing.	
		Integrate Adversarial Testing and Robustness Evaluation: *Implement adversarial testing to assess the resilience of AI systems against attempts to reverse engineer or manipulate model outputs. This involves simulating attacks that exploit output data to uncover model weaknesses or extract sensitive information. Open-source tools like ART and TextAttack can help you develop and run these tests.* **1. Chatbot App:** Use adversarial testing to simulate prompt injection attacks aimed at eliciting unintended responses. Adjust prompt templates and input validation to reduce vulnerability. **2. ML Fraud Detection:** Test the system against adversarial inputs designed to evade fraud detection, such as fabricated transaction patterns. Refine detection algorithms to improve resilience. **3. LLM Platform:** Simulate jailbreaking attempts to bypass guardrails in the LLM's responses. Update output constraints and enhance filtering mechanisms to prevent such exploits. **4. Open-Access LLM Model:** Leverage community participation to create adversarial tests for input patterns, such as queries that manipulate outputs to infer training data. Apply patches and retrain models to address these vulnerabilities.	• ETSI – Security Artificial Intelligence; Security Testing of AI • CSA Companion Guide on Securing AI Systems, List of AI Testing Tools • Linux Foundation AI & Data Foundation: Adversarial Robustness Toolbox • NCSC Machine Learning Principles, 1.4 Analyse vulnerabilities against inherent ML threats OWASP AI Exchange TextAttack: Generating adversarial examples for NLP models
9.4.1 Additionally, Developers **should** evaluate model outputs to ensure they do not provide System Operators or End-users with unintended influence over the system.	Without evaluating for unintended influence, Operators or End-users may exploit system behaviour to align outputs with personal or malicious goals, leading to bias, misuse, or compromised trust in the system.	***Implement Safeguards Against Manipulation:*** *Set controls to prevent Operators or End-users from adjusting inputs in ways that could intentionally influence the AI system's outcomes.* **1. Chatbot App:** apply prompt engineering and guardrails to constraint prevent output manipulation and abuse the app using prompt injection attacks. **2. ML Fraud Detection**: Limit Operator access to input variables to prevent modifications that could allow compromised insiders to tamper model predictions. **3. LLM Platform**: Implement rate limits and input validation to detect and block repeated adversarial inputs designed to manipulate model responses, ensuring consistent and unbiased outputs. **4. Open-Access LLM Model:** Develop or encourage community source plugins to enforce input constraints, preventing users from crafting queries aimed at exploiting or biasing the model's responses.	• OWASP Top 10 for LLM applications OWASP AI Exchange
Principle 10: Communication and processes associated with End-users and Affected Entities			
Provisions	**Related threats/risks**	**Example Measures/Controls**	• **Reference/Resource**

10.1 System Operators **shall** convey to End-users in an accessible way where and how their data will be used, accessed, and stored (for example, if it is used for model retraining, or reviewed by employees or partners). If the Developer is an external entity, they **shall** provide this information to System Operators.	Insufficient communication about data usage, access, or storage practices can lead to misunderstandings and mistrust among End-users. This lack of transparency increases the risk of misuse or unauthorized access to data, as users may not fully understand or consent to how their data is handled, potentially resulting in regulatory and reputational damage.	***Establish Transparent Data Usage Communication:*** *Provide a transparent overview of data usage policies, purpose of processing, specifying whether data will be used for model retraining, third-party access, or employee review. Each processing activity needs to be logged, and a compatibility assessment needs to be made for each new purpose in accordance with ICO guidance. Ensure end users understand all potential uses of their data and how it contributes to AI model performance or security and that documentation is an accessible format* **1. Chatbot App:** Outline to the end user of the chatbot app how their conversations will be logged, used, and monitored. Include in a downloadable accessible PDF File. **2. ML Fraud Detection**: Explain with clear descriptions in the end-user agreement and privacy policy how customer data will be used for online training and the protections, e.g., anonymisation, in place. Document transparently regions where data is stored. Documentation is WCAG 2.1 compliant **3. LLM Platform**: Explain with clear descriptions in the end-user agreement and privacy policy how customer data will be used for online training and the protections, e.g., anonymisation, in place. Document transparently regions where data is stored. Documentation is WCAG 2.1 compliant. **4. Open-Access LLM Model:** Document how personal data, if any, are processed. Make this information available both as a downloadable accessible PDF file and on a dedicated, well-structured, and accessible HTML web page to ensure broad accessibility.	• ICO : Generative AI second call for evidence: Purpose limitation in the generative AI lifecycle • GOV.UK - Guidance and tools for digital accessibility
10.2 System Operators **shall** provide End-users with accessible guidance to support their use, management, integration, and configuration of AI systems. If the Developer is an external entity, they **shall** provide all necessary information to help System Operators.	Without clear guidance, End-users may mismanage the software, leading to data leaks, vulnerabilities, or system misuse, exposing AI systems to security risks.	***Provide Comprehensive User Guides and Tutorials:*** *Develop and distribute detailed user guides and tutorials that cover secure configuration, integration steps, and recommended usage practices, ensuring users understand safe operation protocols. Provide accessible notification options, such as screen reader-compatible text alerts, and customisable notifications for users with sensory impairments* **1. Chatbot App:** Provide internal system documentation app configurations including LLM API, highlighting security aspects such as secrets management, encryption, access control configuration and so on, with a downloadable accessible PDF File. **2. ML Fraud Detection**: Internal detailed guide on deploying the model and associated services, highlighting security requirements. Documentation is WCAG 2.1 compliant **3. LLM Platform**: Documentation includes step-by-step instructions on integrating the model with existing systems securely, explaining the need for least-privilege access with read-only roles when invoking APIs, use of TLS, secret management for configuration credentials and so on. Documentation is WCAG 2.1 compliant. **4. Open-Access LLM Model:** Like the LLM Platform example, but less detailed. The organisation takes advantage of their Wiki's accessible features to ensure accessibility.	• GOV.UK - Guidance and tools for digital accessibility
10.2.1 System Operators **shall** include guidance on the appropriate use of	Failing to highlight limitations and potential failure modes can lead to	***Highlight Model Limitations and Failure Modes:*** *Clearly outline the model's appropriate uses, limitations, and potential failure modes to inform end-users of scenarios in which the model might produce inaccurate or unreliable outputs.*	• NIST AI RMF

the model or system, which includes highlighting limitations and potential failure modes.	End-users relying on the AI system for unsupported or inappropriate tasks, increasing the risk of operational errors, data misuse, or unintended consequences.	**1. Chatbot App:** Inform users that the chatbot may struggle with complex or ambiguous user queries and encourage fallback to human operators for critical or high-impact scenarios. **2. ML Fraud Detection**: Inform users of scenarios (e.g. unusual market conditions) where the model may produce less accurate predictions, allowing operators to interpret results carefully. **3. LLM Platform**: Document the inherent LLM tendency of hallucinations, providing plausible sounding but incorrect information. Highlight hallucinations in code generation and its effect on creating vulnerable code. **4. Open-Access LLM Model:** Provide a summary of known strengths and weaknesses. Highlight the strength on legal queries and poor performance on other domains such as finance.	
10.2.2 System Operators **shall** proactively inform End-users of any security relevant updates and provide clear explanations in an accessible way.	Without proactive communication about security-relevant updates, End-users may fail to understand changes in system behaviour or associated risks. This lack of awareness can lead to improper use or failure to take necessary precautions, increasing the system's exposure to potential exploitation or security breaches.	***Notify Users of Security Updates***. *Proactively inform End-users about security update, detailing the purpose and impact of each update to promote user compliance. Ensure all users, can be informed by using accessible formats.* **1. Chatbot App**: Provide clear notifications to internal end-users when a security patch is applied to the hiring chatbot that disables Word documents and restricts to PDF files to avoid certain type of attacks. Similarly, notify internal users when e-mailing functionality is disabled following a prompt injection attack abusing the functionality. Explain how it enhances system resilience and inform them of any changes to features or functionality they need to be aware of. The notification should be available as an accessible web page or accessible downloadable PDF file. **2. ML Fraud Detection:** Inform users of a security update applied to the fraud detection system, such as enhanced protections against adversarial evasion attacks. Explain, in a WCAG- 2.1 compliant page, how the update improves the system's ability to detect malicious patterns and reduce false negatives. **3. LLM Platform:** Inform end-users of updates to API functionality, including improvements to safety guardrails, such as enhanced mitigation against hallucinations or bias in outputs. Provide documentation of these updates in accessible formats, such as a changelog or dedicated update page, ensuring the information is easy to understand and actionable and is WCAG 2.1-compliant. **4. Open-Access LLM Model:** Publish detailed release notes in the repository's changelog and mailing list when security-related updates are applied to the model, such as improved robustness against poisoning attacks. Include an accessible summary page and a downloadable accessible PDF.	• GOV.UK - Guidance and tools for digital accessibility
10.3 Developers and System Operators **should** support affected End-users and Affected Entities during and following a cyber security	Failure to support affected End-users and Affected Entities during an incident can result in prolonged recovery times,	***Establish a Documented Incident Support and Communication Process:*** *Develop and document a support process for responding to incidents, covering steps for containment, impact assessment, and recovery, and specify the roles and responsibilities of Developers and System Operators. This should include specialist skills and AI expertise that may require. Provide support through accessible formats, ensuring usability for all affected stakeholders.*	• GOV.UK - Guidance and tools for digital accessibility • NCSC Incident Management, Plan: Your cyber incidence response process

incident to contain and mitigate the impacts of an incident. The process for undertaking this **should** be documented and agreed in contracts with End-users.	mismanagement of containment efforts, and increased reputational damage due to inadequate communication or guidance.	**1. Chatbot App:** Establish a process for assisting End-users who report biased or inappropriate chatbot responses, ensuring NLP/LLM experts are available to investigate the issue. Determine whether the problem stems from model updates, adversarial inputs, or misconfigurations. Provide clear and accessible guidance to End-users on mitigating such issues and share updates as an accessible downloadable PDF file. **2. ML Fraud Detection**: Create a support mechanism to help affected entities (e.g., financial institutions) interpret and respond to a surge in incorrect fraud predictions. Ensure that experts in interpretability techniques such as SHAP and LIME are available to explain the root cause of the issue and guide End-users on adapting thresholds or re-evaluating flagged transactions. Provide all guidance in accessible formats, such as WCAG 2.1-compliant documentation. **3. LLM Platform**: Develop a process for handling incidents such as jailbreaking attempts or misuse of generated content (e.g., deepfakes). Engage AI ethics and adversarial attack experts to assist affected users by providing timely updates, risk mitigation advice, and recovery strategies. Share detailed incident analysis and recovery steps in accessible formats, such as a dedicated incident response webpage with downloadable PDFs. **4. Open-Access LLM Model:** Leverage contributions from the community of researchers and developers to support affected End-users in responding to issues like data poisoning or toxic content generation. Provide detailed incident updates and remediation guidance via accessible Wiki pages, ensuring compliance with accessibility standards. Offer End-users additional support through mailing lists or forums, ensuring all communication is clear and actionable.	NCSC Machine Learning Principles, 4.3 Develop incident and vulnerability management processes
Principle 11: Maintain Regular Security Updates, Patches, and Mitigations			
Provisions	**Related threats/risks**	**Example Measures/Controls**	• **Reference/Resource**
11.1 Developers **shall** provide security updates and patches, where possible, and notify System Operators of the security updates. System Operators **shall** deliver these updates and patches to End-users.	Delayed updates allow attackers to exploit vulnerabilities, increasing the risk of unauthorized access, data breaches, and compromised AI system functionality.	***Implement a Structured Patch Management Process:*** *Establish a structured process for developing, testing, and releasing security patches for AI systems, ensuring regular updates to address known vulnerabilities with user notifications.* **1. Chatbot App:** App packages and containers are patched weekly to mitigate new library vulnerabilities being exploited to stage a breach with e-mails. **2. ML Fraud Detection**: In addition to packages, a patching process for the model helps update the system with security-related fixes, emailing customers of forthcoming updates. **3. LLM Platform**: Like the Fraud Detection example, with email-notifications to API customers. **4. Open-Access LLM Model:** Like Fraud Detection example but different cadence closer to the regular monthly releases and posting updates for new updates on community forums.	• CSA Companion Guide on Securing AI Systems, 2.2.4 Operations and Maintenance • NCSC Incident Management, Plan: Your cyber incidence response process NCSC Machine Learning Principles, 4.3 Develop incident and vulnerability management processes • GOV.UK - Guidance and tools for digital accessibility

		Enable Automatic Updates Where Possible: *Configure AI systems to support automatic security updates, minimizing the risk of delayed patch implementation.* **1. Chatbot App:** the organisation operates automated container image patching combined with regression testing to avoid breaking changes. **2. ML Fraud Detection:** Similar to the implementation in the Chatbot App example, as part of staged rollouts with automated regression tests and rollback mechanisms. **3. LLM Platform:** Similar to the implementation in the Chatbot App example, as part of staged rollouts with automated regression tests and rollback mechanisms. **4. Open-Access LLM Model:** Implement automatic updates for both open-source code and model weights. Use CI/CD tools to pull, test, and deploy new releases, ensuring updates are tested for vulnerabilities and announced on community forums to notify System Operators.	• NCSC Vulnerability Management – Put in a policy to update by default
11.1.1 Developers **shall** have mechanisms and contingency plans to mitigate security risks, particularly in instances where updates cannot be provided for AI systems.	Unaddressed vulnerabilities can lead to exploitation if updates are not feasible, increasing risks of unauthorized access and system disruptions.	***Develop Contingency Plans for Non-Updateable Components****: Establish and document contingency plans, including compensating controls, for components that cannot receive updates due to system limitations or dependencies.* **1. Chatbot App:** Deploy compensating controls such as rate limiting and enhanced logging for an outdated chatbot version integrated with a legacy CRM system that cannot be updated due to compatibility issues. **2. ML Fraud Detection**: An earlier version of the fraud detection model for a specific market segment with some custom escalation rules is running on legacy software platform that cannot be upgraded due to incompatibilities. Instead, compensating controls such as network segmentation to isolate the system from other critical environment and use of Intrusion Detection Systems and enhanced monitoring are added to detect potential exploitations. **3. LLM Platform**: For a legacy deployment of the platform's API gateway that cannot be updated, implement compensating controls like API request filtering, strict authentication mechanisms, and enhanced monitoring and alerting to minimize risks. **4. Open-Access LLM Model:** Recommend compensating controls such as sandboxing to customers using an older version of the model that cannot be updated due to dependency on outdated libraries.	• NCSC Vulnerability Management – The organisation must own the risks of not updating
11.2 Developers **should** treat major AI system updates as though a new version of a model has been developed and therefore undertake a new security testing and evaluation	Inadequate testing of major updates can introduce vulnerabilities and changes in model behaviour, risking data breaches, system manipulation, or	***Conduct Comprehensive Security Testing for Major Updates*** *Perform a security assessment, including penetration testing and model read teaming, for any major AI system updates, treating each update as a new version.* **1. Chatbot App:** When switching to a new major version of the supporting LLM, use automated tests to ensure guardrails continue to be effective. **2. ML Fraud Detection**: use adversarial robustness tests, when performing major retraining of the model with some new datasets to mitigate risks of new vulnerabilities. **3. LLM Platform**: Use pen tests and red teaming to test a major new version with additional API endpoints.	• NCSC Machine Learning Principles, 1.4 Analyse vulnerabilities against inherent ML threats • NCSC Guidelines for Secure AI System Development, Secure Deployment • CSA Companion Guide on Securing AI Systems, Part 2.2.3.

process to help protect users.	unintended behaviour.	**4. Open-Access LLM Model:** Conduct community-driven red teaming exercises for a major update, inviting experts to test for vulnerabilities like poisoning risks or model extraction attacks, and document mitigation actions.	Deployment, Item 3 (Release AI systems responsibly) and Annex A – Technical Testing and System Validation NIST: AI Test, Evaluation, Validation, and Verification (TEVV)
11.3 Developers **should** support System Operators to evaluate and respond to model changes, (for example by providing preview access via beta-testing and versioned APIs).	Without evaluation support, System Operators may overlook risks in updates, leading to potential configuration errors or unmitigated vulnerabilities.	***Offer Preview Access for Major Model Updates*** *Provide System Operators with preview access to major model updates, allowing for evaluation and adjustment to any new system behaviour.* **1. Chatbot App:** Internal System Operators should have access to test environments for new releases. **2. ML Fraud Detection**: Similar to the Chatbot App example and for new model versions. **3. LLM Platform**: The provider offers a quarterly beta program via API versions to help customers evaluate changes before they upgrade their applications. **4. Open-Access LLM Model:** Early beta versions of the model are released regularly as part of a vetted beta program with confidentiality agreements and with instructions to customers' System Operators on how to host and evaluate it.	• NIST AI RMF OWASP AI Exchange

Principle 12: Monitor your system's behaviour			
Provisions	**Related threats/risks**	**Example Measures/Controls**	• **Reference/Resource**
12.1 System Operators **shall** log system and user actions to support security compliance, incident investigations, and vulnerability remediation.	Insufficient logging limits incident investigation and compliance enforcement, weakening the ability to detect and respond to security incidents.	***Implement Comprehensive Logging for Security and Compliance:*** *Establish a logging framework that captures key aspects of system behaviour, including user interactions, access events, data flows, and model outputs, ensuring logs support compliance and security standards.* 1. **Chatbot App**: Log user interactions with the chatbot, including prompt metadata, such as timestamps, session Ids, and response times, to support auditability and investigations. Avoid logging full user prompts or responses unless anonymised or explicitly necessary for troubleshooting and ensure compliance with data protection regulations by implementing data minimisation, retention policies, and secure storage. 2. **ML Fraud Detection**: Log user access attempts, model predictions, and flagged transactions to maintain a robust audit trail for compliance and incident analysis. 3. **LLM Platform**: Log fine-tuning activities, model deployments, and system usage metadata, ensuring sensitive operations are traceable and aligned with security policies. 4. **Open-Access LLM Model**: Maintain logs of public feedback contributions and model training iterations, capturing details such as contributor IDs, timestamps, and flagged training data to improve traceability.	• CSA Companion Guide on Securing AI Systems, 2.2.4 Operations and Maintenance • CISA Joint Cybersecurity Information - Deploying AI Systems Securely • NCSC guidance on Logging for Security Purposes • NCSC Machine Learning Principles, 3.2 Monitor and log user activity

		Ensure Secure Storage and Retention of Logs: *Implement secure storage mechanisms, such as encryption (both at rest and in transit), to protect logs from unauthorized access. Define a retention policy that aligns with compliance obligations and supports security incident investigations. Avoid logging personal data unless strictly necessary; if personal data must be logged, ensure it is anonymized or obfuscated and managed in compliance with applicable privacy laws (e.g., UK GDPR, CCPA). Periodically review and update the retention policy to address evolving compliance and operational requirements.* **1. Chatbot App:** Store chat metadata log (in encrypted storage with access restricted to authorized personnel, retaining logs for one year to comply with operational policies. **2. ML Fraud Detection:** Use an encrypted cloud-based storage solution for transaction logs, retaining them for X years per financial regulations, with secure deletion for older logs. **3. LLM Platform:** Encrypt logs from training and model management systems, retaining them for a defined period based on regulatory and contractual obligations, and operational policies. **4. Open-Access LLM Model:** Securely store logs of training contributions and flagged outputs in an encrypted repository, with RBAC granted to minimum number of people, as needed. Retain the logs for six months to enable community-driven debugging and transparency.	• CISA Joint Cybersecurity Information - Deploying AI Systems Securely • NCSC guidance on Logging for Security Purposes
		Establish Routine Log Analysis for Model Validation: *Define a regular schedule for log analysis to assess model output consistency, detect anomalies, and verify that outputs align with desired outcomes.* **1. Chatbot App:** Regularly analyse session logs to identify trends in user escalations or repeated failed responses, which could indicate prompt misalignment. **2. ML Fraud Detection**: Use of regular log analysis help detect poisoning patterns or drift when online learning is introduced in the fraud detection scenario for certain volatile markets where fraud patterns evolve quickly. **3. LLM Platform**: Perform periodic log reviews to detect anomalies in API usage patterns, such as repeated requests for sensitive topics, which may signal adversarial testing or misuse. **4. Open-Access LLM Model:** Use community feedback to support customer System Operators with log analysis to identify trends in toxic or biased outputs, correlating them with specific input patterns to address potential data poisoning.	• CISA Joint Cybersecurity Information - Deploying AI Systems Securely
12.2 System Operators **should** analyse their logs to ensure that AI models continue to produce desired outputs and to detect anomalies,	Without regular log analysis, security breaches or model issues may go undetected, potentially leading to incorrect outputs	***Implement Alerts for Anomalous Model Behaviour:*** *Set up alerts to notify operators of unexpected behaviour, such as unusual outputs, abnormal input patterns, or significant deviations from historical performance.* **1. Chatbot App:** Configure alerts to flag instances where user query failures exceed a threshold, indicating potential issues with the underlying language model or prompt configurations.	• CISA Joint Cybersecurity Information - Deploying AI Systems Securely

security breaches, or unexpected behaviour over time (such as due to data drift or data poisoning).	or system vulnerabilities	**2. ML Fraud Detection**: For a fraud detection system, implement alerts to notify operators when the model starts flagging an unusually high number of transactions as fraudulent in a short period. The alert triggers a review to investigate whether the issue is due to changes in input data, a model drift event, or potential adversarial activity. **3. LLM Platform**: Set up alerts for sudden spikes in requests targeting specific APIs, especially for sensitive topics or jailbreaking attempts, prompting a review for adversarial testing. **4. Open-Access LLM Model:** Encourage users and System Operators to notify you via e-mail of anomalous behaviour.	
12.3 System Operators and Developers **should** monitor internal states of their AI systems where they feel this could better enable them to address security threats, or to enable future security analytics.	Lack of internal state monitoring can delay detection of security threats or model drift, increasing risks of unauthorised modifications and system failure.	***Implement Monitoring of Key Internal States:*** *Identify and monitor critical internal states of the AI system, such as hidden layers, attention weights, or feature importance, which could provide early indicators of security threats.* **1. Chatbot App:** No action taken since model is operated by another party and existing logging is sufficient. **2. ML Fraud Detection**: Monitor the importance of features like transaction amount, geolocation, or merchant type in the fraud detection scenario; a sudden drop in geolocation importance could indicate drift or tampering by fraudsters adapting their strategies. **3. LLM Platform:** Critical use cases might require introspection of internal model weights, such as where patterns are identified during development and testing that indicate a failure state. **4. Open-Access LLM Model:** The small organisation needs to investigate and understand reported biases. As a result, implements this advanced type of monitoring to track shifts in attention weights during community training contributions to detect how biases are introduced.	• CISA Joint Cybersecurity Information - Deploying AI Systems Securely
		Use Secure Storage for Internal State Data: *Store data from monitored internal states securely, ensuring that sensitive internal metrics are protected from unauthorized access.* **1. Chatbot App:** Encrypt and store user session data and conversation states securely in a database with restricted access. **2. ML Fraud Detection:** The company saves the weights in a protected and isolated storage using a separate dedicated enclave as recommended by the CSI/CISA/NCSC Joint Cybersecurity Information - Deploying AI Systems Securely **3. LLM Provider:** Similar to the implementation in the Fraud Detection example. **4. Open-Access LLM Model:** Use secure storage with access restrictions and encryption for any internal state information that might be generated and used during development.	• CISA Joint Cybersecurity Information - Deploying AI Systems Securely

		Track and Benchmark Model Performance Metrics*: Define and monitor key performance metrics for the AI system, such as statistical accuracy, factual correctness, response time, and error rate establishing benchmarks to detect deviations.* **1. Chatbot App:** Tracking user feedback in the UI escalation rates (conversations requiring human intervention) can reveal gradual degradation of the chatbot app **2. ML Fraud Detection**: Monitoring prediction accuracy ensures the fraud detection model remains within the target range and detects evasion attacks; Usage spikes may indicate extraction or reconnaissance attacks. **3. LLM Platform**: Using automated checks with semantic similarity metrics and trusted datasets, and periodic fact-checking of outputs by LLM developers can detect poisoning. Use of third-party safety APIs and custom classifiers can be used to detect biased, toxic, or inappropriate language in outputs. **4. Open-Access LLM Model:** Monitor community-reported feedback and benchmark outputs against open datasets to detect performance issues such as reduced factual accuracy or increased bias in generated text. Use automated tools to flag significant deviations for review.	• ICO: What do we need to know about accuracy and statistical accuracy?
12.4 System Operators and Developers **should** monitor the performance of their models and system over time so that they can detect sudden or gradual changes in behaviour that could affect security.	Failing to monitor performance over time can hide behavioural shifts, making the system more vulnerable to degradation, attacks, and inconsistencies	***Implement Drift Detection to Identify Behavioural Shifts****: Use drift detection tools to identify shifts in model behaviour due to changing data patterns or environmental factors, allowing proactive responses* **1. Chatbot App:** No action taken, as user prompts are not used for model training. **2. ML Fraud Detection**: Since online training has been adopted, drift monitoring is applied, detecting shifting tactics like the use of VPNs and proxies that decrease the importance of transaction location to mask fraud. Use statistical tests to detect shifts in transaction distributions. **3. LLM Platform**: Concept drift is actively monitored to detect and adapt to changes in language, emojis, and jargon, which could bypass content filters altering model behaviour and be exploited by attackers for jailbreak attempts. This monitoring helps identify emerging vulnerabilities and ensures regular updates to the model's safety guardrails. **4. Open-Access LLM Model:** No action taken since no online training takes place and the model is new.	• CISA Joint Cybersecurity Information - Deploying AI Systems Securely • NCSC Machine Learning Principles, 4.1 Understand and mitigate the risks of using continual learning (CL)
Principle 13: Ensure proper data and model disposal			
Provisions	**Related threats/risks**	**Example Measures/Controls**	**Reference/Resource**
13.1 If a Developer or System Operator decides to transfer or share ownership of training data and/or a model to another	Improper disposal or transfer can lead to unauthorized data recovery, risking breaches, IP loss, and non-compliance	***Develop and Implement a Secure Transfer and Disposal Policy with Data Custodian Oversight:*** *Establish a comprehensive policy to govern the secure transfer and disposal of training data and models. Ensure that Data Custodians oversee all actions, confirming compliance with regulatory standards, protection of intellectual property, and adherence to organizational policies. This includes compliance with UK GDPR when involving personal data.*	• ICO Disposal and deletion ICO: Guidance on AI and data protection • ICO: Retention and destruction of information

entity they **shall** involve Data Custodians and securely dispose of these assets. This will protect AI security issues that may transfer from one AI system instantiation to another.	with data protection laws.	**1. Chatbot App:** Define procedures for securely transferring or deleting fine-tuning datasets used in the chatbot's LLM.. Data Custodians must actively approve all deletions or transfers. **2. ML Fraud Detection:** Develop protocols for securely transferring or deleting training data and models, including sensitive customer transaction records. Require Data Custodian active approval before executing any actions. **3. LLM Platform:** Establish a policy for securely transferring model ownership or licensing agreements, ensuring all fine-tuned and proprietary models are disposed of or transferred in compliance with contractual and regulatory obligations. **4. Open-Access LLM Model:** For open-source contributions, include guidelines for securely deleting intermediate datasets used during training, reviewed by the nominated Data Custodian.	• CSA Companion Guide on Securing AI Systems, 2.2.5 End of life • NCSC Machine Learning Principles, Part 5: End of Life
13.2 If a Developer or System Operators decides to decommission a model and/or system, they **shall** involve Data Custodians and securely delete applicable data and configuration details.	Insecure data deletion during decommissioning may lead to unauthorised access to residual data, increasing regulatory and security risks.	***Implement a Secure Data Deletion Policy with Data Custodian Oversight:*** *Establish a policy for securely deleting data and models during decommissioning, specifying methods compliant with standards. Ensure Data Custodians validate all deletions to maintain regulatory compliance and traceability.* **1. Chatbot App:** Securely delete all conversation logs, prompt data, and configurations when decommissioning the chatbot system. Involve Data Custodians to verify deletion of fine-tuning data accessed via the API. **2. ML Fraud Detection:** Enforce a policy requiring the secure deletion of all historical transaction data, model weights, and configurations during system decommissioning. Data Custodians ensure compliance with financial and privacy regulations. **3. LLM Platform:** Notify customers before decommissioning public LLM services, ensuring all uploaded data is securely deleted after notification periods expire. Require Data Custodians to validate deletion processes. **4. Open-Access LLM Model:** Securely delete all training and test datasets, intermediate model versions, and unreleased models when ceasing development of an open-access LLM. Require that your Data Custodians to verify the process to ensure transparency and compliance.	• See References in "*Develop and Implement a Secure Transfer and Disposal Policy with Data Custodian Oversight*" • NIST SP 800-88 Rev. 1 Guidelines for Media Sanitization • NCSC - Secure sanitisation of storage media