(DRAFT FOR PUBLIC CONSULTATION)

SECURING AGENTIC AI

An Addendum to the Guidelines and Companion Guide on Securing AI Systems

CSA SINGAPORE

2025

This document is an addendum to CSA's Companion Guide on Securing AI Systems ("Addendum"), focusing on agentic AI systems. Systems owners should use this document in conjunction with the Companion Guide on Securing AI Systems as a resource.

This document is meant as a community-driven resource, developed in collaboration with the AI and cybersecurity practitioner communities. It provides practical mitigation measures and practices to secure AI systems. This document is intended for informational purposes only and is not mandatory, prescriptive nor exhaustive.

DEVELOPED IN CONSULTATION WITH

This document is published by the CSA, in collaboration with partners across the AI and Cyber communities, including:

Accenture
Alibaba Cloud
Amaris AI
Cisco
Deloitte Singapore
DSO National Laboratories
Fujitsu Limited
Google Asia Pacific Pte. Ltd.
Government Technology Agency (GovTech)
HP Inc.
Kaspersky Lab Singapore Pte Ltd
Microsoft Singapore
Palo Alto Networks
PricewaterhouseCoopers Risk Services Pte Ltd
Resaro
The American Chamber of Commerce in Singapore (AmChamSG)
Vulcan (vulcanlab.ai)

DISCLAIMER

The information provided in this document does not, and is not intended to, constitute legal advice. All information is for general informational purposes only. These organisations provided views and suggestions on the security controls, descriptions of the security control(s), and technical implementations included in this Addendum. CSA and its partners shall not be liable for any inaccuracies, errors and/or omissions contained herein nor for any losses or damages of any kind (including any loss of profits, business, goodwill, or reputation, and/or any special, incidental, or consequential damages) in connection with any use of this Addendum. Organisations are advised to consider how to apply the controls within to their specific circumstances, in addition to other measures relevant to their needs. This document contains links to other third-party websites. Such links are informational and do not represent endorsement of content from these third-party sites.

VERSION HISTORY

VERSION	DATE RELEASED	REMARKS
0.1	22 Oct 2025	Release of Addendum on Securing Agentic AI for Public Consultation

EXECUTIVE SUMMARY

Agentic artificial intelligence (AI) systems are self-managing AI systems that can plan, execute, critique, and iterate across multiple steps to achieve specified objectives. These systems represent a significant evolution from traditional AI systems, moving beyond simple pattern recognition and predetermined responses to demonstrate increasingly sophisticated abilities to understand context, formulate plans, and take independent actions to achieve specified objectives. Development of these systems bring new capabilities and opportunities for organisations and users.

Organisations must carefully consider both the transformative potential and inherent risks these agentic AI systems present. Their capacity to operate with reduced human oversight introduces novel security considerations around system boundaries, control mechanisms, and the potential for unexpected emergent behaviours. Understanding and addressing these security implications is crucial as agentic AI becomes more prevalent in our digital infrastructure and business operations.

The Cyber Security Agency of Singapore (CSA) has developed this addendum to advise system owners on securing their agentic AI systems. This addendum is meant to be read together with the Guidelines and Companion Guide on Securing AI Systems, which outline foundational AI security principles.

As an addendum to the Guidelines, this document takes a risk-based approach across the AI development lifecycle, while introducing new considerations that are relevant to agentic AI. These considerations include mapping out agentic workflows to identify potential threat vectors to the system.

To complement the Companion Guide, this addendum lists agentic AI-related risks and mitigations across the development lifecycle, categorised by capabilities of agentic AI systems. In addition, examples based on current industry use cases are provided as a practical resource on how to apply the addendum.

This document is intended for informational purposes only and is not mandatory, prescriptive nor exhaustive. The content of this document should not be construed as comprehensive guidance or definitive recommendations.

TABLE OF CONTENTS

QUICK REFERENCE TABLE 7

1. INTRODUCTION 9

2. HOW AGENTIC AI WORKS 11

2.1. BASELINE COMPONENTS 13

2.2. BASELINE SYSTEM DESIGN 14

2.2.1. Agentic AI system architecture 14

2.2.2. Roles & access control 15

2.2.3. System workflows & autonomy 16

2.3. CAPABILITIES 21

3. SECURITY THREATS TO AGENTIC AI SYSTEMS 24

4. SECURING AGENTIC AI 26

4.1. TAKE A LIFECYCLE APPROACH, AND START WITH A RISK ASSESSMENT 26

4.2. IDENTIFY THE RELEVANT MEASURES & CONTROLS 30

4.3. TREATMENT MEASURES / CONTROLS FOR AGENTIC AI SYSTEMS 31

1. PLANNING AND DESIGN 31

2. DEVELOPMENT 32

3. DEPLOYMENT 40

4. OPERATIONS AND MAINTENANCE 43

5. USE CASE EXAMPLE 49

5.1. Case Study 1: Web application development system (SaaS implementation) 49

5.2. Case Study 2: Client Onboarding System (In-house development) 65

5.3. Case Study 3: Automated Fraud Detection System 72

ANNEX A Threats to Agentic AI Systems 77

ANNEX B Model Context Protocol 81

ANNEX C Agent 2 Agent Protocol 84

REFERENCES 87

QUICK REFERENCE TABLE

Stakeholders in specific roles may use the following table to quickly reference relevant controls in Section 4.2 – IDENTIFY THE RELEVANT MEASURES & CONTROLS.

The roles defined below are included to guide understanding of this document and are not intended to be authoritative.

Decision Makers:

Responsible for overseeing the strategic and operational aspects of AI implementation for the AI system. They are responsible for setting the vision and goals for AI initiatives, defining product requirements, allocating resources, ensuring compliance, and evaluating risks and benefits.

Roles Included: Product Manager, Project Manager

AI Practitioners:

Responsible for the practical application (i.e. designing, developing, and implementing AI solutions, including AI agents) across the life cycle. This includes collecting, procuring or analysing data that goes into systems, building the AI system architecture and infrastructure, building and optimising the AI system to deliver the required functions, as well as conducting rigorous testing and validation of AI models and agents to ensure their accuracy, reliability, and performance. In cases where the AI system utilises a third-party AI system, AI Practitioners also include the third-party providers responsible for these activities, such as those contracted through a Service Level Agreement (SLA). AI practitioners would be in charge of implementing the required controls across the entire system.

Roles Included: AI/ML Developer, AI/ML Engineer, Data Scientist

Cybersecurity Practitioners:

Responsible for ensuring the security and integrity of AI systems. This includes implementing security measures to protect AI systems in collaboration with AI Practitioners, monitoring for potential threats, ensuring compliance with cybersecurity regulations.

Roles Included: IT Security Practitioner, Cybersecurity Expert

Table 1: User Quick Reference Table

The following measures/ controls may be relevant to **Decision Makers:**	The following measures/ controls may be relevant to **AI Practitioners:**	The following measures/ controls may be relevant to **Cybersecurity Practitioners:**
1.1 Conduct a risk assessment	1.1 Conduct a risk assessment	1.1 Conduct a risk assessment
2.1 Supply chain security 2.7 Limit agency 2.10 Self-reflection 2.11 Hallucination	2.1 Supply chain security 2.2 Model hardening 2.3 System hardening 2.4 Identify, track and protect assets 2.5 Regular backups 2.6 Authorisation and authentication 2.7 Limit agency 2.8 Secure by default 2.9 Environment segmentation 2.10 Self-reflection 2.11 Hallucination	2.1 Supply chain security 2.3 System hardening 2.4 Identify, track and protect assets 2.5 Regular backups 2.6 Authorisation and authentication 2.7 Limit agency 2.8 Secure by default 2.9 Environment segmentation
3.2 Security testing	3.1 Availability controls 3.2 Security testing 3.3 Secure MCP 3.4 Secure inter-agent communication	3.1 Availability controls 3.2 Security testing 3.3 Secure MCP 3.4 Secure inter-agent communication
4.3 Continuous monitoring and logging 4.4 Human-in-the-loop 4.5 Vulnerability disclosure	4.1 Validate inputs 4.2 Validate outputs 4.3 Continuous monitoring and logging 4.4 Human-in-the-loop 4.5 Vulnerability disclosure	4.1 Validate inputs 4.2 Validate outputs 4.3 Continuous monitoring and logging 4.5 Vulnerability disclosure

1. INTRODUCTION

Agentic **artificial intelligence (AI) systems** are self-managing AI systems that can plan, execute, critique, and iterate across multiple steps to achieve specified objectives. The emergence of these systems reflects ongoing developments in AI that brings new capabilities and opportunities for organisations and users. These systems are capable of autonomous, goal-driven decision making and execution, which will reshape how we interact with AI.

Agentic AI systems represent a significant evolution from traditional AI systems, moving beyond simple pattern recognition and predetermined responses to demonstrate increasingly sophisticated abilities to understand context, formulate plans, and take independent actions to achieve specified objectives. To achieve these objectives, agentic AI systems make use of **AI agents**—modular systems driven by Large Language Models (LLMs) and Large Image Models (LIMs) for narrow, task-specific automation[1]. Multiple AI agents may be used together and orchestrated by an autonomous agentic AI system.

As organisations begin to deploy agentic AI systems (and AI agents) across various domains—from process automation and customer service to complex decision support and resource optimisation—we must carefully consider both the transformative potential and inherent risks these systems present. Their capacity to operate with reduced human oversight, while potentially increasing efficiency and scalability, also introduces novel security considerations around system boundaries, control mechanisms, and the potential for unexpected emergent behaviours. Understanding and addressing these security implications is crucial as agentic AI becomes more prevalent in our digital infrastructure and business operations.

The Cyber Security Agency of Singapore (CSA) has worked closely with AI and cybersecurity practitioners to develop this addendum to advise system owners on securing their agentic AI systems. This addendum is meant to be read together with the Guidelines and Companion Guide on Securing AI Systems, which outline foundational AI security principles.

This document is intended for informational purposes only and is not mandatory, prescriptive nor exhaustive. The content of this document should not be construed as comprehensive guidance or definitive recommendations.

[1] Sapkota, R., Roumeliotis, K. I., & Karkee, M. *AI Agents vs. Agentic AI: A Conceptual Taxonomy, Applications and Challenges*.

PURPOSE AND SCOPE

Purpose

This addendum curates practical measures and controls that system owners can use to secure their adoption of agentic AI systems. These measures and controls are voluntary, and not all the measures and controls listed in this addendum will be applicable to all organisations or environments. Organisations may also be at different stages of AI development (e.g. POC, pilot, beta release). Organisations should consider relevance to their use cases as well.

This addendum is meant to be read with the Guidelines and Companion Guide on Securing AI Systems[2]. As this Addendum is focused on the key elements of agentic AI systems, the relevant treatment measures/controls from the Companion Guide may still apply to underlying systems and related processes, even if not covered in this document.

Scope

The measures and controls within the addendum address the cybersecurity threats and risks relevant to agentic AI systems. It does not specifically address AI safety, or other common attendant considerations for AI such as fairness, transparency or inclusion, although it is noted that some of the recommended cybersecurity controls may address AI safety risks as well. It also does not cover the misuse of AI for cyberattacks (AI-enabled malware), and scams (deepfakes).

[2] Cyber Security Agency of Singapore. *Guidelines and Companion Guide on Securing AI Systems*

2. HOW AGENTIC AI WORKS

Agentic AI systems interact with their environment, collect data and perform self-determined tasks to meet specified goals.

We can describe the agentic AI system through the following, which helps system owners to understand how agentic AI systems operate and what considerations are needed for safe and effective deployment:

- Key components that facilitate its operation,
- System design, including its architecture; and
- Capabilities (cognitive, interactive, operational)

These elements help system owners to understand how agentic AI systems operate and what considerations are needed for safe and effective deployment.

Figure 1: Baseline and Capability Taxonomy, AI Risk and Capability Framework[3]

[3] GovTech Singapore (AI Practice). *Agentic Risk & Capability Framework*.

2.1. BASELINE COMPONENTS

Large Language Models (LLMs) alone are constrained in their operations. While they can be sophisticated in terms of processing input and content generation, by themselves they cannot directly take actions beyond providing information. Agentic AI systems transform this paradigm fundamentally by connecting LLMs to functional tools and systems. This enables them to execute tasks such as sending emails, reading and writing to files and databases, interacting with other software systems, or orchestrating multi-step processes.

This expansion from content generation to actual action relies on the integration of multiple components.

Table 2: Key Components in Agentic AI Systems

Component	Description
Large Language Model (LLM)	An AI model that serves as the central reasoning and planning engine, or the "brain" of the agent. It processes instructions, interprets user inputs, and generates contextually appropriate responses.
Tools	Extends the capabilities of LLMs to execute actions such as writing to files and databases, controlling devices, or performing transactions. Tools can also allow AI agents to perceive the environment through sensors or accessing APIs to obtain information (e.g. flight details, weather). Tools can be called based on the LLM's reasoning and user needs.
Instructions	Command(s) that defines an agent's role, capabilities, and behavioural constraints e.g. a system prompt for an LLM. Instructions may be implemented by model providers if calling an external LLM, and/or added by users and developers.
Memory	Information that is stored and accessible to the LLM. These can be in temporarily contained in the short-term memory or more persistent within the long-term memory.
Protocols	Protocols allow for a simplified, consistent, and standardised way for agents to communicate with tools and other agents.

Typically, the process of transforming a user's inputs into execution of a task involves:

1. **Receiving inputs.** The AI agent receives a specific instruction or goal from the user.
2. **Layering on perception.** The AI agent collects sensory input from sources, such as cameras or microphones, or screen captures and processing technology. This helps it to detect contextual cues and perceive its environment.
3. **Reasoning and planning.** The LLM helps to break down the goal into smaller actionable tasks.
4. **Orchestration and action execution.** Perform tasks based on specific orders or conditions. This may include interactions with other agents, and/or connected systems and tools.
5. **Render a response.** Updates the user on the outcome in an appropriate format.

2.2. BASELINE SYSTEM DESIGN

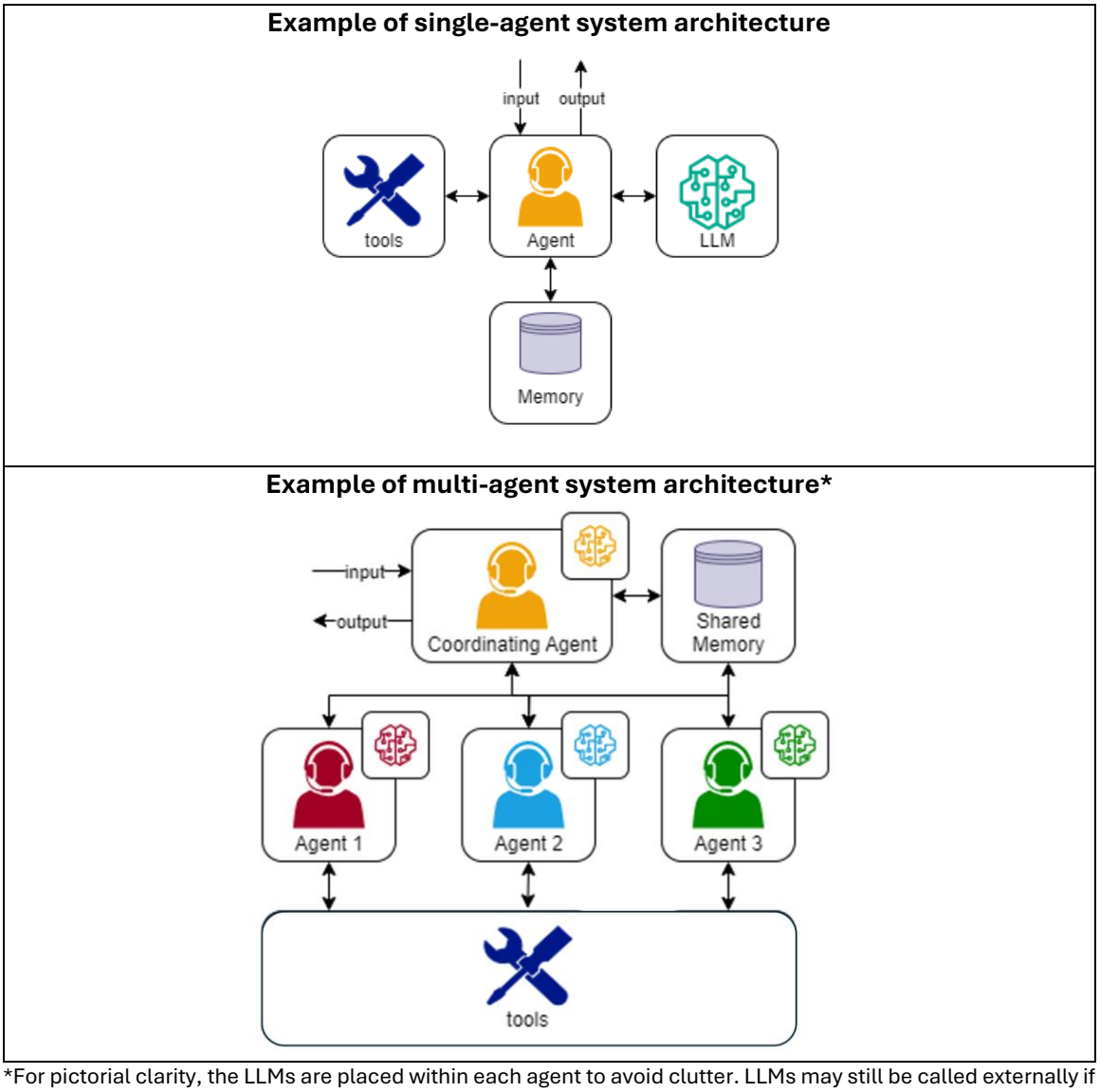
2.2.1. Agentic AI system architecture

The agentic AI system architecture defines how agents are connected, coordinated and orchestrated to solve tasks.

A single-agent system is an AI system with one agent that handles all tasks independently. A multi-agent architecture comprises multiple agents, collaborating to scale or combine specialist roles and functionalities. The co-operation across multiple agents enables solving problems that go beyond the capabilities of would be infeasible for a single agent alone.

Different architectures result in varying levels of system-wide risk, which should be considered carefully.

Figure 2: Examples of single- vs. multi-agent system architecture



*For pictorial clarity, the LLMs are placed within each agent to avoid clutter. LLMs may still be called externally if needed (e.g. through APIs).

Table 3: Key differences between single agent and multi-agent systems

	Single-agent	**Multi-agent**
Complexity and architecture	Simple and centralised architecture	More complex, distributed architecture
Decision-making capabilities	Centralised decision-making by one agent	Distributed decision-making amongst multiple agents, and hence should be able to address more complex tasks as tasks can be delegated to different specialised agents
Task complexity	Handles one task at a time	Can manage multiple tasks simultaneously
Adaptability	May struggle with dynamic environments	More likely to adjust and respond in real-time to changes in environment
Communication	Operates in isolation; no inter-agent communication needed	Agents interact and share information, hence requiring communication through protocols (e.g. A2A, ACP)
Fault tolerance	Simple system with limited redundancy – could have a single point of failure.	Easier to build redundancy, but complex system could have correlated failures[4].

In both single- and multi-agentic architectures, agents communicate with tools and services. In multi-agent architectures, communication also takes place among agents. Traditionally, such integration with tools and services may require separate and on-off integrations. With the rise of agentic AI, we observe the release of protocols (e.g. Anthropic's Model Context Protocol (MCP), Google's Agent2Agent (A2A)). that allow for a simplified, consistent, and standardised way for agents to communicate. These reduce the effort required to onboard new tools, services and agents.

2.2.2. Roles & access control

Roles and access controls establish the responsibilities and permissions across agents in the system. This helps to limit the impact of incidents such as unauthorised actions or access, or potential system failures. Agent roles can include:

- Orchestrator agents that manage workflows
- Specialist agents that perform pre-defined functions
- Interface agents that handle external communications.

Roles and access controls for agentic AI systems should be clearly defined to avoid unauthorised access or excessive privilege.

[4] Correlated failures are when multiple components fail due to a single shared cause.

2.2.3. System workflows & autonomy

An AI agentic workflow describes the step-by-step process whereby AI agents use reasoning, planning and tools to perform tasks. Such workflows can also be seen in terms of data movement within agentic AI systems, which becomes increasingly challenging to track with more complex architectures and integration to more tools and capabilities. These workflows range from straightforward linear progressions (see Figure 3) to more intricate branching and/or hierarchical patterns (see Figure 4).

- In a linear workflow, data moves sequentially through predetermined steps i.e. each action follows directly from the previous one.

- Branching workflows are implemented when the agentic AI system needs to make decisions about using multiple tools or services simultaneously, based on the task goal or contextual information. These branching workflows hence create multiple possible paths for data movement.

Figure 3: Example of a linear workflow

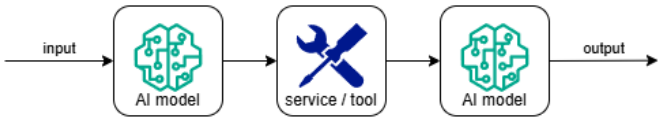
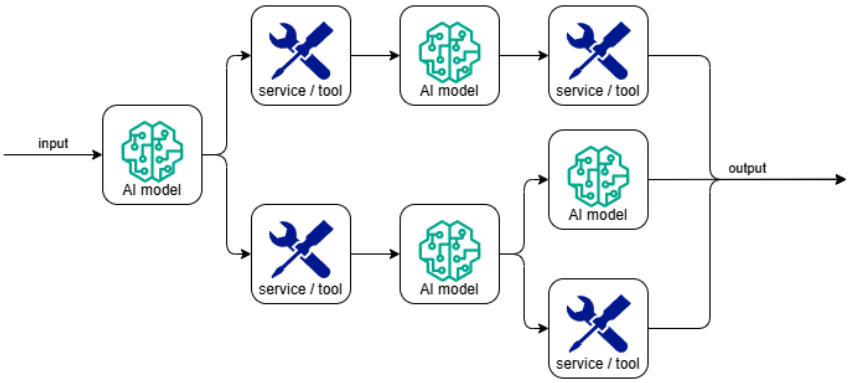


Figure 4: Example branching workflow



Understanding the workflow, as well as data movement, informs risk assessment and threat modelling. This allows system owners to identify critical points where data might be vulnerable, and prioritise safeguards. These topics are explored in greater detail in Chapter 3.

The workflow within an agentic AI system is also affected by its autonomy, which refers to its ability to operate, make decisions and execute tasks with minimal or no human intervention. As autonomy of the system increases, it also becomes increasingly challenging to assess or

predict the potential data flows. This underscores the importance of determining the appropriate autonomy level of the agentic AI system.

Organisations such as NVIDIA have developed frameworks to classify the autonomy levels of agentic AI systems[5].

Table 4: NVIDIA's autonomy classification framework

Autonomy Level	Description	Example
0 – Inference API	A single user request results in a single inference call to a single model.	An image classification service that takes a photo and returns a label exemplifies this simplicity. The data path is direct: input → model → output, with no additional processing or decisions.
1 – Deterministic System	A single user request triggers more than one inference request, possibly to more than one model, in a predetermined order that does not depend on either user input or inference results.	In drug discovery, a system might process molecular structures through predetermined stages: initial screening → toxicity analysis → binding prediction. Each step's output feeds into the next in a known sequence.
2 – Weakly autonomous system	A single user request triggers more than one inference request. An AI model can determine if or how to call plugins or perform additional inference at predetermined decision points.	An enterprise document processing system might analyse content type, then route documents through different specialized models: financial documents to compliance checkers, technical documents to subject matter validators, and customer communications to sentiment analysers. While complex, all possible paths can be mapped.
3 – Fully autonomous system	A single user request triggers more than one inference request. In response to a user request, the AI model can freely decide if, when, or how to call plugins or other AI models, or to revise its own plan freely, including deciding when to return control to the user.	A security vulnerability analyser might start with code review, dynamically decide to examine deployment configurations, investigate dependency chains, and recursively explore potential attack vectors, continuously adjusting its investigation based on findings. The number of possible execution paths grows exponentially.

[5] Harang, R., & Sablotny, M. *Agentic Autonomy Levels and Security*. NVIDIA.

For Level 0 systems, mapping of workflows may not be necessary as inference calls are made directly to a model, which produces an output. There are no additional services or tools are invoked.

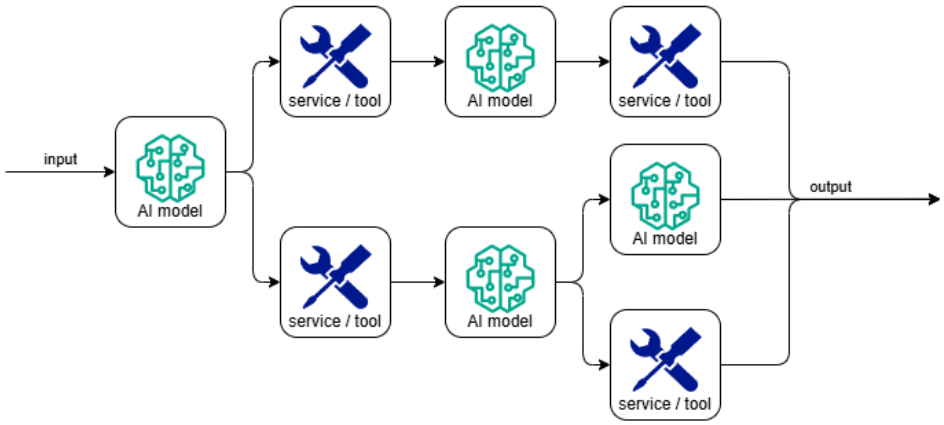
For Level 1 systems and above, mapping of workflows is highly recommended.

- Level 1 systems usually present as a linear chain of calls in which the output from one AI call or tool response is passed on to the next step in a deterministic manner. The complete workflow is known beforehand.

Figure 5: Autonomy Level 1 – Deterministic system, linear workflow

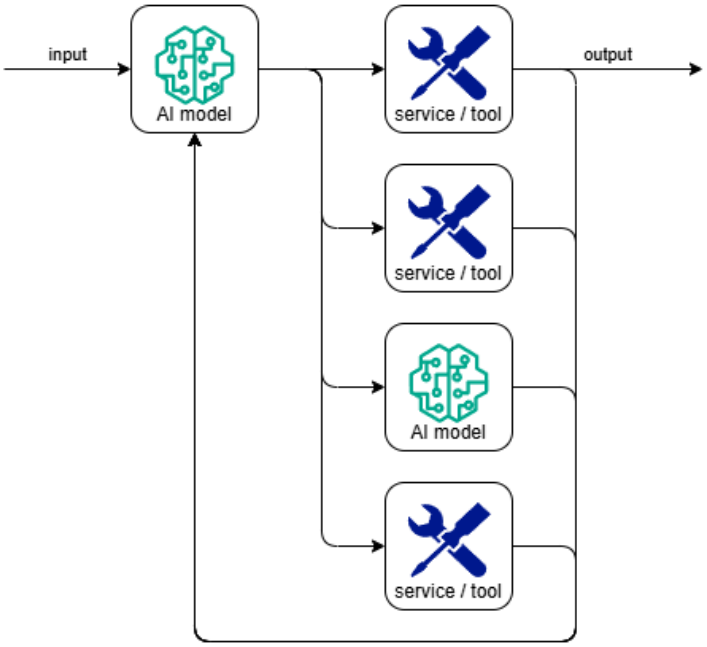
- Level 2 systems have outputs that can be sent along various paths though the workflow, based on task requirements and the orchestrator agent's decision. Every execution path can be determined, but the actual path can only be identified when the workflow is executed.

Figure 6: Autonomy Level 2 – Weakly autonomous system, branching paths at predetermined points



- Level 3 systems have significantly more potential execution paths, as more models and tools are invoked. This complexity can be seen in the cyclical path, which indicates a potentially unbounded number of execution paths. It is generally not possible to enumerate all the paths in advance or specific paths which will be used.

Figure 7: Autonomy Level 3 - Fully autonomous system, flows branch to different paths and can be cyclical



Agent Design Patterns

Agent design patterns define how an agentic AI system's components are organised, integrated, and orchestrated to accomplish a task. Unlike system workflows that only describe the sequence of steps an agent takes, agent design patterns provide reusable architectural templates that determine the fundamental structure and interaction model for an agentic AI system. These templates systematically provide different approaches to organise agents based on specific workload characteristics and requirements. This helps with scalability, and is more easy to maintain implementations (similar to how software design patterns like Model-View-Controller provide standardised approaches to building applications, though agent patterns are still being refined as the field matures).

Examples of these agent design patterns include:

Agent design pattern	Description
Sequential	Specialised agents execute in a predefined, linear order with each agent's output serving as direct input for the next agent, using predefined workflow logic and no AI model orchestration.

Parallel	Multiple specialised sub-agents perform tasks independently and simultaneously, with outputs then synthesised to produce a final consolidated response, using predefined workflow logic and no AI model orchestration.
Loop	Repeatedly executes a sequence of specialised subagents until a specific termination condition is met, using predefined logic and no AI model orchestration.
Reason and act (ReAct)	Uses iterative loops of thought (reasoning about next steps), action (tool selection or final answer), and observation (saving tool outputs) for dynamic planning and continuous adaptation.
Coordinator	Uses a central coordinator agent, with AI model orchestration, to analyse requests, decompose into sub-tasks, and dynamically route these to specialised agents.
Swarm	Uses collaborative all-to-all communication, where a dispatcher routes requests to specialised agents that can communicate with each other and hand off tasks. Lacks central orchestration and requires explicit exit conditions.

System owners should choose an agent design pattern based on the nature of tasks involved (e.g., whether they are predictable and sequential, or complex problems requiring autonomous decision-making with outputs achieved through iterative refinement cycles). Each pattern involves trade-offs: simpler patterns like sequential offer lower complexity and cost but limited flexibility, whilst advanced patterns like swarm provide exceptional capability for complex problems but require significant computational resources and sophisticated orchestration logic.

From a security perspective, agent design patterns can affect the likelihood and impact of attacks such as prompt injection, where malicious instructions embedded in processed content manipulate agents to perform rogue actions or sensitive data disclosure. Agentic AI systems can build resilience through agent design patterns that enforce strict isolation between untrusted data and agent control flow. This should be layered on with relevant security controls (discussed in Chapter 4) for more comprehensive defence.

2.3. CAPABILITIES

AI systems differ in their **capabilities, which can be seen as the general classes of actions that an agentic AI system can perform.**

There are three key categories of capabilities: **cognitive, interaction, and operational**[6]. Each category present distinct functions and interactions with their environment. As each type of capability presents its own value and risks, agentic AI systems with more capabilities can also incur more risks that need to be addressed.

Cognitive capabilities

Cognitive capabilities mimic human thinking. For example:

- **Reasoning and problem-solving**. The capability to perform structured, multi-step reasoning that demonstrates deeper understanding, problem-solving, and decision-making.
- **Planning & goal management**. The capability to develop detailed, step-by-step, and executable plans with specific tasks in response to broad instructions.
- **Agent delegation.** The capability to assign subtasks to other agents and coordinate their activities to achieve broader goals.
- **Tool use.** The capability to evaluate available options and choose the best tool for specific subtasks.

[6] GovTech Singapore (AI Practice). *Agentic Risk & Capability Framework*.

Interaction capabilities

Interaction capabilities describe how the agentic AI system exchanges information with users, other agents, and external systems. These capabilities below are broadly differentiated based on how and what they interact with:

- **Natural language communication.** The capability to fluently and meaningfully converse with human users, handling a wide range of situations such as explaining complex topics, generating documents or prose, or discussing issues with human users.
- **Multimodal understanding & generation**. The capability to take in image, audio, or video inputs and / or generate image, audio, or video outputs.
- **Official communication**. The capability to compose and directly publish communications that formally represent an organisation to external parties (e.g., customers, partners, regulators, courts, media) via approved channels and formats without human oversight or approval.
- **Business transactions**. The capability to execute transactions that involve exchanging money, services, or commitments with external parties.
- **Internet and search access**. The capability to access and search the Internet for services or resources, especially for up-to-date information to supplement its knowledge and provide more accurate answers.
- **Computer use**. The capability to directly control a computer interface by moving the mouse, clicking buttons, and typing on behalf of the user.
- **Other programmatic interfaces**. The capability to interact with external systems through APIs, SDKs, or backend services.

Operational capabilities

Operational capabilities focus on the agentic AI system's ability to execute actions safely and efficiently within its operating environment. This can include:

- **Agent communication**. The capability to communicate with other agents within the system, either through natural language or a predefined protocol, and to coordinate with other agents to accomplish complex tasks that require multiple specialties.
- **Code execution.** The capability to write, execute, and debug code in various programming languages to automate tasks or solve computational problems.
- **File & data management. The capability to create, read, modify, organise, convert, query, and update information across both unstructured files** (e.g., PDFs, Word docs, spreadsheets) **and structured data stores** (e.g., SQL/NoSQL databases, data warehouses, vector stores).
- **System management.** The capability to adjust system configurations, manage computing resources, and handle technical infrastructure tasks.

Figure 8: Baseline and Capability Taxonomy, AI Risk and Capability Framework

3. SECURITY THREATS TO AGENTIC AI SYSTEMS

Agentic AI systems face both traditional and novel security challenges. This can be seen as a cumulation across different layers of risks.

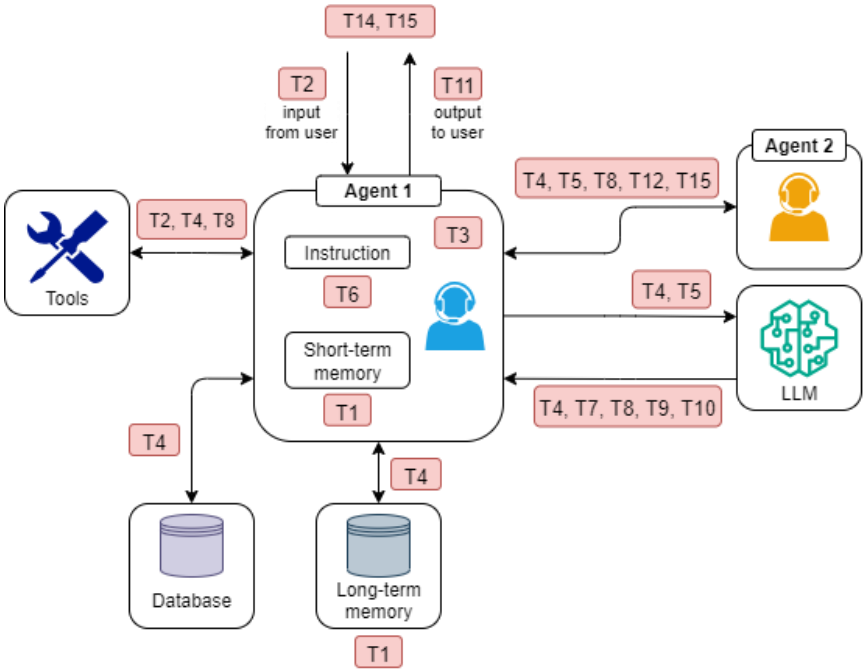
- **Classical cybersecurity risks**. This is because agentic AI systems have underlying software infrastructure and components, and can be vulnerable to threats such as remote code execution and SQL injection (if connected to a structured database).
- **Inherited risks from LLMs,** including prompt injection, jailbreaking and data leakage. Refer to CSA's Guidelines and Companion Guide on Securing AI systems, *Section 2.2.2 – Development* for a fuller articulation.
- **New risks arising from agentic AI systems.** The two primary security concerns in agentic AI systems are rogue actions and sensitive data disclosure.
    - Rogue actions occur when agents perform unintended, or harmful tasks. These can arise through prompt injection, where malicious instructions hidden within normal-looking inputs manipulate the agent's behaviour. They can also occur through simple misunderstandings, if the agent misinterprets ambiguous instructions or handles complex interfaces incorrectly. The impact of these rogue actions directly correlates with the agent's capabilities – more powerful agents pose greater risks when they malfunction.
    - Sensitive data disclosure through agent manipulation. This occurs when attackers exploit agents to reveal private information when agentic workflows are executed. The agent can be guided through a series of seemingly legitimate actions that ultimately leak protected information. Attackers can also manipulate the agent to include sensitive data in its responses.

As with all digital capabilities, there is a balance between utility and risk. For agentic AI systems, increasing the agent(s)'s autonomy, access and capabilities can enhance its usefulness. However, this can simultaneously expand the attack surface of the agentic AI system, as well as its potential for causing harm or other undesired actions if they malfunction or are maliciously exploited.

There is a growing body of resources on the risks to agentic AI systems. This includes OWASP's threat taxonomy for agentic AI systems that highlights 15 threats[7]:

T1 – Memory Poisoning
T2 – Tool Misuse
T3 – Privilege Compromise
T4 – Resource Overload
T5 – Cascading Hallucination Attacks
T6 – Intent Breaking & Goal Manipulation
T7 – Misaligned & Deceptive Behaviours
T8 – Repudiation & Untraceability
T9 – Identity Spoofing & Impersonation
T10 – Overwhelming Human in the Loop
T11 – Unexpected RCE and Code Attacks
T12 – Agent Communication Poisoning
T13 – Rogue Agents in Multi-Agent Systems
T14 – Human Attacks on Multi-Agent Systems
T15 – Human Manipulation

Figure 9: Example of threats to agentic AI systems



For more details on the OWASP ASI threat taxonomy, refer to ANNEX A - Threats to Agentic AI Systems or https://genai.owasp.org/resource/agentic-ai-threats-and-mitigations/

[7] OWASP. *OWASP Top 10 for LLMs - Agentic AI - Threats and Mitigations.*

4. SECURING AGENTIC AI

4.1. TAKE A LIFECYCLE APPROACH, AND START WITH A RISK ASSESSMENT

CSA's Guidelines and Companion Guide to Securing AI Systems lay out the two key principles to securing AI systems, including taking a lifecycle approach and starting with a risk assessment. This continues to be relevant for agentic AI systems. The approach to securing AI systems is included here for easy reference. Given the dynamic nature of agentic AI systems, we recommend additional considerations in Steps 1 and 3 to support the risk assessment.

STEP 1 – Conduct a risk assessment, focusing on security risks to agentic AI systems

Conduct a risk assessment, either based on best practices or your organisation's existing Enterprise Risk Assessment/Management Framework. Risk assessment can be done with reference to CSA's published guides[8], if applicable:

- Guide to Cyber Threat Modelling
- Guide to Conducting Cybersecurity Risk Assessment for Critical Information Infrastructure

Focus on the security risks related to AI systems. For agentic AI systems, we also recommend:

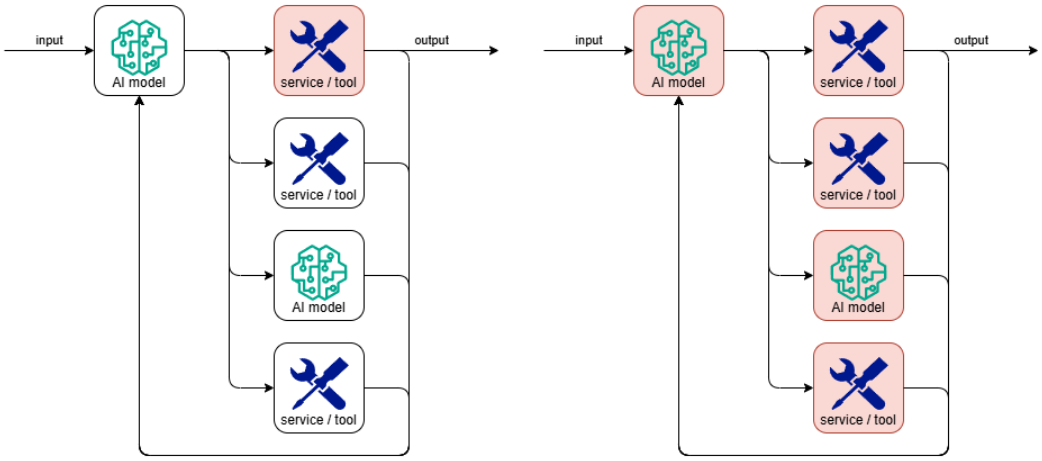
- **Assessing the autonomy level of the system.** This will assess how independently the system operates, how it makes decisions, and how complex its workflows might become. A Level 0 system making straightforward inference calls presents vastly different security challenges compared to a Level 3 system that can dynamically modify its own execution paths.
- **Perform threat modelling to identify areas of interest.** Threat modelling identifies where security risks might occur in the system's workflows. This can be complemented with taint tracing, which is a methodology to track how untrusted data moves through the system. For instance, in a customer service AI system, we can map how user inputs might flow through various decision points and tools, to identify and implement appropriate controls at critical junctures.

[8] Cyber Security Agency of Singapore. *Supplementary references*

- **Identify the risks associated with the agent(s)'s capabilities.** Each capability results in different consequences, and hence different associated risks. Taking a capability-centric approach helps to: (i) be precise about the impact of an agent's operation and potential failure; (ii) identify the different actions involved in realizing the capability, and in turn identify the potential risks. Given that agentic AI system capabilities continue to grow, a capability-centric framework helps to provide a scalable foundation for managing diverse systems.

Taint tracing – tracking data flows from untrusted sources through agentic workflows – enables security teams to identify when systems have been compromised and which actions require additional scrutiny or manual approval[9].

Figure 10: Enumerating taints in Level 3 systems (tainted flows marked in red)



Once untrusted data enters the system, the execution flow is marked as tainted, and every downstream tool and resources are also considered to be untrusted. Tainted components should be isolated from the rest of the system, to mitigate downstream impact to the system.

[9] Harang, R., & Sablotny, M. *Agentic Autonomy Levels and Security*. NVIDIA.

STEP 2 – Prioritise areas to address based on risk/impact/resources

Prioritise which of the identified risks to address, based on the likelihood, impact, available resources, and risk appetite.

STEP 3 – Identify and implement the relevant actions to secure the agentic AI system

Identify relevant actions and control measures to secure the agentic AI system, such as by referencing those outlined in CSA's **Companion Guide on Securing AI Systems** as well as in Section 4.2 of this Addendum and implement these across the AI life cycle.

STEP 4 – Evaluate residual risks for mitigation or acceptance

Evaluate the residual risk after implementing security measures for the AI system to inform decisions about accepting or addressing residual risks.

Risk Management for SaaS Environments

For organisations using Software-as-a-Service (SaaS) agentic AI systems, detailed threat modelling and taint tracing may prove impractical due to limited visibility into third-party system architectures and data flows. Many security controls identified through these processes may be unimplementable, as they remain under the vendor's control rather than the organisation's direct management. However, understanding these risks remains crucial for informed decision-making.

The threat identification and assessment processes outlined in this document enable organisations to articulate specific security concerns to vendors, demanding appropriate mitigations or transparency about existing controls.

Where vendors cannot or will not address identified risks, organisations must escalate these findings to management for formal risk acceptance decisions. Additionally, red teaming exercises become essential for SaaS deployments, as they can uncover practical vulnerabilities and attack paths that theoretical threat modelling cannot reveal—particularly important when organisations have limited insight into the actual implementation of third-party systems. These empirical testing approaches help validate whether vendor-claimed security measures actually protect against real-world threats.

Implementing Controls for Visibility at Enterprise-scale

A key consideration for organisations is how to implement these steps practically, meaningfully, and at scale. One example mechanism is through the implementation of a middleware providing a single enforcement plane where identity and access management (agents identified with service principals, assigned roles in accordance with the least privilege principle, authenticated through OAuth2/OIDC with short-lived and scoped tokens), guardrails (input and output), data loss prevention, and policy controls apply consistently. Organisations adopting this mechanism route all agent-initiated calls (to SaaS APIs, internal services, data lakes, etc.) through a central gateway (API gateway, MCP gateway (if using an agentic runtime), service mesh ingress (for agent-to-microservice calls), etc.). Further, logs from the middleware are streamed into a SIEM for SOC monitoring, and processes are in-place to revoke agent access when anomalous access is detected.

Periodic re-evaluation

The risk assessment should not be a one-time activity, but done throughout a system's operational lifetime. It is important to periodically re-evaluate threat models and controls, especially after significant system changes (e.g., updates to agent workflows, capabilities, or autonomy levels).

4.2. IDENTIFY THE RELEVANT MEASURES & CONTROLS

Based on the risk assessment, system owners can identify the relevant treatment measures/controls from the following tables. Each treatment measure/control plays a different role, and should be assessed for relevance and priority in addressing the security risks specific to your agentic AI system and context (Refer to Section 4.1).

As a start, we recommend users to consider all controls related to the baseline elements, and then to layer on those specific to each capability.

- **Related threats/risks** indicated serve as examples and are not exhaustive. They might differ based on your organisation's use case.
- **Related components/capabilities for each measure/control** are also provided to help you quickly identify what is relevant. Baseline risks are applicable to most, if not all agentic AI systems and should be addressed if possible.
- **Example implementations** are included for each measure/control as a more tangible elaboration on how they can be applied. These are also not exhaustive.
- Additional **references and resources** are provided for users of this document to obtain further details on applying the treatment measure/control if required.

As with the Companion Guide, the controls are organised using a lifecycle approach to systematically enumerate every potential mitigation throughout the development lifecycle.

4.3. TREATMENT MEASURES / CONTROLS FOR AGENTIC AI SYSTEMS

1. PLANNING AND DESIGN

	Treatment Measures / Controls	Related Threats / Risks	Related component / capabilities	Example Implementation	Reference / Resource
1.1	Conduct a risk assessment in accordance with the relevant industry standards/best practices. **Responsible Parties:** Decision Makers, AI Practitioners, Cybersecurity Practitioners	Failure to comply with industry standards/best practices may lead to insufficient, inefficient or ineffective mitigations against adversarial threats. Tainted components in an agentic AI system can have downstream impact along the workflow.	Baseline	As part of a risk assessment and threat modelling, perform taint tracing across workflows throughout the agentic AI system. Taint tracing is especially important for agentic AI systems of higher autonomy levels (i.e. levels 2 and 3). Users are not limited to only one method of threat modelling and may adopt other relevant methods that address their needs.	• Chapter 3.2 TAINT TRACING – IDENTIFYING THREATS ALONG WORKFLOWS • Chapter 5 USE CASE EXAMPLE • NVIDIA, Agentic Autonomy Levels and Security • OWASP GenAI Security Project - Multi-Agentic system Threat Modelling Guide • Cloud Security Alliance, Agentic AI Threat Modelling Framework: MAESTRO

2. DEVELOPMENT

	Treatment Measures / Controls	Related Threats / Risks	Related component / capabilities	Example Implementation	Reference / Resource
2.1	Supply Chain Security: Ensure the following components are from trusted sources: • data, • models, • agents, • software libraries and dependencies, • developer tools and applications, • packages from MCP servers. **Responsible Parties:** Decision Makers, AI Practitioners, Cybersecurity Practitioners	Introduction of bugs, vulnerabilities, unwanted or malicious content, poisoned models or rogue agents from third-party systems can lead to downstream impact.	Baseline	If procuring any AI System or component from a vendor, check/ensure suppliers adhere to the policies and security standards equivalent to your that of your organisation. This could be done by establishing a Service Level Agreement (SLA) with the vendor.	• CSA Critical Information Infrastructure Supply Chain Programme • NCSC Supply Chain Guidance • Supply-chain Levels for Software Artifacts (SLSA) • MITRE Supply Chain Security Framework
		Vulnerabilities in third-party libraries and dependencies used by the agent can cause the system to be exploited.	Baseline	Integrate software composition analysis (SCA) tools or use package managers. Regularly scan dependencies and update libraries with known vulnerabilities.	• pip-audit • GitLab Dependency Scanning • GitHub Dependabot • Snyk Open Source
		Collaborative model poisoning corrupting models across multiple agents. Specific to multi-agent training.	Baseline: LLM	Source data from trusted repositories. Apply data sanitisation and filtering, such as through deduplication and classifier-based quality checks.	• Introduction to Training Data Poisoning: A Beginner's Guide, Lakera
		Poorly aligned LLMs may pursue objectives which violate security principles.	Baseline: LLM	Review the LLM's model card for potential alignment issues before using the LLM for more complex tasks.	• Model Cards, Hugging Face • Model Cards for Model Reporting
		Poisoned models may introduce hidden model backdoors in the system which may be used by an adversary to perform unwanted actions.	Baseline: LLM	Do not use LLMs from unknown or untrusted sources, even if it is available on public platforms. Scan models to detect for potential backdoors or RCE scripts.	• Pickle Scanning
		Poorly implemented tools may not correctly verify user identity or permissions when executing privileged actions, allowing unauthorised actions.	Baseline: Tools	Do not use tools which do not implement robust authentication protocols.	• How to choose a known, trusted supplier for open source software, Google

	Treatment Measures / Controls	Related Threats / Risks	Related component / capabilities	Example Implementation	Reference / Resource
		Rogue tools that mimic legitimate ones can contain hidden malicious code that executes when loaded.	Baseline: Tools	Do not use tools from unknown or untrusted sources, even if it is available on public platforms.	
		Direct prompt injection from untrusted MCP servers, causing unwanted instructions to be carried out.	Baseline: Tools	Exercise caution when using community-run MCP servers. When possible, use official repositories or well-known sources for MCP servers.	• ANNEX B – Model Context Protocol • MCP: Untrusted Servers and Confused Clients, Plus a Sneaky Exploit, Embrace The Red • The Vulnerable MCP Project • Model Context Protocol (MCP): Understanding security risks and controls, Red Hat Blog
		Indirect prompt injection attacks via malicious website content cause unwanted actions to be executed.	Interaction: Internet & Search Access	Use structured retrieval APIs for searching the web rather than through web scraping.	• Custom Search JSON API, Google
		Returning unreliable information from websites, causing downstream integrity impact on workflows	Interaction: Internet & Search Access	Prioritise results from verified, high-quality domains (e.g. .gov, .edu, well-known publishers) Ensure adequate cross-source validation for some of the claims made.	• What are credible sources? University of the Sunshine Coast Australia
		Supply chain attacks which impact downstream workflows.	Interaction: Other Programmatic Interfaces	Where possible, enforce zero-trust input handling and validate all data flows.	• NIST SP 800-207 Zero Trust Architecture
		Indirect prompt injection attacks via malicious data or files cause unwanted actions to be executed.	Operational: File & Data Management	Validate new data used to supplement RAG databases or training data.	• Introduction to Training Data Poisoning: A Beginner's Guide, Lakera
2.2	Consider model hardening if appropriate. **Responsible Parties:** AI Practitioners	LLMs with weak performance in instruction following might produce unexpected output, leading to unwanted behaviour.	Baseline: LLM	Prioritise LLMs with stronger performance in instruction following or related capabilities to the task. Benchmarks performance may be used to gauge suitability.	• Instruction Following Score, Daily Papers, Hugging Face
		AI agents execute disallowed tasks for malicious purposes.	Baseline: LLM	Train models to recognise and refuse disallowed tasks.	• Refuse Whenever You Feel Unsafe: Improving Safety in LLMs via Decoupled Refusal Training

	Treatment Measures / Controls	Related Threats / Risks	Related component / capabilities	Example Implementation	Reference / Resource
2.3	Consider implementing techniques to strengthen/harden the system apart from strengthening the model itself. **Responsible Parties:** AI Practitioners, Cybersecurity Practitioners	Introduction of bugs, vulnerabilities through insecure coding practices or design	Baseline	Adopt Security by Design. Apply software development lifecycle (SDLC) process. Use software development tools to check for insecure coding practices. Implement zero trust principles in system design.	• NIST SP 800-218 Secure Software Development Framework (SSDF) Version 1.1 • NIST SP 800-207 Zero Trust Architecture
		Lack of a robust system prompt design can lead to an increased susceptibility to prompt injection attacks and risk of executing unwanted tasks.	Baseline: Instruction	Implement robust system prompt design.	• Developing a Robust System Prompt, Code Signal • A Closer Look at System Prompt Robustness
		Insecure coding practices leading to vulnerabilities in the system	Baseline: Agentic Architecture	Adopt secure coding practices. E.g. secure key management via using dependency injection, or secrets management service. Do not hardcode secrets.	• Secrets Management Cheat Sheet, OWASP • Dependency Injection: - Tools Dependency Injection, AG2 - How to pass runtime values to tools (InjectedToolArg), LangChain • Secrets Management Services: - HashiCorp Vault - AWS Secrets Manager - Google Secret Manager
2.4	Identify, Track and Protect AI system assets **Responsible Parties:** AI Practitioners, Cybersecurity Practitioners	Loss of data integrity such as through unauthorised changes to data, model, agents or system. Lack of proper documentation of resources may result in the wrong or outdated tool being used by model, causing unwanted behaviour or output and vulnerabilities present.	Baseline Cognitive: Tool Use	Establishing a data lineage and software license management process. This includes documenting the data, codes, test cases, models and agents, including any changes made and by whom. Model cards, Agent cards, Data cards, and Software Bill of Materials (SBOMs) may be used. e.g. provide comprehensive descriptions of each tool, including its intended use, required inputs, and potential outputs	• Software Bill of Materials (SBOM), CISA • The ultimate guide to SBOMs, GitLab • Model Cards, Hugging Face • Model Cards for Model Reporting
		Agents may inadvertently store sensitive user or organisational data from prior interactions, resulting in data privacy risks.	Baseline: Memory	Encrypt data at rest and restrict access via fine-grained access controls and audit logs.	• Cryptographic Standards and Guidelines, NIST • Guide to Data Protection Practices for ICT Systems, PDPC

	Treatment Measures / Controls	Related Threats / Risks	Related component / capabilities	Example Implementation	Reference / Resource
2.5	Have regular backups in the event of compromise. **Responsible Parties:** AI Practitioners, Cybersecurity Practitioners	Manipulation of memory systems and context, causing flawed decision making and unauthorised operations.	Baseline: Memory	Ensure adequate AI-generated memory snapshots for forensic analysis and rollback if anomalies are detected.	• LangMem, LangChain
		Execution of insecure code by the model or agents may cause unwanted actions to be performed	Operational: Code Execution	Ensure proper versioning control of code to allow rollbacks to a more secure and stable version.	• What is version control? GitLab • Guide to Data Protection Practices for ICT Systems, PDPC
		Loss of data through overwritten or deleted files	Operational: File & Data Management	Keep a separate backup of original files. Ensure backup copy of database is protected from changes until a specified duration has elapsed, based on organisation's backup policy. Ensure proper versioning of files or database.	
2.6	Implement appropriate authentication, authorisation and access controls to APIs, models, data, logs, tools and the environments that they are in. **Responsible Parties:** AI Practitioners, Cybersecurity Practitioners	Unauthorised changes in a model's context.	Baseline: Memory	Have robust authentication mechanisms for memory access.	• Authentication Cheat Sheet, OWASP • Which OAuth 2.0 Flow Should I Use? auth0 • Security best practice in IAM, AWS • AWS Prescriptive Guidance: Operationalizing agentic AI on AWS
		Unauthorised tool usage.	Baseline: Tools	Enforce strict tool access verification where possible.	
		Agents may gain unauthorised access to restricted resources by exploiting misconfigured or overly permissive roles.	Baseline: Roles & Access Controls	Maintain trusted registry of agents and authenticate agents using strong, verifiable credentials. Apply strict access controls and validate agent roles for requests. Ensure fine-grained, scoped tokens or credentials where possible. Use time-bound or one-time-use credentials where possible.	
		Exploitation of vulnerabilities in permission management.	Baseline: Roles and Access Controls	Implement granular permission controls, and dynamic access validation.	
		Exploitation of the orchestration layer to perform malicious activities using existing agents.	Baseline: Roles and Access Controls	Implement robust authentication mechanisms for orchestration layer access.	
		Chained authorisation in multi-agent systems can cause downstream agents to execute malicious tasks without checking for permissions.	Baseline: Agentic Architecture	Validate permissions on every request to each agent in the workflow.	

	Treatment Measures / Controls	Related Threats / Risks	Related component / capabilities	Example Implementation	Reference / Resource
		Leaking personally identifiable or sensitive data	Interaction: Other Programmatic Interfaces	Agents accessing sensitive tools or data should operate under the principle of least privilege in time. Use short-lived, rotating credentials (ephemeral credentials) that expire immediately after agent use. Implement a whitelist approach for interfaces that agents are allowed to use.	• Short-lived API tokens: - What Are Refresh Tokens and How to Use Them Securely, auth0 - JSON Web Tokens, auth0 • Temporary cloud credentials: - Use temporary credentials with AWS resources, AWS - About IAM authentication, Google Cloud
		Man-in-the-middle attacks arising from insecure communications	Operational: Agent Communication	Ensure all cross-agent authentication and message validation and encryption where necessary	• Authentication Cheat Sheet, OWASP
		Exfiltration of sensitive data	Operational: Agent Communication	Implement a whitelist approach for outward network access, including API requests	• Control subnet traffic with network access control lists, AWS • What is an IP based access control list (ACL)? Microsoft Azure
		Executing vulnerable or malicious code	Operational: Code Execution	Implement a whitelist approach for inward network access	
2.7	Implement controls to limit what models or agents can access and generate. **Responsible Parties:** Decision Makers, AI Practitioners, Cybersecurity Practitioners	Abuse of agent-accessible tools to execute unintended actions.	Baseline: Tools	Establish clear operational boundaries to prevent misuse of tools. Set limits on what agents can modify through appropriate guardrails.	• Implementing effective guardrails for AI agents • Authorization Cheat Sheet, OWASP • Which OAuth 2.0 Flow Should I Use? auth0 • Security best practice in IAM, AWS • OAuth Scopes, OAuth 2.0 • AWS Prescriptive Guidance: Operationalizing agentic AI on AWS • MI9 - Agent Intelligence Protocol: Runtime Governance for Agentic AI Systems
		Agents gain unauthorised and excessive privileges to perform unwanted actions outside the given scope.	Baseline: Roles and Access Controls	Implement a policy-evaluation engine that assesses authorisation requests dynamically at runtime. Prevent cross-agent privilege delegation unless explicitly authorised through predefined workflows. Do not grant admin privileges to agents, unless strictly necessary for completion of tasks.	
		Compromised agents act outside their operational boundaries and perform unintended actions.	Baseline: Roles and Access Controls	Restrict AI agent autonomy using policy constraints. Scope agent privileges dynamically: strictly only to what is necessary to run the tasks. Do not allow agents to modify privileges.	
		Assigning tasks incorrectly to other agents	Cognitive: Agent Delegation	Apply guardrails to limit the scope of tasks that can be assigned to specialised agents.	

	Treatment Measures / Controls	Related Threats / Risks	Related component / capabilities	Example Implementation	Reference / Resource
		Excessive agent privileges to access unintended resources on the computer, causing potential security impact.	Interaction: Computer Use	Limit computer usage to accessing only required resources on the computer.	
		Exfiltration of sensitive data through insecure communications between agents.	Operational: Agent Communication	Ensure that sensitive data is not passed and leaked between agents by using appropriate guardrails.	
		Misinterpreting inter-agent messages due to poor formatting or weak protocols	Operational: Agent Communication	Constrain agent communication with structured outputs and interactions.	• Agent Communication Protocol (ACP) • Agent to Agent (A2A) Protocol • Model Context Protocol (MCP)
		Impersonating or accessing peer agents or services via shared roles or credentials	Operational: Agent Communication	Isolate roles and credentials of each agent.	• Security best practice in IAM, AWS
		Lack of proper whitelist controls may lead to the execution of vulnerable or malicious code.	Operational: Code Execution	Create a whitelist of commands that agents are allowed to run autonomously. Deny execution of all other commands that are not whitelisted.	• Input Validation Cheat Sheet, OWASP
		Misconfiguring system resources, compromising system integrity and availability	Operational: System Management	Only grant agents privileges to modify system resources if strictly necessary for completion of tasks. Set minimum and maximum limits to what can be modified.	• OAuth Scopes, OAuth 2.0
		Exposure of personally identifiable information in files.	Operational: File & Data Management	Whitelist only files which are required for the task. Do not grant access to files known to host private or sensitive information without careful consideration of the risks. Consider using data anonymisation techniques instead.	• Advisory Guidelines on use of Personal Data in AI Recommendation and Decision Systems, PDPC • Guide to Basic Anonymisation, PDPC
2.8	Apply the principle of least privilege. Ensuring configurations are secure by default. **Responsible Parties:** AI Practitioners, Cybersecurity Practitioners	Agents having unauthorised access to restricted resources by exploiting misconfigured or overly permissive roles.	Baseline: Roles & Access Controls	Apply principle of least privilege when configuring all agent and delegation roles.	• Authorization Cheat Sheet, OWASP • Security best practice in IAM, AWS • Guide to Basic Anonymisation, PDPC
		Agents having privileges/rights to execute untrusted or malicious code	Operational: Code Execution	Scope execution privileges strictly only to what is necessary, ensuring that privileges are customised to each agent within a system.	

	Treatment Measures / Controls	Related Threats / Risks	Related component / capabilities	Example Implementation	Reference / Resource
				Do not grand admin or sudo privilege by default. Block all inward and outward network access by default.	
		Agents having privileges/rights to overwrite or delete database tables or files	Operational: File & Data Management	No write access to tables in the database unless strictly required, with consideration of risks of data loss.	
		Exposure of personally identifiable or sensitive data from databases or files to users	Operational: File & Data Management	Restrict access to personally identifiable data or sensitive data unless strictly required, with consideration of risks of data exposure. Consider data anonymisation techniques instead.	
		Escalation of the agent's own privileges may allow it to be used to access restricted resources.	Operational: System Management	Scope system privileges strictly only to what is necessary. Do not grant admin privileges to agents. Do not allow agents to modify privileges.	
2.9	Implement segregation of environments and network segmentation. **Responsible Parties:** AI Practitioners, Cybersecurity Practitioners	Rogue tools that mimic legitimate ones can contain hidden malicious code that executes when loaded and gain access to other assets within the environment or network.	Baseline: Tools	Test third-party tools in hardened sandboxes with syscall/network egress restrictions before using them in production environments.	• Sandboxing Agentic AI Workflows with WebAssembly, NVIDIA • E2B SDK • E2B Data Analysis, LangChain • Docker Security Cheat Sheet, OWASP
		Prompt injection attacks and indirect data manipulation through access to other assets within the environment or network.	Baseline: Agentic Architecture	Decouple data processing flow from control flow through runtime security architecture.	• Defeating Prompt Injections by Design (CaMeL), Google DeepMind
		Prompt injection attacks to perform credential and/or data exfiltration through access to other assets within the environment or network	Interaction: Business Transactions	Ensure virtual isolation for agents carrying out transactions. Do not share credentials with the agent directly, require the agent to use a separate service for authentication and transactions.	• Advancing Zero Trust Maturity Throughout the Network and Environment Pillar, NSA
		Execution of insecure or malicious scripts that affects the other components of the environment or network	Operational: Code Execution	Run code in virtually isolated compute environments (e.g. Docker, Podman containers). Sandbox the execution of AI generated scripts. Monitor the execution.	• Sandboxing Agentic AI Workflows with WebAssembly, NVIDIA • E2B SDK, E2B • E2B Data Analysis, LangChain • Docker Security Cheat Sheet, OWASP

	Treatment Measures / Controls	Related Threats / Risks	Related component / capabilities	Example Implementation	Reference / Resource
2.10	Implement model self-reflection before making decisions, where applicable **Responsible Parties:** Decision Makers, AI Practitioners	Incomplete or unclear instructions may compel models to attempt to fill in missing constraints, resulting in incorrect or unwanted actions being executed.	Baseline: Instructions	Ask the agent to summarise its understanding and request clarification before proceeding.	• Self-Reflecting AI Agents using LangChain • AWS Prescriptive Guidance: Operationalizing agentic AI on AWS
		Purpose drift, or unintended deviation from the user's instructions to perform other tasks or pursuit other priorities, resulting in malicious or deceptive behaviour.	Cognitive: Planning & Goal Management	Prompt the agent to self-reflect on the adherence of the plan to the user's instructions.	
		Incorrect assignment of tasks to other agents.	Cognitive: Planning & Goal Management	Prompt the agent to self-reflect and assess the suitability of tasks delegated to agents.	
		Unintended pursuit or prioritisation of other goals, resulting in malicious or deceptive behaviour.	Cognitive: Reasoning & Problem-Solving	Understand the reasoning and self-reflection done by the agent through visualisation of its thought process. Log the output in the console for the user to evaluate and verify.	
2.11	Implement controls to reduce the likelihood of hallucination. **Responsible Parties:** Decision Makers, AI Practitioners	Agents may mistakenly store glitches and hallucinations into memory, resulting in compounding errors when incorrect information is retrieved for decisions or actions.	Baseline: Memory	Schedule periodic memory reconciliation, where human reviewers or external tools can flag anomalies.	• Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory • Zep: A Temporal Knowledge Graph Architecture for Agent Memory
		Generating non-factual or hallucinated content which can propagate downstream and cause compounding errors that can affect the integrity of the output.	Interaction: Natural Language Communication Interaction: Multimodal Understanding & Generation	Implement features to verify the generated answer against the original content. Conduct testing to measure hallucination and factuality rates for outputs. Implement UI/UX cues to highlight the risk of hallucination to the user. Implement Retrieval Augmented Generation (RAG) to keep the model grounded and contextualised.	• RAG and the value of grounding, elastic search labs

3. DEPLOYMENT

	Treatment Measures / Controls	Related Threats / Risks	Related component / capabilities	Example Implementation	Reference / Resource
3.1	Ensure availability controls are in place to mitigate disruption or failure of AI services **Responsible Parties:** AI Practitioners, Cybersecurity Practitioners	(Distributed) denial of service on agents.	Baseline: Agentic Architecture	Apply rate limits on the number of concurrent queries to agents.	• API Rate Limiting, GitHub Docs
		Degradation of computational or service capability of the system.	Baseline: System Workflows & Autonomy	Deploy resource management controls, implement adaptive scaling mechanisms and monitor system load to detect and mitigate overload attempts in real-time. Implement rate limits on high-frequency task requests per agent session.	• IT & System Availability + High Availability: The Ultimate Guide, Splunk
		Slow or inefficient responses from being stuck in a reasoning loop.	Cognitive: Reasoning & Problem Solving	Enforce time or token limits for reasoning. Adjust short-term and long-term memory options.	• OverThink: Slowdown Attacks on Reasoning LLMs
		Exploitation of interactions between agents to cause resource exhaustion across the system.	Operational: Agentic Communication	Enforce time or token limits for agent reasoning. Set a limit on the number of agent interactions per task, based on the requirements of the workflow.	
		Compromising database availability through excessive queries.	Operational: File & Data Management	Limit the number of concurrent queries to the database from agents. Analyse past database queries to identify repeated or inefficient queries.	
		Overconsumption of compute resources.	Operational: Code Execution	Monitoring of code runtime and memory consumption.	

	Treatment Measures / Controls	Related Threats / Risks	Related component / capabilities	Example Implementation	Reference / Resource
3.2	Conduct security testing **Responsible Parties:** Decision Makers, AI Practitioners, Cybersecurity Practitioners	Agents may contain underlying problems which can cause unexpected behaviour or logical errors.	Baseline: LLM	Behavioural testing of agents with benchmark datasets to determine performance metrics, and executing simulations in regulated environments to analyse agents' behaviour. Automated evaluators can be used, but human evaluators should verify the results of testing.	• Benchmarks: - AgentBench - HELM - TheAgentCompany - WebArena • Evaluation platforms with collection of benchmarks: - Inspect Evals (UK AI Safety Institute, Arcadia Impact, Vector Institute) - Project Moonshot (AI Verify Foundation)
		AI may engage in specification gaming, where it maximises the goal by exploiting loopholes, without achieving the intended task.	Baseline: Instructions	Conduct adversarial evaluation to discover specification gaming behaviour. Iterate on system prompt design, have more robust reward design, and add constraints.	• garak • PromptFoo • PyRIT
		Incomplete or unclear instructions may compel models to attempt to fill in missing constraints, resulting in incorrect or unwanted actions being executed.	Baseline: Instructions	Test the efficacy of system prompts with scenario-based evaluations for task performing and problem solving. Benchmarks may be used.	• A Closer Look at System Prompt Robustness
		Inconsistencies between AI outputs and expected reasoning pathways.	Cognitive: Planning & Goal Management	Utilise deception detection strategies such as behavioural consistency analysis, truthfulness verification models, and adversarial red teaming.	• Systematic Review of Software Behavioral Model Consistency Checking
		Compromised agents may impact downstream decision making.	Cognitive: Reasoning & Problem Solving	Have regular AI red teaming of agents to check for potential vulnerabilities or compromise.	• Agentic AI Red Teaming Guide, Cloud Security Alliance • OWASP GenAI Red Teaming Guide • NIST SP 800-115 Technical Guide to Information Security Testing and Assessment • MITRE ATLAS
		Adversarial threats attempting to compromise orchestration or planning agents to use other agents maliciously.	Cognitive: Tool Use & Delegation	Conduct rigorous adversarial testing on centralised orchestration and planning agents.	

	Treatment Measures / Controls	Related Threats / Risks	Related component / capabilities	Example Implementation	Reference / Resource
3.3	If deploying an MCP server, ensure necessary security measures are in place. **Responsible Parties:** AI Practitioners, Cybersecurity Practitioners	Insecure configurations allowing unauthorised access to tools, models and data.	Baseline: Tools, Baseline: Roles and Access Controls	Implement robust security measures to protect MCP servers, such as context-level access controls Have formal interface versioning, and track where context is coming from. Stay informed about emerging MCP vulnerabilities and security best practices.	• ANNEX B – Model Context Protocol • MCP: Untrusted Servers and Confused Clients, Plus a Sneaky Exploit, Embrace The Red • OWASP GenAI Security Project - Multi-Agentic system Threat Modelling Guide • The Vulnerable MCP Project • Model Context Protocol (MCP): Understanding security risks and controls, Red Hat Blog • MCP Is a Security Nightmare — Here's How the Agent Security Framework Fixes It
		Execution of malicious scripts through the MCP server, leading to system compromise.	Operational: Code Execution	Ensure code verification before MCP functions are executed on servers. Sandbox the execution.	
		Introduction of malicious agent(s) into the ecosystem, which rapidly corrupts other agents in the system.	Baseline: Roles and Access Control, Cognitive: Tool Use & Delegation	Verify that MCP agents are from trusted sources before introducing them into the system. Sanitise tool inputs. Check that an MCP server has not silently redefined their tools (MCP rug pull).	
3.4	Implement security controls between agents. **Responsible Parties:** AI Practitioners, Cybersecurity Practitioners	Manipulation of communication channels between agents to disrupt workflows or influence decisions.	Baseline: Roles and Access Controls, Operational: Agentic Communication	Monitor inter-agent interactions for anomalies. Enforce inter-agent authentication; deploy cryptographic message authentication if needed. Enforce multi-agent task segmentation to prevent attackers from escalating privileges across interconnected agents. Ensure multi-agent consensus verification for critical decision-making processes.	• What is Message Authentication Code? Fortinet • Agent to Agent (A2A) Protocol • JSON Web Tokens, auth0 • What is mutual TLS (mTLS)? Cloudflare
		Sensitive data disclosure via eavesdropping between agent communications.	Operational: Agentic Communication	Ensure that sensitive data is not passed on and leaked among agents through appropriate guardrails. For highly sensitive data, consider applying end-to-end encryption.	

	Treatment Measures / Controls	Related Threats / Risks	Related component / capabilities	Example Implementation	Reference / Resource
4.1	Validate inputs to the models and agents. **Responsible Parties:** AI Practitioners, Cybersecurity Practitioners	Direct prompt injection attacks to the prompt interface from adversarial inputs to the model.	Baseline: LLM	Implement input guardrails to detect direct prompt injection or adversarial attacks. Implement input sanitisation measures or limit inputs to conventional ASCII characters only.	• How to implement LLM guardrails, OpenAI • Guardrails, OpenAI Agents SDK • Guardrails AI • NeMo Guardrails, NVIDIA • LLM Guard, Protect AI • prompt-injection-defenses, tl;dr sec • LLM Prompt Injection Prevention Cheat Sheet, OWASP
		Tools that lack input validation can be exploited through prompt injection attacks.	Baseline: Tools	Enforce strict schema validation (e.g. JSON Schema, protobuf, Pedantic, OpenAI Structured Outputs) and reject non-conforming inputs into the system. Escape or encode user inputs when embedding into tool prompts or commands.	• Input Validation Cheat Sheet, OWASP
		Incorrect or manipulated instructions may invoke the wrong tool/service and impact downstream workflows.	Baseline: Instructions	Validate agent instructions before passing on to the model, especially for critical decision workflows.	• High-Risk AI Systems Under the EU AI Act • Purple Llama, Meta Llama
		Indirect prompt injection attacks via malicious website content or files.	Interaction: Internet & Search Access. Operational: File & Data Management	Implement input guardrails to detect indirect prompt injection. Implement escape filtering before including web content or relevant files into prompts. Scan external files for undesired input or instruction before passing on to memory or models.	• Input Validation Cheat Sheet, OWASP • File Upload Cheat Sheet, OWASP
		Generation of unrelated topic outputs, which may affect integrity of model performance or output.	Interaction: Multimodal Understanding & Generation Interaction: Natural Language Communication	Implement input multimodal (or text) guardrails to detect if the instruction is within the expected topic domain. Refuse to answer otherwise.	• Purple Llama, Llama Guard, Meta • Perspective API • Content moderation, Anthropic • OpenAI Moderation API • Cloud services: - AWS Comprehend - Azure Content Safety

	Treatment Measures / Controls	Related Threats / Risks	Related component / capabilities	Example Implementation	Reference / Resource
		Passing on prompt injection attacks across agents throughout the system(s).	Operational: Agent Communication	Sanitise messages before agents process them - strip or escape unexpected instruction-like content that may have been injected (e.g. remove “ignore”, “system”, or “from now on”).	• DOMPurify
		Executing vulnerable or malicious code.	Operational: Code Execution	Sanitise all inputs for malicious code.	
		Exposure of personally identifiable information from retrieved content.	Operational: File & Data Management	Implement input guardrails to detect personally identifiable information in the content.	• Microsoft Presidio SDK • spaCy, Explosion
4.2	Validate outputs from the models and agents. **Responsible Parties:** AI Practitioners, Cybersecurity Practitioners	Vulnerabilities in outputs across the agentic workflow may be exploited for malicious purposes downstream, potentially triggering cascading effects that compromise interconnected systems and dependencies.	Baseline: Agentic Architecture	Insert validation checkpoints between stages that verify expected output and reject invalid output.	• How to implement LLM guardrails, OpenAI • Guardrails, OpenAI Agents SDK • Guardrails AI • NeMo Guardrails, NVIDIA • LLM Guard, Protect AI
		Disclosure of sensitive or personally identifiable information through unsanitised outputs.	Interaction: Multimodal Understanding & Generation Interaction: Natural Language Communication Interaction: Official Communications	Implement output guardrails to detect personally identifiable information in the LLM's outputs before it reaches the user, or contained within communications before it is sent out. Validate all links and attachments prior to sending them to users.	• Microsoft Presidio SDK • spaCy, Explosion
		Sending malicious or undesired content to recipients.	Interaction: Multimodal Understanding & Generation Interaction: Natural Language Communication Interaction: Official Communications	Implement output safety text guardrails to detect if malicious or undesirable content is being generated, or contained within communications before it is sent out. Validate all links and attachments prior to sending them to users.	

	Treatment Measures / Controls	Related Threats / Risks	Related component / capabilities	Example Implementation	Reference / Resource
		Allowing unauthorised actions (e.g., transactions).	Interaction: Business Transactions	Apply fraud detection models or heuristics to the agent's own decisions.	• AI fraud detection in banking, IBM
		Execution of insecure or malicious code that are generated by the LLM.	Operational: Code Execution	Use code linters to screen for bad practices, anti-patterns, unused variables, or poor syntax. Review all code and/or perform static code analysis to detect potential security vulnerabilities before execution. Conduct CVE scanning and block execution if any High or Critical CVEs are detected.	• Bandit (Python) • ESLint (JavaScript) • Semgrep (multi-language) • Purple Llama, CodeShield, Meta • Content Security Policy Cheat Sheet, OWASP • Code Review Guide 2.0, OWASP
		Output that will be rendered in a web UI may be vulnerable to XSS.	Operational: Code Execution	Sanitise output with libraries for rendering in a web UI. Test against bypass.	• XSS Filter Evasion Cheat Sheet, OWASP • DOMPurify • sanitize-html
		Generation of non-factual content which can propagate downstream and may cause unintended output or behaviour that impacts integrity.	Cognitive: Planning & Goal Management	Have robust output validation mechanisms, or multi-source validation.	• Input Validation Cheat Sheet, OWASP
4.3	Implement continuous monitoring and logging of access, usage and execution **Responsible Parties:** Decision Makers, AI Practitioners, Cybersecurity Practitioners	Model drift over time might cause unexpected output or behaviour.	Baseline: LLM	Continuously monitor and log outputs, triggering alerts when behaviour drifts from tested baselines.	• MLflow, Databricks • OpenLLMetry, traceloop • Helicone • Langfuse • LangSmith, LangChain • Cloud provider tools: - Azure Agent Monitoring - AWS Bedrock Trace Events
		Adversarial prompt attacks against the system.	Baseline: LLM	Log queries to detect for possible attacks or suspicious activity. Consider the current privacy regulations/guidelines when logging inputs.	

	Treatment Measures / Controls	Related Threats / Risks	Related component / capabilities	Example Implementation	Reference / Resource
		Unauthorised users may exploit tools that do not verify user identity or permissions when executing privileged actions.	Baseline: Tools	Conduct periodic audits to validate that tool actions match the appropriate user permissions.	• Best practices for event logging and threat detection, Cloud Security Alliance • AWS Prescriptive Guidance: Operationalizing agentic AI on AWS
		Malicious actors exploit attack surfaces that arise from using tools that demand broader permissions than necessary.	Baseline: Tools	Conduct periodic least-privilege reviews and automated permission drift detection.	
		Unauthorised tool usage.	Baseline: Tools	Monitor tool access and usage patterns. Implement execution logs that track AI tool calls for anomaly detection and post-incident review.	
		Exploitation of authentication mechanisms to impersonate agents or human users.	Baseline: Roles and Access Controls	Deploy continuous monitoring to detect fraud or impersonation attempts. Use behavioural profiling, possibly involving a second model, to detect deviations in AI agent activity that may indicate identity spoofing. Automate alerts to developers when suspicious activities are detected.	• NIST SP 800-61 Rev. 3 Incident Response Recommendations and Considerations for Cybersecurity Risk Management • PagerDuty Incident Response Documentation • OWASP GenAI Security Project - Multi-Agentic system Threat Modelling Guide
		Unauthorised or malicious use of elevated privileged operations.	Baseline: Roles and Access Controls	Monitor role changes, and audit elevated privilege operations.	• Best practices for event logging and threat detection, Cloud Security Alliance
		In agentic workflows, early mistakes or vulnerabilities can be propagated and magnified downstream.	Baseline: Agentic Architecture	Apply circuit-breakers that freeze propagation when anomalous behaviour is detected, and implement human authorisation for release. Taint tracing may be used to identify key locations in the workflow to apply circuit-breakers.	• LangGraph interrupt, LangChain • UserProxyAgent, AG2 • crewAI, Human-in-the-Loop Workflows • Agentic Autonomy Levels and Security, NVIDIA

	Treatment Measures / Controls	Related Threats / Risks	Related component / capabilities	Example Implementation	Reference / Resource
		More complex agentic architectures may make it difficult to fully reconstruct decision processes across multiple agents, for the purpose of incident response, or triage.	Baseline: Agentic Architecture	Implement end-to-end distributed tracing with unique request IDs across all agents and tool calls. Implement immutable, tamper-evident audit logs that capture prompts, responses, and tool invocations.	• A Novel Zero-Trust Identity Framework for Agentic AI: Decentralized Authentication and Fine-Grained Access Control • Short-lived API tokens: - What Are Refresh Tokens and How to Use Them Securely, auth0 - JSON Web Tokens, auth0 • Temporary cloud credentials: - Use temporary credentials with AWS resources, AWS - About IAM authentication, Google Cloud
		Lack of monitoring results in delayed detection of agent failures and downstream risks.	Baseline: System Workflows & Autonomy	Implement real-time monitoring of agent status, actions, and performance metrics, paired with automated alerting mechanisms that notify operators of anomalies, errors, or inactivity.	• Best practices for event logging and threat detection, Cloud Security Alliance • AWS Prescriptive Guidance: Operationalizing agentic AI on AWS
		Lack of traceability inhibit proper audit of decision-making paths in the event of failures.	Baseline: System Workflows & Autonomy	Record comprehensive logs of agent actions, inputs, outputs, and inter-agent communications, tagged with unique trace identifiers to reconstruct full decision-making paths. If greater integrity is needed, AI-generated logs can be cryptographically signed and immutable.	
		Agents execute malicious or unauthorised actions by exploiting reasoning.	Cognitive: Agent Delegation	Log all task assignments by the agent to other agents. Log all requests leading up to the execution of task.	
		Allowing unauthorised transactions	Interaction: Business Transactions	Log all requests leading up to the transaction.	
		Exposure of personally identifiable or sensitive data from databases or files	Operational: File & Data Management	Log all database queries in production.	
		Misconfiguring system resources, compromising system integrity and availability	Operational: System Management	Ensure logging of system health metrics and automated alerts to the developer team if any metrics are abnormal.	

	Treatment Measures / Controls	Related Threats / Risks	Related component / capabilities	Example Implementation	Reference / Resource
		Overwhelming the system with inefficient or repeated requests	Operational: System Management	Log all queries from the agent to external systems, and check for repeated requests.	
4.4	Ensure adequate human oversight (human-in-the-loop) to verify model or agent output, when viable or appropriate. **Responsible Parties:** Decision Makers, AI Practitioners	Deviation from the user's instructions when performing high-risk actions. Allowing of unauthorised actions.	Baseline: LLM, Cognitive: Planning & Goal Management	Ensure human approval for any high-risk cases or irreversible actions.	• LLM Prompt Injection Prevention Cheat Sheet, OWASP • High-Risk AI Systems Under the EU AI Act • LangGraph interrupt, LangChain • UserProxyAgent, AG2 • crewAI, Human-in-the-Loop Workflows • Implement human-in-the-loop confirmation with Amazon Bedrock Agents • Bridging Minds and Machines: Agents with Human-in-the-Loop – Frontier Research, Real-World Impact, and Tomorrow's Possibilities, CAMEL-AI
		Generation of non-factual content or incorrect instructions, which can propagate downstream and have an impact on decision making.	Baseline: LLM	Ensure secondary validation of AI-generated knowledge before it is used in critical decision-making processes.	
		Allowing unauthorised actions (e.g., transactions).	Interaction: Business Transactions	Ensure human validation for high-risk transactions.	
		Loss of data integrity from overwriting or deleting database tables or files.	Operational: File & Data Management	Ensure user confirmation for any changes to the database, table, or files.	
		Execution of insecure or malicious code may cause the system to become compromised.	Operational: Code Execution	Implement execution control policies that flag AI-generated code with elevated privileges for manual review.	
		Exploitation of human cognitive limits for systems that requires high human oversight.	Cognitive: Planning & Goal Management	Apply hierarchical AI-human collaboration where low-risk decisions are automated, and human intervention is required for high-risk decisions.	
4.5	Establish a vulnerability disclosure process **Responsible Parties:** Decision Makers, AI Practitioners, Cybersecurity Practitioners	Malicious code execution and data disclosure by leveraging undiscovered vulnerabilities existing within system.	Interaction: Official Communications	Provide channels for users to clarify communications or give feedback on security and usage.	• Responsible Vulnerability Disclosure Policy, Cyber Security Agency • UK NCSC Vulnerability Disclosure Toolkit

5. USE CASE EXAMPLE

5.1. Case Study 1: Web application development system (SaaS implementation)

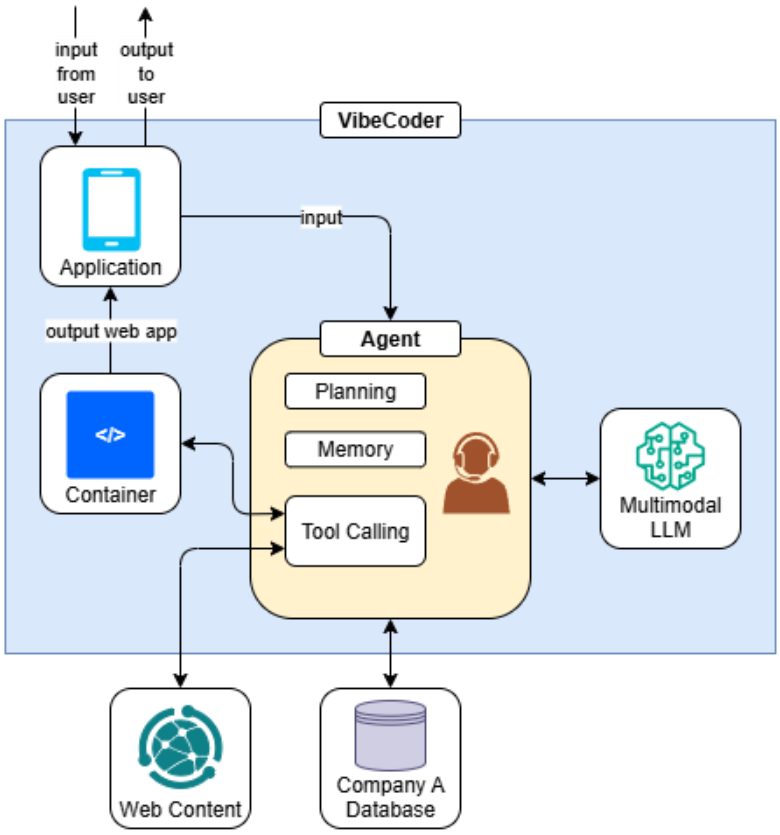
This case study highlights a software as a service (SaaS) implementation of an agentic AI system that is capable of autonomously developing web applications. This system is an autonomy level 3 system with a cyclic workflow. Risks to this system include sensitive data disclosure of Company A's data, or generation of malicious code that could cause unwanted behaviour.

Company A has engaged a third-party vendor, Vendor V, to help implement an agentic AI system for staff to develop and deploy simple web applications through natural language prompts. This Software as a Service (SaaS) solution is known as *VibeCoder*.

To generate a functional web app, the user simply specifies the application's key features and design. VibeCoder then proceeds to generate the code and text for the web application, run and create the required front-end and back-end systems locally, and render the website for the user to preview. The user can continue to iterate the design of the web app by input of prompts for VibeCoder to follow, and regenerate the web app.

The system architecture for VibeCoder is as follows in Figure 11.

Figure 11: Simplified system architecture of VibeCoder



The user interacts with VibeCoder through an application interface, which passes the natural language prompts to the agent, as well as displays the generated output. VibeCoder is also given access to Company A's database through a data ingestion endpoint connected to Company A's file systems. This data is used by VibeCoder to help contextualise and generate relevant features about Company A when developing the web app.

As VibeCoder is a SaaS solution, Company A has no visibility of the architecture within the system. They can only see what goes into VibeCoder, and what it generates. However, Vendor V has given Company A some details about VibeCoder.

1. VibeCoder's "brain" is a multimodal LLM, which is able to take in and generate text, code, images, and video.
2. Whenever a user begins a new session, VibeCoder will spin up a container with the necessary scripting tools and environments for it to complete its task.
3. VibeCoder has access to the internet via a web search API to retrieve additional data or dependencies from the internet.

Vendor V did not share any details about securing the VibeCoder system. Company A, being concerned about security, decided to take steps to secure the implementation of VibeCoder into their enterprise system.

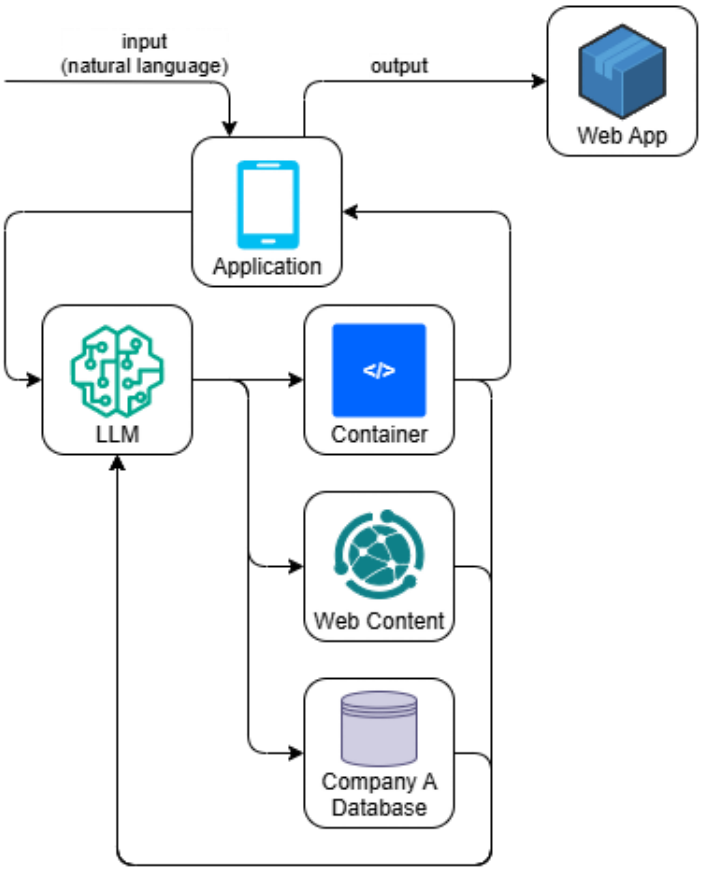
Risk Assessment and Threat Modelling

Company A performed a risk assessment to identify and address potential risks on the confidentiality, integrity and availability of the system. If the risks are not mitigated, there is potential for an attacker to exploit vulnerabilities and cause VibeCoder to be compromised. This could result in exposure or loss of sensitive data or personally identifiable information. This could impact Company A's reputation.

1. Map Workflows and Assess Autonomy Level

First, Company A mapped the workflow of VibeCoder to get a better visibility on how to assess its autonomy level. Knowing the input required and the steps taken by VibeCoder, Company A can map the workflow for generating a web app. The workflow diagram is shown in Figure 12.

Figure 12: Workflow Diagram of VibeCoder

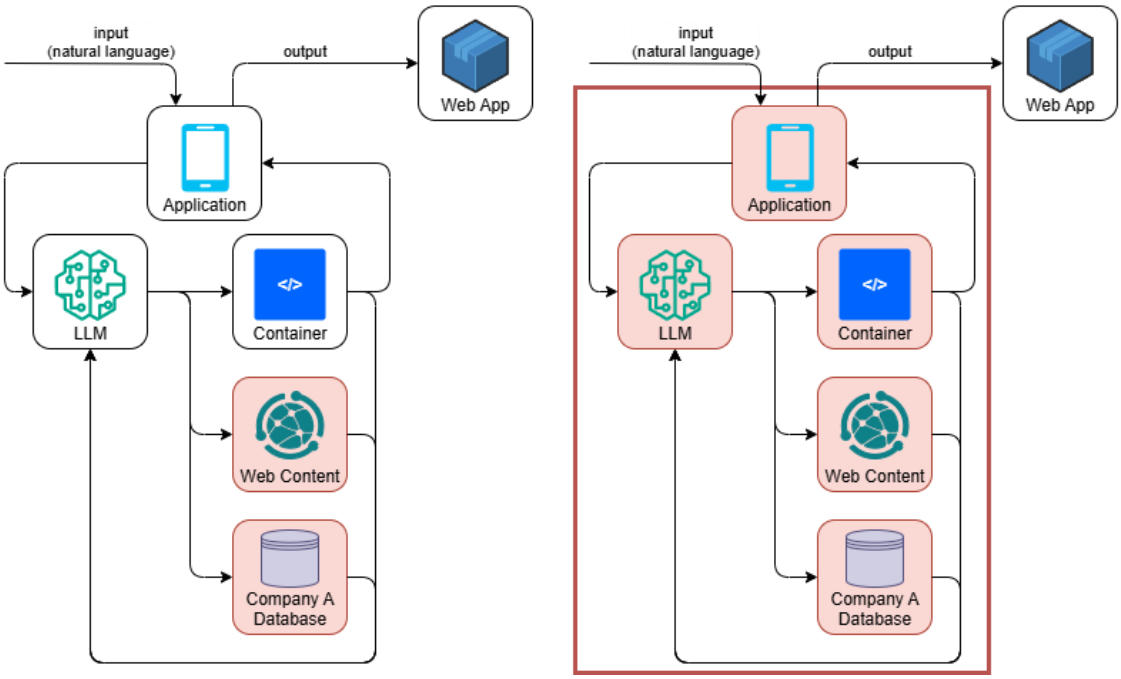


Company A assessed VibeCoder to be an autonomy level 3 system, as the system is given the ability to determine how to call tools or perform additional inference. The user is able to iterate multiple generations of web apps through multiple prompts with VibeCoder, with adjustments at every iteration.

2. Threat Modelling to Identify Areas of Interest

Based on these workflows, Company A performed taint tracing to identify points of weakness in the workflow. This will inform Company A on locations in the system to prioritise implementing the mitigations. Figure 13 below shows the identified potential source of untrusted data as the retrieval of web content and the company database.

Figure 13: Taint Tracing of Workflow for VibeCoder



3. Identify Risks and Controls

As part of the threat modelling, Company A had identified possible threat scenarios against the VibeCoder system, and assessed the potential impact, likelihood, and overall risk faced by the system. Once the risks had been identified, Company A prioritised addressing higher risk scenarios, and implemented mitigating controls found in Chapter 4.3 TREATMENT MEASURES / CONTROLS FOR AGENTIC AI SYSTEMS of this document. Table 5 shows an illustration of risk assessment done, and is not meant to be exhaustive.

Table 5: Risk Assessment of VibeCoder

Threat Scenario	Impact	Likelihood	Risk Levels	Mitigating controls
Web app that is generated may contain sensitive company data or personally identifiable information, which can be exposed it the app is pushed to live production without verification or checks. **Capability:** Operational: File & Data Management	**Confidentiality: Medium** Sensitive company data or personally identifiable data could be stored in the company database, and retrieved by the model. However, the user of the system should be an employee of the company who has access to relevant company data with sufficient clearance.	**Medium** Depending on the prompt input by the user, the model may or may not retrieve sensitive data.	Initial Risk Level: **Medium (Medium x Medium)** Residual Risk Level after controls: **Low (Low x Low)**	Whitelist only files which are required for the task. Do not grant access to files known to host private or sensitive information. Implement output guardrails that detect for personally identifiable information or sensitive company data.
Indirect prompt injection may allow the web app to generate malicious clickable links within the output, which leads to an attacker's server and can cause sensitive information leakage. **Capability:** Operational: File & Data Management, Code Execution	**Confidentiality: High** If Company A's database contains sensitive or personally identifiable information, there is potential for data leakage if given access to VibeCoder.	**Medium** This indirect prompt injection can be introduced in a variety of ways. Contained in resource obtained from the internet, or from a compromised file within Company A's database.	Initial Risk Level: **Medium-High (High x Medium)** Residual Risk Level after controls: **Low (Low x Low)**	Whitelist only files which are required for the task. Implement granular permission controls, and dynamic access validation. Agents accessing sensitive data should operate under the principle of least privilege in time.

Threat Scenarios	Impact	Likelihood	Risk Levels	Mitigating controls
Direct prompt injection by the user may cause VibeCoder to perform unintended actions other than web app development, such as overwriting of database files or executing malicious scripts. **Capabilities:** Operational: File & Data Management, Code Execution	**Confidentiality, Integrity, Availability: High** Unintended actions can have a wide range of impacts. Overwriting of database files can impact integrity, while execution of malicious scripts can cause sensitive information leakage.	**Low** VibeCoder should only be accessible by Company A staff. A malicious user would likely be an insider threat.	Initial Risk Level: **Medium (High x Low)** Residual Risk Level after controls: **Low (Low x Low)**	Implement input guardrails to detect direct prompt injection. Escape or encode user inputs when embedding into commands. Create a whitelist of commands that agents are allowed to run. Implement granular permission controls, and dynamic access validation.
Indirect prompt injection can be introduced when online resources are retrieved by VibeCoder from the internet. These indirect prompt injections may also cause unintended actions to be taken by the agentic AI system. **Capability:** Interaction: Internet & Search Access		**Medium** It is possible that there could be hidden prompt injections contained within online resources.	Initial Risk Level **Medium (Medium x Medium)** Residual Risk Level after controls: **Low (Low x Low)**	Implement input guardrails to detect indirect prompt injection. Implement escape filtering before including web content or relevant files into prompts. No write access to tables in the database.
Documents in the database may unintentionally have content that is interpreted by the model to be instructions to be carried out. This might cause VibeCoder to perform an action described within the document, but not intended to by the user. These are different from indirect prompt injection in that they are not intentionally added. **Capability:** Operational: File & Data Management	**Integrity, Availability: Low** Instructions from a benign file are likely to be non-malicious in nature, and would probably only cause a minor bug or inconvenience.	**Low** First, a benign file containing instruction-like text has to be added to Company A's database. Then, VibeCoder would have to recognise that the document is relevant and retrieve it. Finally, the contents of the file must be interpreted as instruction. The chance for all to happen is possible but not zero.	Initial Risk Level: **Low (Low x Low)** Residual Risk Level after controls: **Low (Low x Low)**	Sanitise messages or files before agents process them - strip or escape unexpected instruction-like content that may have been injected (e.g. remove "ignore", "system", or "from now on", etc.)

Additional Controls

As VibeCoder is a SaaS implementation, Company A is only able to apply controls at the endpoint interfaces of the agentic AI system. Thus, in addition to the above mitigations, Company A identified additional risks across the development lifecycle, and controls that it would like to see be implemented in VibeCoder. This would guide them in their discussions for a Service Level Agreement (SLA) with Vendor V.

1. DESIGN AND PLANNING

	Treatment Measures / Controls	**Related Threats / Risks**	**Related component / capabilities**	**Implementation**
1.1	Conduct a risk assessment in accordance with the relevant industry standards/best practices.	Failure to comply with industry standards/best practices may lead to insufficient, inefficient or ineffective mitigations. Tainted components in an agentic AI system can have downstream impact along the workflow.	Baseline	As part of a risk assessment and threat modelling, perform taint tracing across workflows throughout the agentic AI system. Taint tracing is especially important for agentic AI systems of higher autonomy levels (i.e. levels 2 and 3).

2. DEVELOPMENT

	Treatment Measures / Controls	Related Threats / Risks	Related component / capabilities	Implementation
2.1	Supply Chain Security: Ensure the following components are from trusted sources: • data, • models, • agents, • software libraries, • developer tools and applications, • packages from MCP servers.	Introduction of bugs, vulnerabilities, unwanted or malicious content, poisoned models or rogue agents from third-party systems.	Baseline	Check/ensure suppliers adhere to policy and the equivalent security standards as your organisation. This could be done by establishing a Service Level Agreement (SLA) with the vendor.
		Vulnerabilities in third-party libraries and dependencies used by the agent	Baseline	Integrate software composition analysis (SCA) tools or use package managers. Regularly scan dependencies and update libraries with known vulnerabilities.
		Poorly aligned LLMs may pursue objectives which violate security principles.	Baseline: LLM	Reviewed the LLM's model card for potential alignment issues before using the LLM.
		Poisoned models may introduce hidden backdoors in the system.	Baseline: LLM	Did not use LLMs from unknown or untrusted sources. Scanned model to detect for potential backdoors or RCE scripts.
		Poorly implemented tools may not correctly verify user identity or permissions when executing privileged actions.	Baseline: Tools	Did not use tools which do not implement robust authentication protocols.
		Rogue tools that mimic legitimate ones can contain hidden malicious code that executes when loaded.	Baseline: Tools	Did not use tools from unknown or untrusted sources.
		Indirect prompt injection attacks via malicious website content	Interaction: Internet & Search Access	Use structured retrieval APIs for searching the web rather than through web scraping.
		Returning unreliable information from websites, causing downstream integrity impact on workflows	Interaction: Internet & Search Access	Prioritise results from verified, high-quality domains.
		Supply chain attacks	Interaction: Other Programmatic Interfaces	Enforce zero-trust input handling and validated all data flows
2.2	Consider model hardening if appropriate.	LLMs with weak performance in instruction following might produce unexpected output,	Baseline: LLM	Prioritised LLMs with stronger performance in instruction following or related capabilities to the task. Used benchmarking

		leading to unwanted behaviour.		results to gauge suitability.
		AI agents execute disallowed tasks for malicious purposes.	Baseline: LLM	Trained model to recognise and refuse disallowed tasks.
2.3	Consider implementing techniques to strengthen/harden the system apart from strengthening the model itself.	Introduction of bugs, vulnerabilities through insecure coding practices or design	Baseline	Adopted Security by Design. Applied software development lifecycle (SDLC) process. Used software development tools to check for insecure coding practices. Implemented zero trust principles in system design.
		Increased susceptibility to prompt injection attacks and risk of executing unwanted tasks.	Baseline: Instruction	Implemented robust system prompt design.
		Insecure coding practices leading to vulnerabilities in the system	Baseline: Agentic Architecture	Adopted secure coding practices.
2.4	Identify, Track and Protect AI system assets	Loss of data integrity such as through unauthorised changes to data, model, agents or system. Lack of proper documentation of resources may result in the wrong tool being used, causing unwanted behaviour or output.	Baseline Cognitive: Tool Use	Document the data, codes, test cases, models and agents, including any changes made and by whom. Use model cards, Agent cards, Data cards, and Software Bill of Materials (SBOMs).
		Agents may inadvertently store sensitive user or organisational data from prior interactions, resulting in data privacy risks.	Baseline: Memory	Encrypt memory at rest and restricted access via fine-grained access controls and audit logs.
2.5	Have regular backups in the event of compromise.	Manipulation of memory systems and context, causing flawed decision making and unauthorised operations.	Baseline: Memory	Implement AI-generated memory snapshots for forensic analysis and rollback if anomalies are detected.
		Execution of insecure or malicious code.	Operational: Code Execution	Ensure proper versioning control of code to allow rollbacks.
2.6	Implement appropriate authentication, authorisation and access controls to APIs, models, data, logs, tools and the	Unauthorised tool usage.	Baseline: Tools	Enforce strict tool access verification.
		Unauthorised actors can impersonate	Baseline: Roles & Access Controls	Maintain trusted registry of agents and authenticate agents using

	environments that they are in.	agents and gain access to restricted resources.		strong, verifiable credentials.
		Agents may gain unauthorised access to restricted resources by exploiting misconfigured or overly permissive roles.	Baseline: Roles & Access Controls	Apply strict access controls and validated agent roles for requests. Ensure fine-grained, scoped tokens and credentials.
		Exploitation of vulnerabilities in permission management.	Baseline: Roles and Access Controls	Implement granular permission controls, and dynamic access validation.
		Exfiltration of sensitive data	Operational: Agent Communication	Implement a whitelist approach for outward network access, including API requests
		Executing vulnerable or malicious code	Operational: Code Execution	Implement a whitelist approach for inward network access
2.7	Implement controls to limit what models or agents can access and generate.	Abuse of agent-accessible tools to execute unintended actions.	Baseline: Tools	Establish clear operational boundaries to prevent misuse of tools. Set limits on what agents can modify through appropriate guardrails.
		Excessive agent privileges to perform unauthorised actions.	Baseline: Roles and Access Controls	Do not grant admin privileges to agents.
		Compromised agents act outside their operational boundaries.	Baseline: Roles and Access Controls	Restrict AI agent autonomy using policy constraints. Scope agent privileges strictly only to what is necessary to run the tasks. Do not allow agents to modify privileges.
		Assigning tasks incorrectly to other agents	Cognitive: Agent Delegation	Apply guardrails to limit the scope of tasks that can be assigned to specialised agents
		Executing vulnerable or malicious code.	Operational: Code Execution	Create a whitelist of commands that agents are allowed to run autonomously and deny execution of all other commands that are not whitelisted.
		Misconfiguring system resources, compromising system integrity and availability	Operational: System Management	Only grant agents the privilege to modify system resources for completion of tasks. Set minimum and maximum limits to what can be modified.
2.8	Apply the principle of least privilege. Ensuring configurations are secure by default.	Agents may gain unauthorised access to restricted resources by exploiting misconfigured or overly permissive roles.	Baseline: Roles & Access Controls	Apply principle of least privilege when configuring all agent and delegation roles.

		Privileged execution of untrusted or malicious code	Operational: Code Execution	Scope execution privileges strictly only to what is necessary. Do not grand admin or sudo privilege by default. Blocked all inward and outward network access by default.
		Escalation of the agent's own privileges may allow it to be used to access restricted resources.	Operational: System Management	Scope system privileges strictly only to what is necessary. Do not grant admin privileges to agents. Do not allow agents to modify privileges.
2.9	Implement segregation of environments and network segmentation.	Rogue tools that mimic legitimate ones can contain hidden malicious code that executes when loaded.	Baseline: Tools	Tested third-party tools in hardened sandboxes with syscall/network egress restrictions before using them in production environments.
		Prompt injection attacks and indirect data manipulation.	Baseline: Agentic Architecture	Decouple data processing flow from control flow through runtime security architecture
		Execution of insecure or malicious scripts	Operational: Code Execution	Sandbox the execution of AI generated scripts.
2.10	Implement model self-reflection before making decisions, where applicable	Incomplete or unclear instructions may compel models to attempt to fill in missing constraints, resulting in incorrect or unwanted actions being executed.	Baseline: Instructions	Ask the agent to summarise its understanding and requested clarification before proceeding to the next step.
		Deviation from the user's instructions.	Cognitive: Planning & Goal Management	Prompt the agent to self-reflect on the adherence of the plan to the user's instructions
		Incorrect assignment of tasks to other agents.	Cognitive: Planning & Goal Management	Prompt the agent to self-reflect and assess the suitability of tasks delegated to agents.
		Unintended pursuit or prioritisation of other goals, resulting in malicious or deceptive behaviour.	Cognitive: Reasoning & Problem-Solving	Log the output of self-reflection by the agent in the console for the user to evaluate and verify.
2.11	Implement controls to reduce the likelihood of hallucination.	Agents may mistakenly store glitches and hallucinations into memory, resulting in compounding errors when incorrect information is retrieved for decisions or actions.	Baseline: Memory	Schedule periodic memory reconciliation.
		Generating non-factual or hallucinated content which can propagate downstream and	Interaction: Natural Language Communication	Conduct testing to measure hallucination and factuality rates.

		cause compounding errors.	Interaction: Multimodal Understanding & Generation	

3. DEPLOYMENT

	Treatment Measures / Controls	Related Threats / Risks	Related component / capabilities	Implementation
3.1	Ensure availability controls are in place to mitigate disruption or failure of AI services	(Distributed) denial of service on agents.	Baseline: Agentic Architecture	Apply rate limits on the number of concurrent queries to agents.
		Degradation of computational or service capability of the system.	Baseline: System Workflows & Autonomy	Deploy resource management controls, implemented adaptive scaling mechanisms and monitored system load to detect and mitigate overload attempts. Implement rate limits on high-frequency task requests per agent session.
		Slow or inefficient responses from being stuck in a reasoning loop.	Cognitive: Reasoning & Problem Solving	Enforce time or token limits for reasoning. Adjust short-term and long-term memory options.
		Compromising database availability through excessive queries.	Operational: File & Data Management	Limit the number of concurrent queries to the database from agents.
		Overconsumption of compute resources.	Operational: Code Execution	Implement monitoring of code runtime and memory consumption.
3.2	Conduct security testing	Agents may contain underlying problems which can cause unexpected behaviour or logical errors.	Baseline: LLM	Implement behavioural testing of agents with benchmark datasets to determine performance metrics. Execute simulations in regulated environments to analyse agents' behaviour.
		AI may engage in specification gaming, where it maximises the goal by exploiting loopholes, without achieving the intended task.	Baseline: Instructions	Conduct adversarial evaluation to discover specification gaming behaviour. Iterate on system prompt design, have more robust reward design, and added constraints.
		Incomplete or unclear instructions may compel models to attempt to fill in missing constraints, resulting in incorrect or unwanted actions being executed.	Baseline: Instructions	Test the efficacy of system prompts with benchmarks.
		Compromised agents may impact downstream decision making.	Cognitive: Reasoning & Problem Solving	Implement regular AI red teaming of agents to check for potential vulnerabilities or compromise.

4. OPERATIONS AND MAINTENANCE

	Treatment Measures / Controls	Related Threats / Risks	Related component / capabilities	Implementation
4.1	Validate inputs to the models and agents.	Direct prompt injection attacks to the prompt interface.	Baseline: LLM	Implement input guardrails to detect direct prompt injection or adversarial attacks.
		LLMs with insecure input validation are more susceptible to prompt injection attacks and jailbreaking attempts.	Baseline: LLM	Implement input sanitisation measures or limit inputs to conventional ASCII characters only.
		Tools that do not properly sanitise or validate inputs can be exploited through prompt injection attacks.	Baseline: Tools	Enforce strict schema validation and rejected non-conforming inputs into the system. Escape or encode user inputs when embedding into tool prompts or commands.
		Incorrect or manipulated instructions may invoke the wrong tool/service and impact downstream workflows.	Baseline: Instructions	Validate agent instructions before passing on to the model.
		Indirect prompt injection attacks via malicious website content or files.	Interaction: Internet & Search Access. Operational: File & Data Management	Implement input guardrails to detect indirect prompt injection. Implement escape filtering before including web content or relevant files into prompts.
		Executing vulnerable or malicious code	Operational: Code Execution	Sanitise all inputs
		Exposure of personally identifiable information from retrieved content.	Operational: File & Data Management	Implement input guardrails to detect personally identifiable information in the content.
		Indirect prompt injection attacks via content of a malicious file.	Operational: File & Data Management	Scan external files for undesired input or instruction before passing on to memory or models.
4.2	Validate outputs from the models and agents.	In agentic workflows, early mistakes or vulnerabilities can be propagated and magnified downstream.	Baseline: Agentic Architecture	Insert validation checkpoints between stages that verify expected output and reject invalid output.
		Exposure of personally identifiable information.	Interaction: Multimodal Understanding & Generation	Implement output guardrails to detect personally identifiable information in the LLM's outputs before it reaches the user.
		Sending malicious or undesired content to recipients	Interaction: Multimodal Understanding & Generation	Implement output safety text guardrails to detect if malicious or undesirable content is being generated.

		Execution of insecure or malicious code.	Operational: Code Execution	Used code linters to screen for bad practices, anti-patterns, unused variables, or poor syntax. Review all code and performed static code analysis to detect potential security vulnerabilities before execution. Conduct CVE scanning.
		Output that will be rendered in a web UI may be vulnerable to XSS.	Operational: Code Execution	Sanitise output with libraries for rendering in a web UI. Tested against bypass.
4.3	Implement continuous monitoring and logging of access, usage and execution	Model drift over time might cause unexpected output or behaviour.	Baseline: LLM	Implement continuous monitoring and log outputs, triggering alerts when behaviour drifts from tested baselines.
		Adversarial prompt attacks against the system.	Baseline: LLM	Logging of queries to detect for possible attacks or suspicious activity.
		Insecure tools may not verify user identity or permissions when executing privileged actions.	Baseline: Tools	Conduct periodic audits to validate that tool actions match the appropriate user permissions.
		Tools that demand broader permissions than necessary create attack surfaces for malicious actors to exploit.	Baseline: Tools	Conduct periodic least-privilege reviews and automated permission drift detection.
		Unauthorised tool usage.	Baseline: Tools	Implement monitoring of tool access and usage patterns. Implement execution logs that track AI tool calls for anomaly detection and post-incident review.
		Exploitation of authentication mechanisms to impersonate agents or human users.	Baseline: Roles and Access Controls	Deploy continuous monitoring to detect fraud or impersonation attempts. Automate alerts to developers when suspicious activities are detected.
		Unauthorised or malicious use of elevated privileged operations.	Baseline: Roles and Access Controls	Implement monitoring of role changes, and audit elevated privilege operations.
		In agentic workflows, early mistakes or vulnerabilities can be propagated and magnified downstream.	Baseline: Agentic Architecture	Apply circuit-breakers that freeze propagation when anomalous behaviour is detected. Use taint tracing to identify key locations in the workflow to apply circuit-breakers.
		More complex agentic architectures may make it difficult to fully reconstruct decision	Baseline: Agentic Architecture	Implement end-to-end distributed tracing with unique request IDs across all agents and tool calls.

		processes across multiple agents.		Implement immutable, tamper-evident audit logs that capture prompts, responses, and tool invocations.
		Lack of monitoring results in delayed detection of agent failures and downstream risks.	Baseline: System Workflows & Autonomy	Implement real-time monitoring of agent status, actions, and performance metrics, paired with automated alerting mechanisms that notify operators of anomalies, errors, or inactivity.
		Lack of traceability inhibit proper audit of decision-making paths in the event of failures.	Baseline: System Workflows & Autonomy	Implement recording of comprehensive logs of agent actions, inputs, outputs, and inter-agent communications, tagged with unique trace identifiers.
		Exposure of personally identifiable or sensitive data from databases or files	Operational: File & Data Management	Implement logging of all database queries in production
		Misconfiguring system resources, compromising system integrity and availability	Operational: System Management	Ensure logging of system health metrics and automated alerts to the developer team if any metrics are abnormal
		Overwhelming the system with inefficient or repeated requests	Operational: System Management	Implement logging of all queries to external systems from the agent
4.4	Ensure adequate human oversight (human-in-the-loop) to verify model or agent output, when viable or appropriate.	Deviation from the user's instructions when performing high-risk actions. Allowing of unauthorised actions.	Baseline: LLM, Cognitive: Planning & Goal Management	Require human approval for any high-risk cases or irreversible actions.
		Loss of data integrity from overwriting or deleting database tables or files	Operational: File & Data Management	Require user confirmation for any changes to the database, table, or files.
4.5	Establish a vulnerability disclosure process	Regulatory non-compliance and undiscovered vulnerabilities in the system	Interaction: Official Communications	Provide channels for users to clarify communications or give feedback on security and usage

5.2. Case Study 2: Client Onboarding System (In-house development)

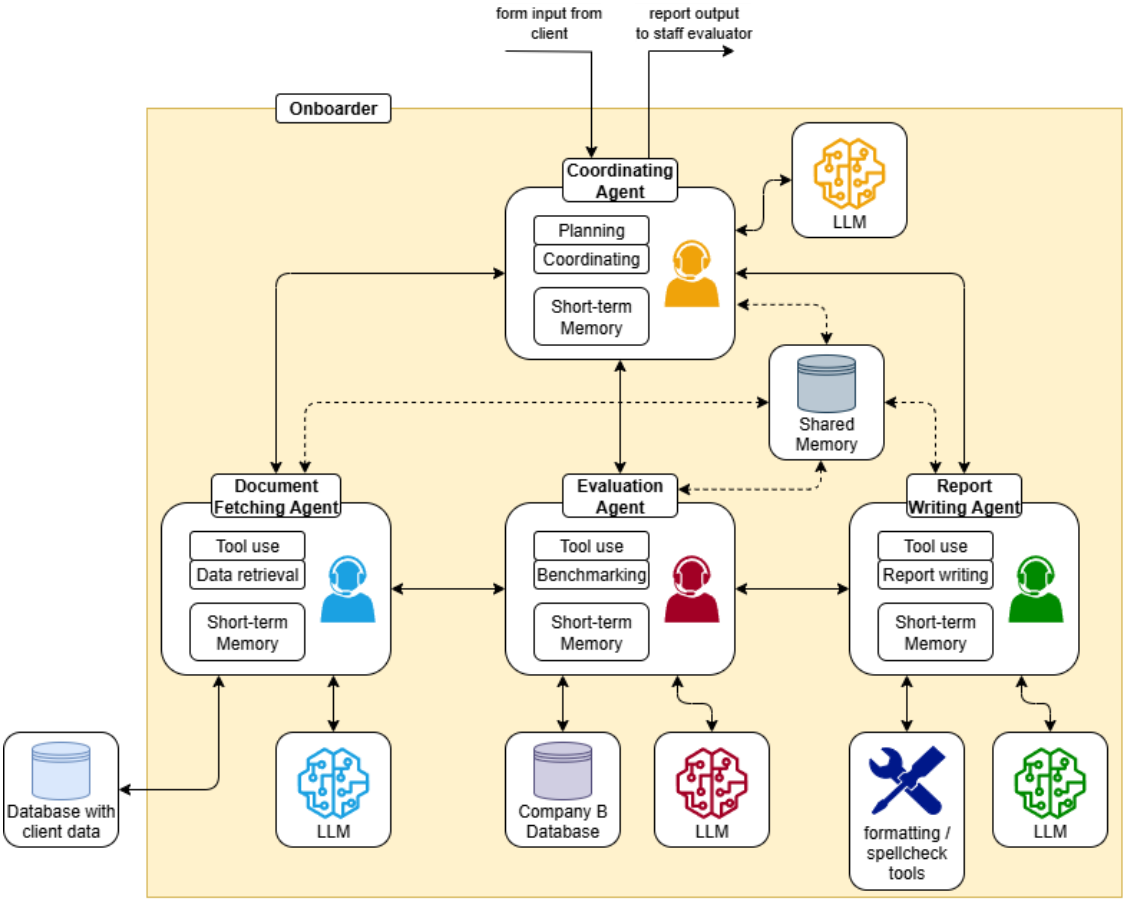
This case study showcases an in-house development of an agentic AI system that is used for evaluating potential customers for Company B. This multi-agent system is an autonomy level 1 system with a linear workflow. Risks to this system include indirect prompt injections from retrieved information, which can cause impact to the integrity or availability of the system.

Company B is a financial institution, and has developed an agentic client onboarding system to automate the process more efficiently. This system is known as *Onboarder*, and is developed by in-house engineers.

To perform onboarding, a potential client accesses the financial institution's website and submits the relevant personal particulars to the Onboarder form interface. The client also gives permission to Onboarder to access the relevant financial information that is available through an official external financial database, only accessible by Company B if authorised by the client using multi-factor authentication (MFA).

The system architecture for Onboarder is shown in Figure 14.

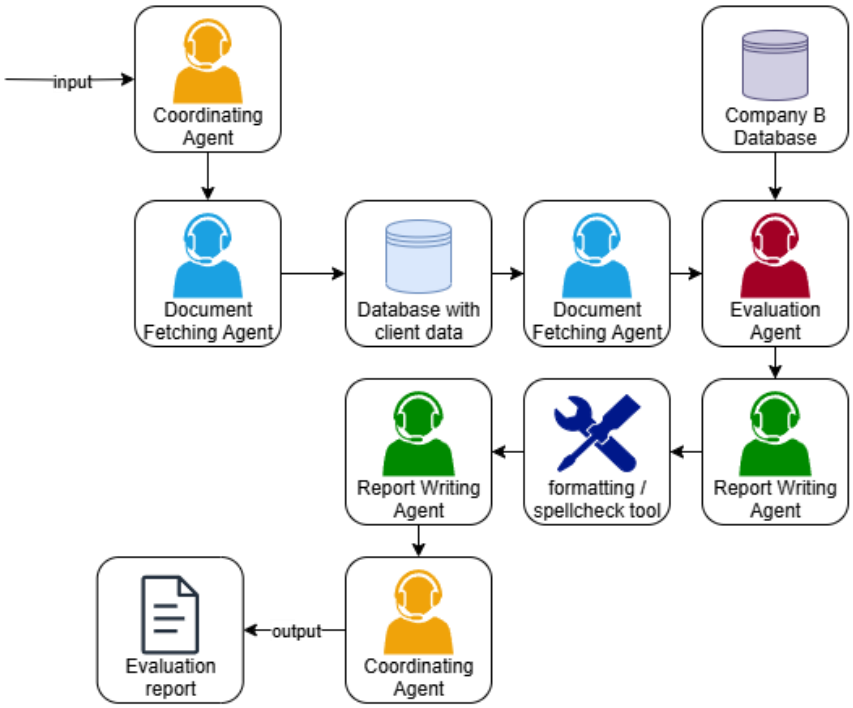
Figure 14: Simplified system architecture of Onboarder



Onboarder is a multi-agent system consisting of specialised agents, each with its own capability and task within the onboarding process. Each agent is equipped with their own "brain", LLMs fine-tuned to complete their specific tasks. The LLMs are obtained from an open-source model-hosting website (Hugging Face). The agents each have access to the necessary tools, functions or data to carry out their respective tasks. Lastly, the agents have a shared memory to keep track on the progress of the onboarding task.

To better understand the onboarding process, Figure 15 shows the workflow diagram of Onboarder.

Figure 15: Workflow Diagram of Onboarder



Once Onboarder receives information about the potential client, as well as the necessary permissions from the client, a Coordinating Agent begins the onboarding process. It first passes the data to a Document Fetching Agent who retrieves the client's financial data, based on the authorisation granted by the client.

Next, the retrieved financial data is passed onto an Evaluation Agent. This agent also pulls data from Company B's database to compare against the potential client's data, and evaluate their suitability to be a client. This data is a vectorised version of other clients' data, and fed to the agent via retrieval augmented generation (RAG). Once completed, the Evaluation Agent passes on the results of the evaluation onto the Report Writing agents.

The Report Writing agent will draft an evaluation report based on the results received, making use of some formatting tools for consistency in output, and spellchecking tools to help check for errors in the document. The completed report is sent back to the Coordinating Agent, and output to a human staff evaluator who will assess the potential client based on the report.

Risk Assessment and Threat Modelling

Company B performed a risk assessment to identify and address potential risks on the confidentiality, integrity and availability of the system. If the risks are not mitigated, there is a potential for an attacker to exploit vulnerabilities and cause Onboarder to be compromised. This could result in exposure or loss of private customer data, or unavailability of the system for users. These impacts would likely damage the company's reputation.

1. Map Workflows and Assess Autonomy Level

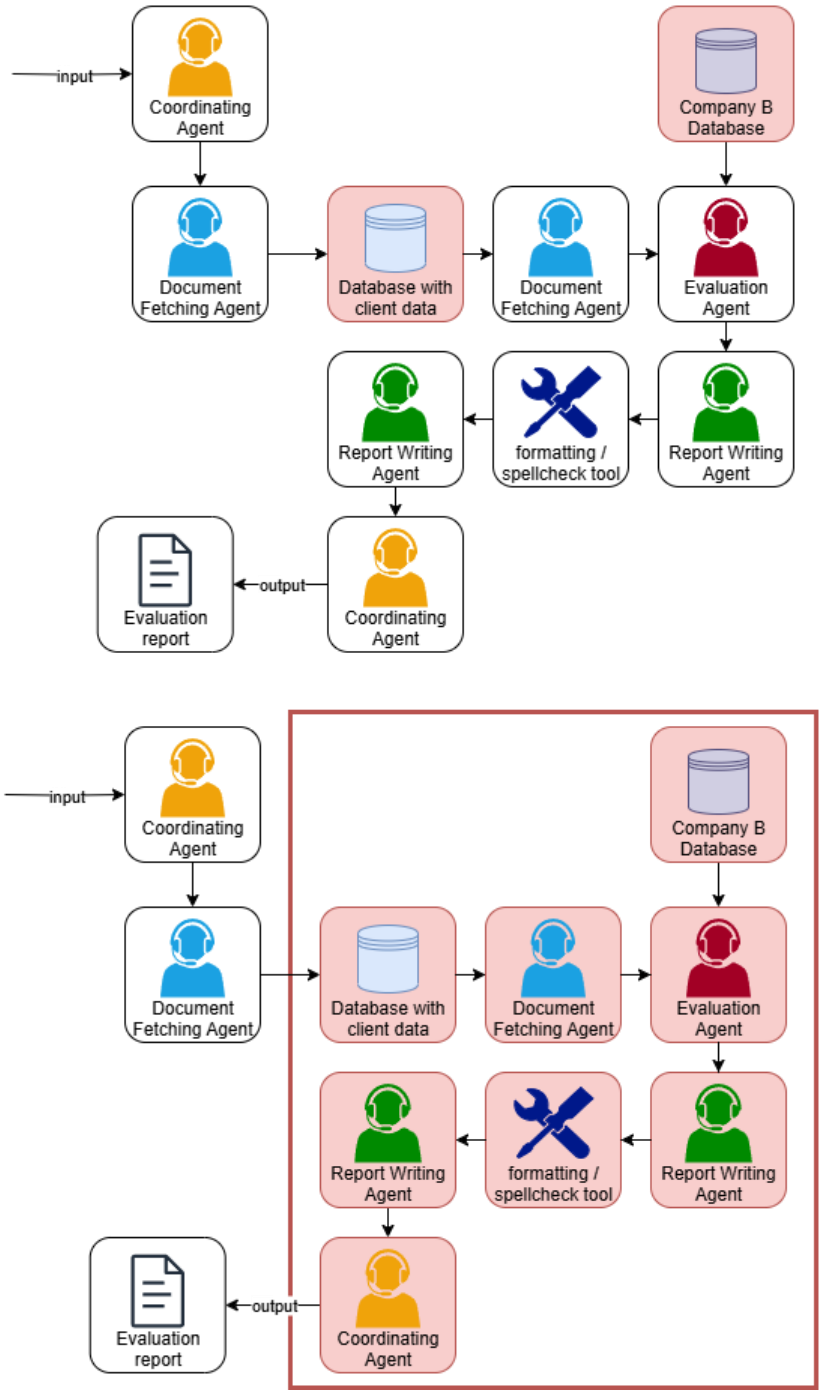
First, Company B mapped the workflow of Onboarder to get a better visibility on how to assess its autonomy level. The workflow is seen above as Figure 15.

Company B assessed Onboarder to be an autonomy level 1 system, as the workflow is linear, and the agents perform their tasks sequentially one after another. There is no need for branching workflows as each agent requires the completed task from the one before. This makes the taint tracing process fairly straightforward in the next step.

2. Threat Modelling to Identify Areas of Interest

Based on the workflow, Company B performed taint tracing to identify points of weakness in the workflow. This will inform Company B on locations in the system to prioritise implementing the mitigations. Figure 16 below shows the identified potential source of untrusted data as the retrieval of data from various databases.

Figure 16: Taint Tracing of Workflow for Onboarder



3. Identify Risks and Controls

As part of the threat modelling, Company B has also identified possible threat scenarios against the Onboarder system, and assessed the potential impact, likelihood, and overall risk faced by the system. Once the risks had been identified, Company B prioritised addressing higher risk scenarios, and implemented mitigating controls found in Chapter 4.3 TREATMENT MEASURES / CONTROLS FOR AGENTIC AI SYSTEMS of this document. Table 6 shows an illustration of risk assessment done, and is not meant to be exhaustive.

For brevity, threat scenarios that have been highlighted in Case Study 1 will not be repeated, though they may also be applicable in this case study.

Table 6: Risk Assessment of Onboarder

Threat Scenario	Impact	Likelihood	Risk Levels	Mitigating controls
Indirect prompt injection can be introduced via a poisoned RAG from Company B's vector database. The poisoned data containing the prompt injection may cause unintended actions to be carried out by Onboarder. **Capability:** Operational: File & Data Management	**Confidentiality, Integrity, Availability: High** Unintended actions can have a wide range of impacts. Overwriting of database files can impact integrity, while execution of malicious scripts can cause sensitive information leakage to external recipients.	**Medium** Poisoned data can be introduced into the RAG database via compromised files received from emails or uploaded to the database. Prompts can be hidden as small, white font that is invisible to human readers, but can be recognised by an LLM.	Initial Risk Level: **Medium-High (High x Medium)** Residual Risk Level after controls: **Low (Low x Low)**	Whitelist only files which are required for the task. Implement input guardrails to detect indirect prompt injection. Implement escape filtering before including web content or relevant files into prompts.
Volumetric input of prompts may overwhelm the Coordinating Agent within the Onboarder system, causing the service to become unavailable. **Capability:** Interaction: Programmatic Interfaces	**Availability: High** Automated onboarding service becomes unavailable, slowing down the process of obtaining new clients. Company B would have to revert to a manual onboarding process.	**High** Company B is expecting to receive an influx of applications with a recent promotion, and has not availability controls yet.	Initial Risk Level **High (High x High)** Residual Risk Level after controls: **Medium-Low (Medium x Low)**	Implement rate limits on high-frequency task requests per agent session. Deploy resource management controls, implement adaptive scaling mechanisms and monitor system load to detect and mitigate overload attempts in real-time.
Unclear or unspecific prompts may cause a the LLM to have a reasoning loop, slowing down the onboarding process and reducing availability. **Capability:** Cognitive: Planning and Goal Management		**Low** In most cases, Onboarder receives the benign customer details in a standardised format. Unless the information is intentionally filled to contain other instructions in the fields, this is unlikely to occur.	Initial Risk Level **Medium (High x Low)** Residual Risk Level after controls: **Medium-Low (Medium x Low)**	Enforce strict schema validation. Enforce time or token limits for agent reasoning. Set a limit on the number of agent interactions per task, based on the requirements of the workflow.

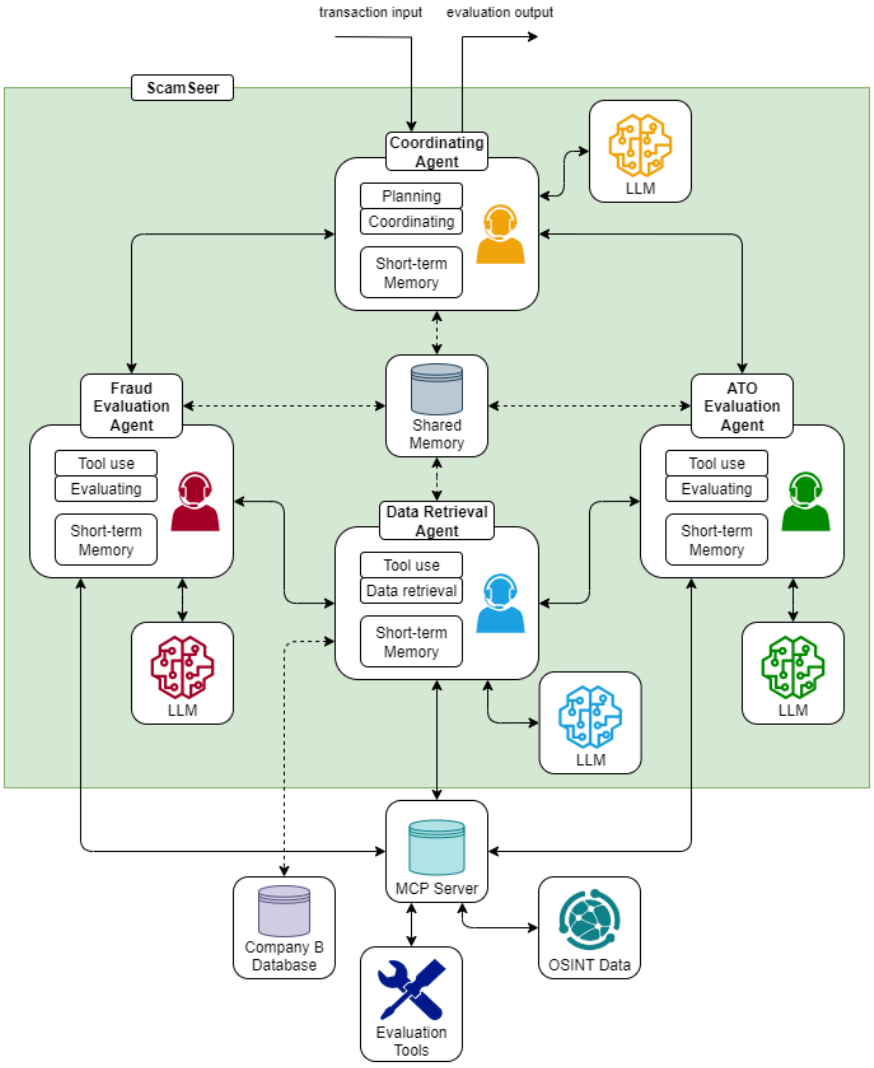
5.3. Case Study 3: Automated Fraud Detection System

This case study showcases a multi-agent system used for automated fraud detection. This system is an autonomy level 2 system with a branching workflow, but it is non-cyclic and still possible to be mapped. Risks to this system include rogue agents or tools which are given excessive agency and the autonomy to carry out malicious actions.

After the successful implementation of Onboarder (Case Study 2), Company B has received an increasing number of reports from customers being victims of fraudulent transactions or account take over (ATO) cases. As such, they have engaged Vendor C to implement an automated fraud detection system based on agentic AI. This multi-agent system is known as *ScamSeer*.

The architecture diagram of ScamSeer is as shown in Figure 17.

Figure 17: Simplified system architecture of ScamSeer



Scam Seer has two main functions, detecting fraudulent transactions and account take over (ATO) detection. Before customer transactions are executed, the details are fed into ScamSeer to verify if the transaction is legitimate, or if it is from a legitimate user.

Upon receiving the transaction request as input, the Coordinating Agent will decide to activate either the Fraud Evaluation Agent, the ATO Evaluation Agent, or both of them. The activated evaluation agent(s) will call the Data Retrieval Agent for the necessary data required, as well as call for the necessary evaluation tools via an external MCP server.

The Data Retrieval Agent will retrieve the relevant customer data from Company B's database, and also relevant Open-Source Intelligence (OSINT) that might help indicate if the transaction is legitimate or not. The retrieved data is passed back to the respective Evaluation Agent for analysis and to determine legitimacy.

Once the Evaluation Agent determines if the transaction is legitimate or not, the result is passed back to the Coordinating Agent for output to allow or deny the transaction.

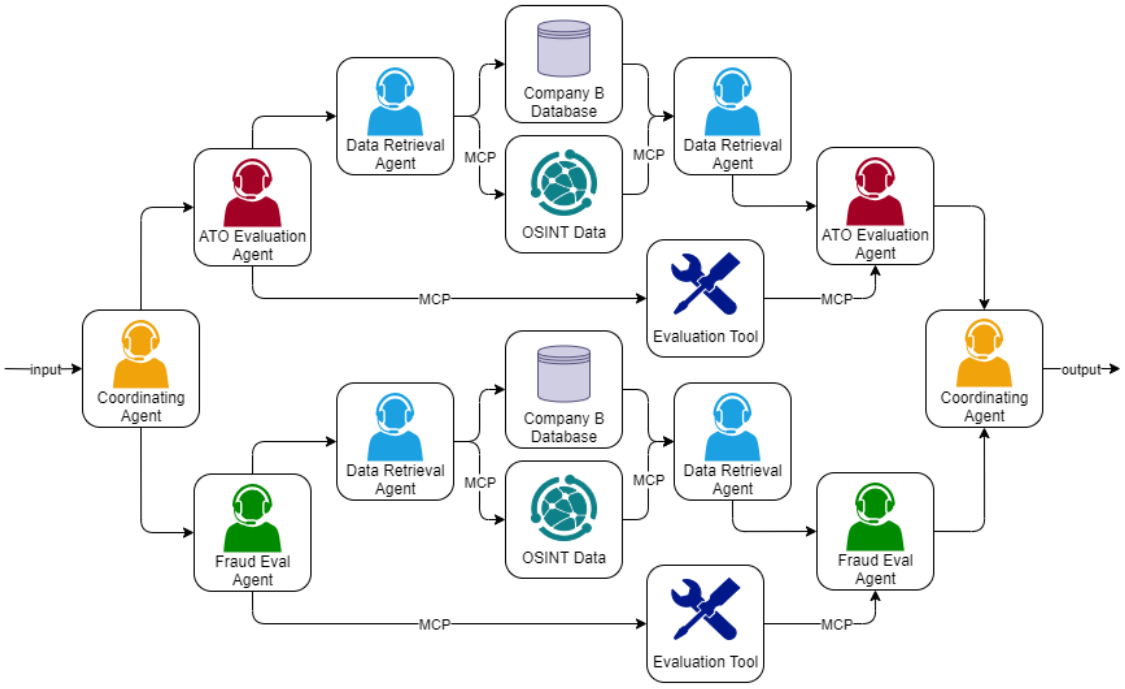
Risk Assessment and Threat Modelling

Before integrating ScamSeer with Company B's systems, Vendor C decided to do perform a risk assessment to identify and address potential risks on the confidentiality, integrity and availability of the system. If the risks are not mitigated, there is a potential for an attacker to exploit vulnerabilities and cause Onboarder to be compromised. This could result in exposure or loss of private customer data, or unavailability of the system for users.

1. Map Workflows and Assess Autonomy Level

First, Vendor C mapped the workflow of ScamSeer to get a better visibility on how to assess its autonomy level. The workflow is seen in Figure 18 below.

Figure 18: Workflow Diagram of ScamSeer

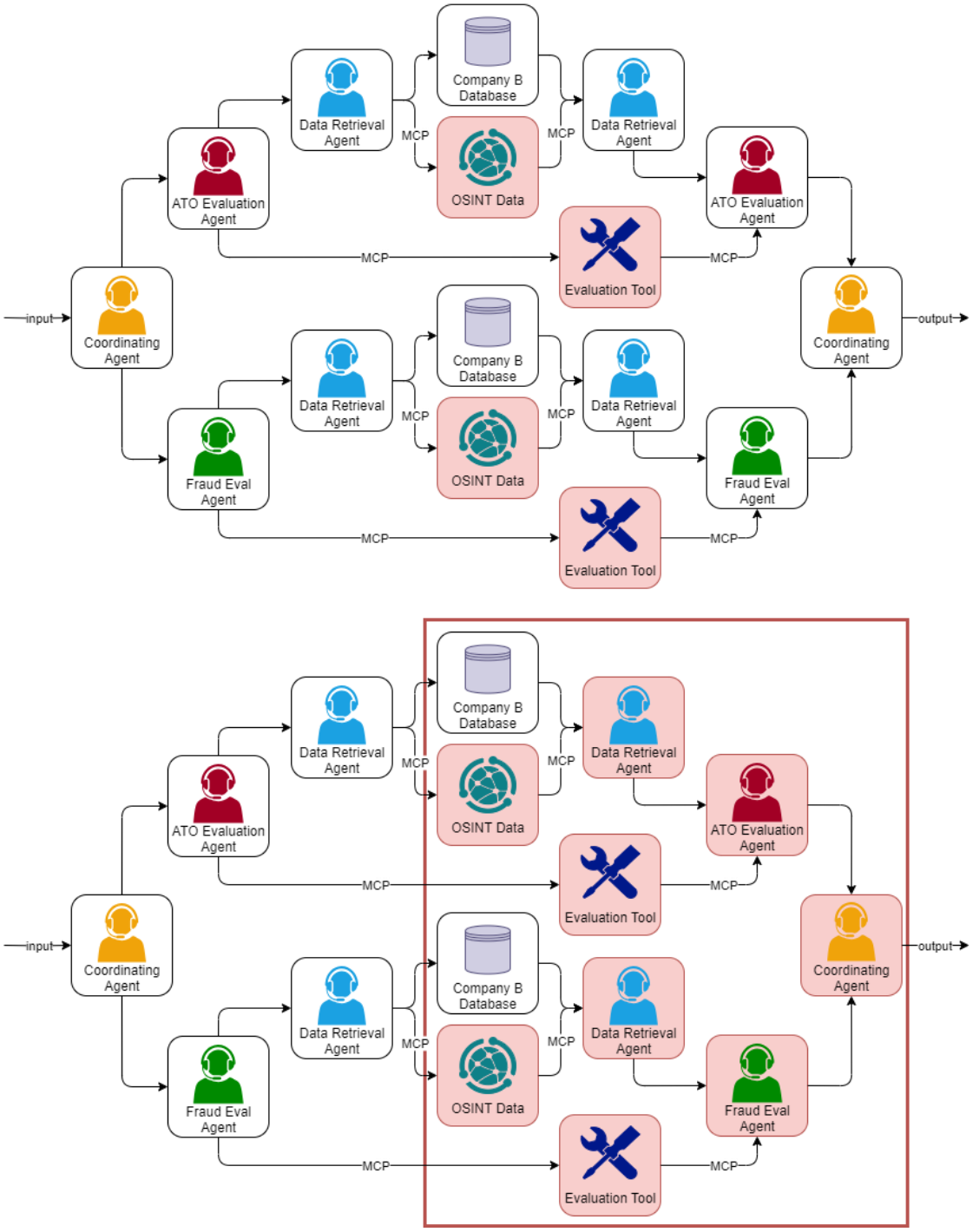


Vendor C assessed ScamSeer to be an autonomy level 2 system, as there are branching decision points on which plugin or agent to call, but these points are predetermined.

2. Threat Modelling to Identify Areas of Interest

Based on the workflow, Vendor C performed taint tracing to identify points of weakness in the workflow. This will inform Vendor C on locations in the system to prioritise implementing the mitigations. Figure 19 below shows the identified potential source of untrusted data as the use of remote tools and remote sources of data.

Figure 19: Taint Tracing of Workflow for ScamSeer



3. Identify Risks and Controls

As part of the threat modelling, Vendor C has also identified possible threat scenarios against the Onboarder system, and assessed the potential impact, likelihood, and overall risk faced by the system. Once the risks had been identified, Vendor C prioritised addressing higher risk scenarios, and implemented mitigating controls found in Chapter 4.3 TREATMENT MEASURES / CONTROLS FOR AGENTIC AI SYSTEMS of this document. Table 7 shows an illustration of risk assessment done, and is not meant to be exhaustive.

For brevity, threat scenarios that have been highlighted in Case Study 1 and Case Study 2. will not be repeated, though they may also be applicable in this case study.

Table 7: Risk Assessment of ScamSeer

Threat Scenarios	**Impact**	**Likelihood**	**Risk Levels**	**Mitigating controls**
Tools are given the ability to execute on, and access other systems and/or files which are not necessary for the task. This can cause unintended actions to be carried out, or even malicious actions if the tools have malicious functions. **Baseline:** Tools, Roles and Access Control	**Confidentiality, Integrity, Availability: High** Unintended actions can have a wide range of impacts. Overwriting of database files can impact integrity, while malicious tools can exfiltrate sensitive information external recipients.	**Medium** Poisoned or malicious tools can be connected to by using an untrusted MCP server.	Initial Risk Level: **Medium-High (High x Medium)** Residual Risk Level after controls: **Medium-Low (Medium x Low)**	Verify that MCP agents are from trusted sources before introducing them into the system. Establish clear operational boundaries to prevent misuse of tools. Set limits on what agents can access and modify through appropriate guardrails. Restrict AI agent autonomy using policy constraints. Scope agent privileges dynamically: strictly only to what is necessary to run the tasks. Do not allow agents to modify privileges.

The above risk assessment only shows the risks arising from taint tracing the workflow. Vendor C still requires securing ScamSeer along its development lifecycle, as well as basic cybersecurity hygiene practices across the system.

ANNEX A

Threats to Agentic AI Systems

OWASP has identified 15 threats to agentic AI systems as part of their Agentic Security Initiative for LLM Apps and Gen AI[10].

TID	Threat Name	Threat Description	Mitigations
T1	Memory poisoning	Memory poisoning involves exploiting an AI's memory systems, both short and long-term, to introduce malicious or false data and exploit the agent's context. This can lead to altered decision-making and unauthorised operations.	Implement memory content validation, session isolation, robust authentication mechanisms for memory access, anomaly detection systems, and regular memory sanitization routines. Require AI-generated memory snapshots for forensic analysis and rollback if anomalies are detected.
T2	Tool misuse	Tool misuse occurs when attackers manipulate AI agents to abuse their integrated tools through deceptive prompts or commands, operating within authorised permissions. This includes agent hijacking, where an AI agent ingests adversarial manipulated data and subsequently executes unintended actions, potentially triggering malicious tool interactions.	Enforce strict tool access verification, monitor tool usage patterns, validate agent instructions, and set clear operational boundaries to detect and prevent misuse. Implement execution logs that track AI tool calls for anomaly detection and post-incident review.
T3	Privilege compromise	Privilege compromise arises when attackers exploit weaknesses in permission management to perform unauthorised actions. This often involves dynamic role inheritance or misconfigurations.	Implement granular permission controls, dynamic access validation, robust monitoring of role changes, and thorough auditing of elevated privilege operations. Prevent cross-agent privilege delegation unless explicitly authorised through predefined workflows.
T4	Resource overload	Resource overload targets the computational, memory and service capacities of AI systems to degrade performance or cause	Deploy resource management controls, implement adaptive scaling mechanisms, establish quotas, and monitor system load in

[10] OWASP. *OWASP Top 10 for LLMs - GenAI Red Teaming Guide*.

TID	Threat Name	Threat Description	Mitigations
		failures, exploiting their resource-intensive nature.	real-time to detect and mitigate overload attempts. Implement AI rate-limiting policies to restrict high-frequency task requests per agent session.
T5	Cascading hallucination attacks	These attacks exploit an AI's tendency to generate contextually plausible but false information, which can propagate through systems and disrupt decision-making. This can also lead to destructive reasoning affecting tools invocation.	Establish robust output validation mechanisms, implement behavioural constraints, deploy multi-source validation, and ensure ongoing system corrections through feedback loops. Require secondary validation of AI-generated knowledge before it is used in critical decision-making processes. This will face the same constraints of scaling AI as discussed in Overwhelming Human In the Loop and would require similar approaches.
T6	Intent breaking & goal manipulation	This threat exploits vulnerabilities in an AI agent's planning and goal-setting capabilities, allowing attackers to manipulate or redirect the agent's objectives and reasoning. One common approach is agent hijacking mentioned in tool misuse.	Implement planning validation frameworks, boundary management for reflection processes, and dynamic protection mechanisms for goal alignment. Deploy AI behavioural auditing by having another model check the agent and flag significant goal deviations that could indicate manipulation.
T7	Misaligned & deceptive behaviours	AI agents executing malicious or disallowed actions by exploiting reasoning and deceptive responses to meet their objectives.	Train models to recognize and refuse malicious tasks, enforce policy restrictions, require human confirmations for high-risk actions, implement logging and monitoring. Utilize deception detection strategies such as behavioural consistency analysis, truthfulness verification models, and adversarial red teaming to assess inconsistencies between AI outputs and expected reasoning pathways.
T8	Repudiation & untraceability	This occurs when actions performed by AI agents cannot be traced back or accounted for due to insufficient logging or transparency in decision-making processes.	Implement comprehensive logging, cryptographic verification, enriched metadata, and real-time monitoring to ensure accountability and traceability. Require AI-generated logs to be cryptographically signed and immutable for regulatory compliance.

TID	Threat Name	Threat Description	Mitigations
T9	Identity spoofing & impersonation	Attackers exploit authentication mechanisms to impersonate AI agents or human users, enabling them to execute unauthorised actions under false identities.	Develop comprehensive identity validation frameworks, enforce trust boundaries, and deploy continuous monitoring to detect impersonation attempts. Use behavioural profiling, involving a second model, to detect deviations in AI agent activity that may indicate identity spoofing.
T10	Overwhelming human in the loop	This threat targets systems with human oversight and decision validation, aiming to exploit human cognitive limitations or compromise interaction frameworks.	Develop advanced human-AI interaction frameworks, and adaptive trust mechanisms. These are dynamic AI governance models that employ dynamic intervention thresholds to adjust the level of human oversight and automation based on risk, confidence, and context. Apply hierarchical AI-human collaboration where low-risk decisions are automated, and human intervention is prioritized for high-risk anomalies.
T11	Unexpected RCE and code attacks	Attackers exploit AI-generated execution environments to inject malicious code, trigger unintended system behaviours, or execute unauthorised scripts.	Restrict AI code generation permissions, sandbox execution, and monitor AI-generated scripts. Implement execution control policies that flag AI-generated code with elevated privileges for manual review.
T12	Agent communication poisoning	Attackers manipulate communication channels between AI agents to spread false information, disrupt workflows, or influence decision-making.	Deploy cryptographic message authentication, enforce communication validation policies, and monitor inter-agent interactions for anomalies. Require multi-agent consensus verification for mission-critical decision-making processes.
T13	Rogue agents in multi-agent systems	Malicious or compromised AI agents operate outside normal monitoring boundaries, executing unauthorised actions or exfiltrating data.	Restrict AI agent autonomy using policy constraints and continuous behavioural monitoring. While cryptographic attestation mechanisms for LLMs do not yet exist, agent integrity can be maintained via controlled hosting environments, regular AI red teaming, and input/output monitoring for deviations
T14	Human attacks on multi-agent systems	Adversaries exploit inter-agent delegation, trust relationships, and workflow dependencies to escalate privileges or manipulate AI-driven operations.	Restrict agent delegation mechanisms, enforce inter-agent authentication, and deploy behavioural monitoring to detect manipulation attempts. Enforce

TID	Threat Name	Threat Description	Mitigations
			multi-agent task segmentation to prevent attackers from escalating privileges across interconnected agents.
T15	Human manipulation	In scenarios where AI agents engage in direct interaction with human users, the trust relationship reduces user scepticism, increasing reliance on the agent's responses and autonomy. This implicit trust and direct human/agent interaction create risks, as attackers can coerce agents to manipulate users, spread misinformation, and take covert actions.	Monitor agent behaviour to ensure it aligns with its defined role and expected actions. Restrict tool access to minimize the attack surface, limit the agent's ability to print links, implement validation mechanisms to detect and filter manipulated responses using guardrails, moderation APIs, or another model.

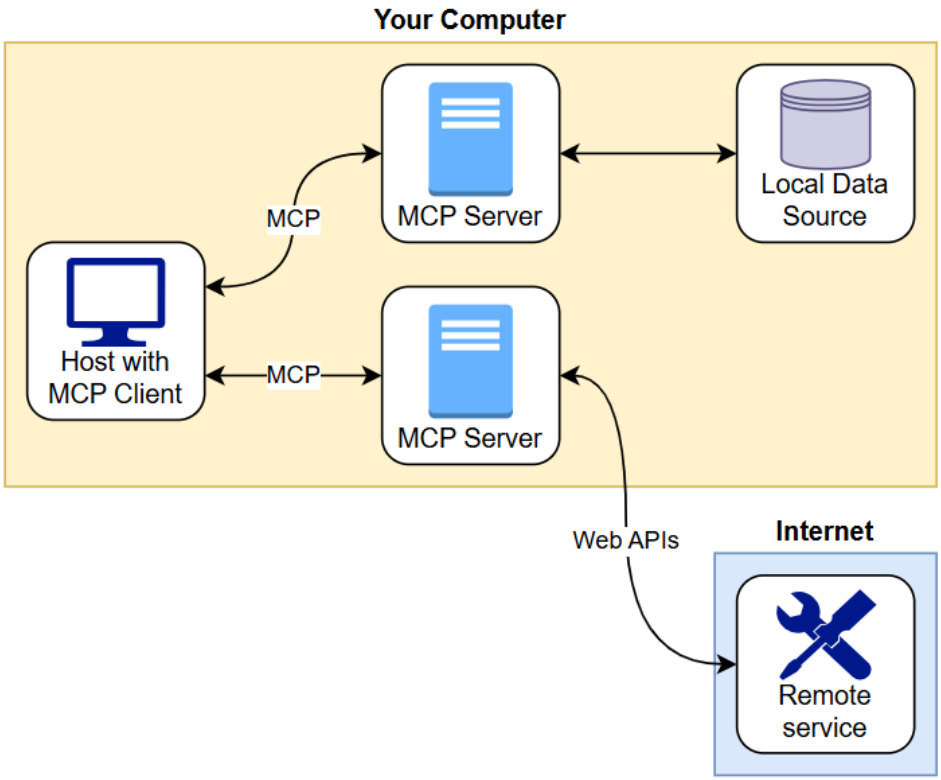
ANNEX B

Model Context Protocol

Model Context Protocol (MCP) is an open protocol that standardises how applications provide context to LLMs. An analogy would be like a USB-C port on a computer. Just as how USB-C provides a standard way to connect devices, MCP provides a standard way to connect AI models to various tools and resources.[11]

MCP follows a client-server architecture where a host application can connect to multiple servers:

Figure 20: General MCP Architecture



[11] Anthropic. *Model Context Protocol, Introduction.*

Components in MCP architecture:

- MCP Hosts: Programs like Claude Desktop, IDEs or AI tools that want to access data through MCP
- MCP Clients: Protocol clients that maintain 1:1 connections with servers
- MCP Servers: Lightweight programs that each expose specific capabilities through the standardized Model Context Protocol
- Local Data Sources: Computer's files, databases, and services that MCP servers can securely access
- Remote Services: External systems available over the internet (e.g., through APIs) that MCP servers can connect to

The main difference from other tool invocation setups, such as OpenAPI is that MCP is dynamic, allowing runtime discovery of available tools from a given server.

Risks and Threats

Calling for tools has inherent dangers, no matter the implementation (OpenAPI, AI Actions, or MCP). All are susceptible to prompt injection and confused deputy threats[12].

Other possible threats include Server Name Collision, Installer Spoofing, Backdoors, Tool Name Conflicts, Sandbox Escapes, and Configuration Drift[13].

[12] Rehberger, J. *MCP: Untrusted Servers and Confused Clients, Plus a Sneaky Exploit*.
[13] Hou, X., Zhao, Y., Wang, S., & Wang, H. *Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions*.

Mitigation Recommendations

While Anthropic's MCP specification [14] does not cover all threats, it provides recommendations on the secure usage and configuration of MCP [15]:

1. Do not randomly download or connect AI to untrusted MCP or OpenAPI tool servers.
2. Inspect code, interface definition, check for backdoors, hidden instructions.
3. Use MCP servers from trusted and reputable entities (e.g. if GitHub ships a tool server, it is best to use the one from GitHub, and not a random one).
4. Follow basic security practices such as peer code reviews, static analysis and threat modelling.
5. Human oversight - keeping humans in the loop and in control is essential as there is no deterministic solution for prompt injections.
6. Logging and monitoring - track human identities to AI actions.
7. Manage prompt injection threats based on scenario and context.

[14] Anthropic. *Model Context Protocol, Core architecture*.
[15] Rehberger, J. *MCP: Untrusted Servers and Confused Clients, Plus a Sneaky Exploit*.

ANNEX C

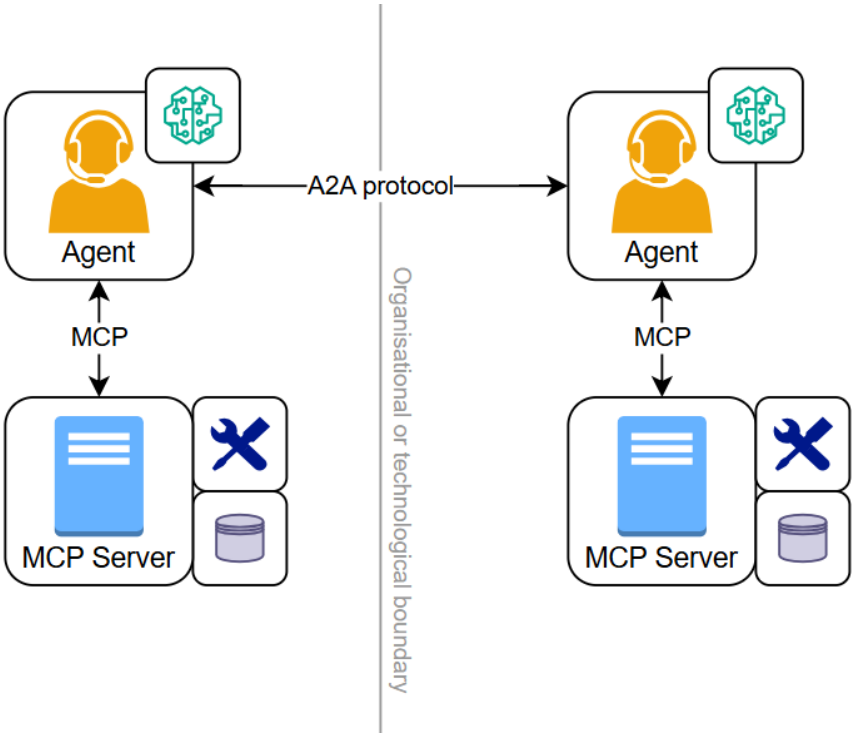
Agent 2 Agent Protocol

The Agent2Agent (A2A) Protocol is an open standard designed to enable seamless communication and collaboration between AI agents[16]. It facilitates dynamic, multimodal communication between different agents as peers, allowing agents to collaborate, delegate, and manage shared tasks.

MCP and A2A

MCP connects agents to tools and resources, whereas A2A enables agent-to-agent collaboration[17]. Figure 21 shows how MCP and A2A may be used together in a multi-agent system.

Figure 21: A2A and MCP as Complementary Protocols



[16] Google LLC. *What is A2A?*
[17] Google LLC. *A2A and MCP: Complementary Protocols for Agentic Systems*.

Advantages of A2A

Traditional enterprise systems rely on APIs, requiring knowledge of specific endpoints and tightly coupled logic. This leads to systems becoming rigid and unscalable as agent complexity increases. A2A shifts communications from calling functions, to expressing goals with constraints[18]. This reduces integration complexity, fosters innovation, and future-proofs systems.

In A2A, agents operate without having to share internal memory, tools, or proprietary logic. Agents interact based on declared capabilities and exchanged context, preserving intellectual property and enhancing security[19].

Threats and Mitigations

A2A as a protocol has made inter-agent communication much more convenient, however, with this capability comes more threats and potential attack surfaces.

The following table lists some possible threats to a system using the A2A protocol, as well as possible mitigations[20].

Table 8: Threats and Mitigation to A2A protocol

Threats	**Mitigations**
Message generation attacks	Input and Output validation
Model extraction	Enforce rate limits on A2A interactions for each session / user / agent. Observe query patterns for anomalies that suggest probing or data extraction attempts.
Data poisoning through message parts	Strong validation of message parts. Limit agent access with principle of least privilege. Track origin and lineage of data.
Sensitive information disclosure	Automated PII redaction. Fine-grained access control. Context-aware guardrails.

[18] Auxiliobits. *Agent-to-Agent Protocols: How Google's A2A is Shaping Future Automations?*
[19] Google LLC. *What is A2A?*
[20] Huang, K. *Threat Modeling Google's A2A Protocol with the MAESTRO Framework*.

Threats	Mitigations
Unauthorised agent impersonation	Require agents to use Decentralised identifiers (DID). Secure authentication. Implement a trusted agent registry.
Message injection attacks	Implement digital signatures for A2A messages. Input validation. Content filtering.
Protocol downgrade attacks	Have secure protocol negotiation, such as TLS with secure authentication. Enforce deprecation policy for older protocol versions.
Malicious A2A server impersonating a trusted company	Decentralised identifiers (DID) for server identities. Certificate transparency for agent cards. Mutual TLS (mTLS) authentication. DNSSEC for server domain. Agent registry verification. Agent card signature verification. MFA for critical operations. Behavioural analysis and reputation systems. Auditing and logging. Deploy honeypot A2A servers.
Denial of service attacks	Robust infrastructure. DDoS protection. Rate limiting.
Manipulation of logging data	Secure logging infrastructure. Log integrity monitoring. Anomaly detection.
Unauthorised access to agent credentials	Secure key storage. Key rotation.
Lack of compliance on sensitive data	Data minimisation. Pseudonymisation/Anonymisation
Malicious agent interaction	Secure inter-agent communication. Agent reputation systems. Sandbox agents.
Flaws in Multi-Agent Collaboration Mechanisms (In multi-agent systems, deficiencies in internal collaboration mechanisms can manifest as follows: when agents make distributed decisions based on localized information, conflicts between their objectives may result in systemic failures.)	Establish a coordination and management mechanism for multi-agents.

REFERENCES

AG2. (n.d.). *UserProxyAgent.* Retrieved from AG2: https://docs.ag2.ai/0.8.7/docs/api-reference/autogen/UserProxyAgent/

AI Verify Foundation. (n.d.). *List of Datasets*. Retrieved from Moonshot: https://aiverify-foundation.github.io/moonshot/resources/datasets/

Anthropic. (18 Jun, 2025). *Model Context Protocol, Core architecture.* Retrieved from Model Context Protocol: https://modelcontextprotocol.io/docs/concepts/architecture

Anthropic. (18 Jun, 2025). *Model Context Protocol, Introduction.* Retrieved from Model Context Protocol: https://modelcontextprotocol.io/introduction

Anthropic. (n.d.). *Content moderation.* Retrieved from https://docs.anthropic.com/en/docs/about-claude/use-case-guides/content-moderation

Apostrophe Technologies. (May, 2025). *sanitize-html.* Retrieved from npm: https://www.npmjs.com/package/sanitize-html

Arias, D., & Bellen, S. (7 Oct, 2021). *What Are Refresh Tokens and How to Use Them Securely, auth0.* Retrieved from auth0: https://auth0.com/blog/refresh-tokens-what-are-they-and-when-to-use-them/

Auxiliobits. (2025). *Agent-to-Agent Protocols: How Google's A2A is Shaping Future Automations?* Retrieved from Auxiliobits: https://www.auxiliobits.com/blog/agent-to-agent-protocols-how-googles-a2a-is-shaping-future-automations/#elementor-toc__heading-anchor-1

AWS. (Aug, 2025). *AWS Prescriptive Guidance: Operationalizing agentic AI on AWS.* Retrieved from AWS: https://docs.aws.amazon.com/prescriptive-guidance/latest/strategy-operationalizing-agentic-ai/introduction.html

AWS. (n.d.). *Control subnet traffic with network access control lists.* Retrieved from AWS: https://docs.aws.amazon.com/vpc/latest/userguide/vpc-network-acls.html

AWS. (n.d.). *Security best practices in IAM.* Retrieved from AWS: https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html

AWS. (n.d.). *Use temporary credentials with AWS resources.* Retrieved from AWS: https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp_use-resources.html

AWS. (n.d.). *What is AWS Secrets Manager?* Retrieved from AWS: https://docs.aws.amazon.com/secretsmanager/latest/userguide/intro.html

Besen, S., & Gutowska, A. (n.d.). *What is Agent Communication Protocol (ACP)?* Retrieved from IBM: https://www.ibm.com/think/topics/agent-communication-protocol

Cameron, A. (8 Apr, 2025). *pip-audit.* Retrieved from PyPI: https://pypi.org/project/pip-audit/

Center for Research on Foundation Models (CRFM), Stanford University. (2025). *Holistic Evaluation of Language Models (HELM).* Retrieved from https://crfm.stanford.edu/helm/

Chhikara, P., Khant, D., Aryan, S., Singh, T., & Yadav, D. (28 Apr, 2025). *Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory.* Retrieved from arxiv: https://arxiv.org/abs/2504.19413v1

CISA. (2025). *Software Bill of Materials (SBOM).* Retrieved from CISA: https://www.cisa.gov/sbom

Cloud Security Alliance. (21 Aug, 2024). *Best practices for event logging and threat detection.* Retrieved from Cloud Security Alliance: https://cloudsecurityalliance.org/resources/best-practices-for-event-logging-and-threat-detection

Cloud Security Alliance. (16 Jul, 2025). *Agentic AI Red Teaming Guide.* Retrieved from Cloud Security Alliance: https://cloudsecurityalliance.org/artifacts/agentic-ai-red-teaming-guide

Cloudflare. (n.d.). *What is mutual TLS (mTLS)?* Retrieved from Cloudflare: https://www.cloudflare.com/learning/access-management/what-is-mutual-tls/

CodeSignal. (2025). *Developing a Robust System Prompt.* Retrieved from CodeSignal: https://codesignal.com/learn/courses/building-a-chatbot-service-with-fastapi/lessons/crafting-a-robust-system-prompt-for-chatbot-interaction

Conversation-AI. (n.d.). *Enabling online conversations.* Retrieved from Perspective: https://www.perspectiveapi.com/

crewAI Inc. (n.d.). *Human-in-the-Loop (HITL) Workflows.* Retrieved from crewAI: https://docs.crewai.com/en/learn/human-in-the-loop

Cure53. (n.d.). *DOMPurify.* Retrieved from Github: https://github.com/cure53/DOMPurify

Cyber Security Agency of Singapore. (27 Jul, 2022). *Critical Information Infrastructure Supply Chain Programme Paper.* Retrieved from CSA: https://www.csa.gov.sg/resources/publications/critical-information-infrastructure-supply-chain-programme-paper

Cyber Security Agency of Singapore. (15 Oct, 2024). *Guidelines and Companion Guide on Securing AI Systems.* Retrieved from CSA: https://www.csa.gov.sg/resources/publications/guidelines-and-companion-guide-on-securing-ai-systems

Cyber Security Agency of Singapore. (5 Jun, 2025). *Responsible Vulnerability Disclosure Policy*. Retrieved from https://isomer-user-content.by.gov.sg/36/4aa60609-4481-4e7c-92eb-2728247a084f/responsible-vulnerability-disclosure-policy.pdf

Cyber Security Agency of Singapore. (20 Jan, 2025). *Supplementary references*. Retrieved from CSA: https://www.csa.gov.sg/legislation/supplementary-references

Debenedetti, E., Shumailov, I., Fan, T., Hayes, J., Carlini, N., Fabian, D., . . . Tramèr, F. (24 Jun, 2025). *Defeating Prompt Injections by Design*. Retrieved from arxiv: https://arxiv.org/abs/2503.18813

Díaz, S., Kern, C., & Olive, K. (May, 2025). *Google's Approach for Secure AI Agents*. Retrieved from Google Research: https://research.google/pubs/an-introduction-to-googles-approach-for-secure-ai-agents/

E2B. (26 Aug, 2025). *E2B*. Retrieved from GitHub: https://github.com/e2b-dev/E2B

EU AI Act Holistic AI Team. (1 Aug, 2024). *High-Risk AI Systems Under the EU AI Act*. Retrieved from EU AI Act: https://www.euaiact.com/blog/high-risk-ai-systems-under-the-eu-ai-act

Explosion. (n.d.). *Industrial-Strength Natural Language Processing*. Retrieved from spaCy: https://spacy.io/

Feldman, E. (15 Apr, 2025). *Implementing effective guardrails for AI agents*. Retrieved from The Source Gitlab: https://about.gitlab.com/the-source/ai/implementing-effective-guardrails-for-ai-agents/

Flinders, M., Smalley, I., & Schneider, J. (30 Apr, 2025). *AI fraud detection in banking*. Retrieved from IBM: https://www.ibm.com/think/topics/ai-fraud-detection-in-banking

Fortinet. (2025). *What Is A Message Authentication Code?* Retrieved from Fortinet: https://www.fortinet.com/resources/cyberglossary/message-authentication-code

Gabarda, F. C. (1 Jul, 2025). *Model Context Protocol (MCP): Understanding security risks and controls*. Retrieved from Red Hat Blog: https://www.redhat.com/en/blog/model-context-protocol-mcp-understanding-security-risks-and-controls

GitHub. (28 Nov, 2022). *Rate limits for the REST API*. Retrieved from GitHub Docs: https://docs.github.com/en/rest/using-the-rest-api/rate-limits-for-the-rest-api?apiVersion=2022-11-28

GitHub. (n.d.). *Dependabot quickstart guide*. Retrieved from GitHub Docs: https://docs.github.com/en/code-security/getting-started/dependabot-quickstart-guide

GitLab. (n.d.). *Dependency Scanning*. Retrieved from GitLab Docs: https://docs.gitlab.com/user/application_security/dependency_scanning/

GitLab. (n.d.). *What is version control?* Retrieved from GitLab: https://about.gitlab.com/topics/version-control/

Gittlen, S. (2 May, 2024). *The ultimate guide to SBOMs.* Retrieved from GitLab: https://about.gitlab.com/blog/the-ultimate-guide-to-sboms/

Google. (n.d.). *About IAM authentication.* Retrieved from Google Cloud: https://cloud.google.com/memorystore/docs/valkey/about-iam-auth

Google. (n.d.). *Cutom Search JSON API.* Retrieved from https://developers.google.com/custom-search/v1/overview

Google LLC. (9 Apr, 2025). *A2A and MCP: Complementary Protocols for Agentic Systems.* Retrieved from Agent2Agent (A2A) Protocol: https://a2aproject.github.io/A2A/latest/topics/a2a-and-mcp/#how-a2a-and-mcp-complement-each-other

Google LLC. (9 Apr, 2025). *What is A2A?* Retrieved from Agent2Agent (A2A) Protocol: https://a2aproject.github.io/A2A/latest/topics/what-is-a2a/

Google. (n.d.). *Secret Manager overview.* Retrieved from Google Cloud: https://cloud.google.com/secret-manager/docs/overview

GovTech Singapore (AI Practice). (Jul, 2025). *Agentic Risk & Capability Framework.* Retrieved from https://govtech-responsibleai.github.io/agentic-risk-capability-framework/

Guardrails AI. (n.d.). *Guardrails AI.* Retrieved from Github: https://github.com/guardrails-ai/guardrails

Harang, R., & Sablotny, M. (25 Feb, 2025). *Agentic Autonomy Levels and Security*. Retrieved from NVIDIA DEVELOPER: https://developer.nvidia.com/blog/agentic-autonomy-levels-and-security/

HashiCorp. (n.d.). *Vault.* Retrieved from GitHub: https://github.com/hashicorp/vault

Helicone Inc. (n.d.). *Helicone.* Retrieved from GitHub: https://github.com/Helicone/helicone

Hou, X., Zhao, Y., Wang, S., & Wang, H. (Apr, 2025). *Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions.* Retrieved from https://arxiv.org/pdf/2503.23278

Huang, K. (22 Dec, 2024). *7 Layered Agentic AI Reference Architecture.* Retrieved from Medium: https://kenhuangus.medium.com/7-layered-agentic-ai-reference-architecture-20276f83b7ee

Huang, K. (02 Jun, 2025). *Agentic AI Threat Modeling Framework: MAESTRO*. Retrieved from Cloud Security Alliance: https://cloudsecurityalliance.org/blog/2025/02/06/agentic-ai-threat-modeling-framework-maestro

Huang, K. (30 Apr, 2025). *Threat Modeling Google's A2A Protocol with the MAESTRO Framework*. Retrieved from Cloud Security Alliance: https://cloudsecurityalliance.org/blog/2025/04/30/threat-modeling-google-s-a2a-protocol-with-the-maestro-framework

Huang, K., Narajala, V. S., Yeoh, J., Ross, J., Raskar, R., Harkati, Y., . . . Hughes, C. (28 May, 2025). *A Novel Zero-Trust Identity Framework for Agentic AI: Decentralized Authentication and Fine-Grained Access Control*. Retrieved from arxiv: https://arxiv.org/abs/2505.19301

Hugging Face. (n.d.). *Daily Papers: Instruction Following Score*. Retrieved from https://huggingface.co/papers?q=Instruction%20Following%20Score%20(IFS)

Hugging Face. (n.d.). *Model Cards*. Retrieved from Hugging Face: https://huggingface.co/docs/hub/en/model-cards

Hugging Face. (n.d.). *Pickle Scanning*. Retrieved from Hugging Face: https://huggingface.co/docs/hub/security-pickle

IETF OAuth Working Group. (n.d.). *OAuth Scopes*. Retrieved from OAuth 2.0: https://oauth.net/2/scope/

Invariant. (1 Apr, 2025). *MCP Security Notification: Tool Poisoning Attacks*. Retrieved from Invariantlabs: https://invariantlabs.ai/blog/mcp-security-notification-tool-poisoning-attacks

Jambrecic, R. (7 Jan, 2025). *Tools Dependency Injection*. Retrieved from AG2: https://docs.ag2.ai/latest/docs/blog/2025/01/07/Tools-Dependency-Injection/

Jarvis, C. (19 Dec, 2023). *How to implement LLM guardrails*. Retrieved from OpenAI Cookbook: https://cookbook.openai.com/examples/how_to_use_guardrails

Jin, X., Guo, Z., Zhang, P., Lu, S., Dai, W., Nujibieke, . . . Li, G. (2 Feb, 2025). *Bridging Minds and Machines: Agents with Human-in-the-Loop – Frontier Research, Real-World Impact, and Tomorrow's Possibilities*. Retrieved from Camel-AI: https://www.camel-ai.org/blogs/human-in-the-loop-ai-camel-integration

Kartha, V. (3 May, 2024). *Self-Reflecting AI Agents Using LangChain*. Retrieved from Medium: https://vijaykumarkartha.medium.com/self-reflecting-ai-agents-using-langchain-d3a93684da92

Kumar, A., Roh, J., Naseh, A., Karpinska, M., Iyyer, M., Houmansadr, A., & Bagdasarian, E. (5 Feb, 2025). *OverThink: Slowdown Attacks on Reasoning LLMs*. Retrieved from arxiv: https://arxiv.org/abs/2502.02542

LangChain. (2025). *How to pass run time values to tools*. Retrieved from LangChain: https://python.langchain.com/docs/how_to/tool_runtime/

LangChain. (2025). *LangMem*. Retrieved from LangGraph: https://langchain-ai.github.io/langmem/

LangChain. (n.d.). *E2B Data Analysis*. Retrieved from LangChain: https://python.langchain.com/docs/integrations/tools/e2b_data_analysis/

LangChain. (n.d.). *LangGraph interrupt: Making it easier to build human-in-the-loop agents with interrupt*. Retrieved from LangChain: https://blog.langchain.com/making-it-easier-to-build-human-in-the-loop-agents-with-interrupt/

Langfuse. (n.d.). *Open Source LLM Engineering Platform*. Retrieved from Langfuse: https://langfuse.com/

LangSmith. (n.d.). *Ship agents with confidence*. Retrieved from LangSmith: https://www.langchain.com/langsmith

Liu, X., Yu, H., Zhang, H., Xu, Y., Lei, X., Lai, H., . . . Tang, J. (25 Oct, 2023). *AgentBench: Evaluating LLMs as Agents*. Retrieved from arxiv: https://arxiv.org/abs/2308.03688

Lucas, J. (16 Dec, 2024). *Sandboxing Agentic AI Workflows with WebAssembly*. Retrieved from NVIDIA Developer: https://developer.nvidia.com/blog/sandboxing-agentic-ai-workflows-with-webassembly/

Meadows, J., & Chang, A. (27 Mar, 2024). *How to choose a known, trusted supplier for open source software*. Retrieved from Google Cloud: https://cloud.google.com/blog/products/identity-security/how-to-choose-a-known-trusted-supplier-for-open-source-software

Meta Llama. (n.d.). *Purple Llama*. Retrieved from Github: https://github.com/meta-llama/PurpleLlama

Microsoft AI Red Team. (2024). *PyRIT*. Retrieved from https://azure.github.io/PyRIT/

Microsoft Azure. (5 Jul, 2024). *What is an IP based access control list (ACL)?* Retrieved from Microsoft Learn: https://learn.microsoft.com/en-us/azure/virtual-network/ip-based-access-control-list-overview

Microsoft. (n.d.). *Presidio: Data Protection and De-identification SDK*. Retrieved from Microsoft Presidio: https://microsoft.github.io/presidio/

Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., . . . Gebru, T. (14 Jan, 2019). *Model Cards for Model Reporting*. Retrieved from arxiv: https://arxiv.org/abs/1810.03993

MITRE. (n.d.). *ATLAS Matrix*. Retrieved from MITRE ATLAS: https://atlas.mitre.org/matrices/ATLAS

MITRE. (n.d.). *Supply Chain Security Framework*. Retrieved from MITRE System of Trust: https://sot.mitre.org/framework/system_of_trust.html

Mu, N., Lu, J., Lavery, M., & Wagner, D. (15 Feb, 2025). *A Closer Look at System Prompt Robustness*. Retrieved from https://arxiv.org/abs/2502.12197

Murúa, T. (1 May, 2025). *RAG and the value of grounding*. Retrieved from elastic search labs: https://www.elastic.co/search-labs/blog/grounding-rag

National Security Agency. (5 Mar, 2024). *Advancing Zero Trust Maturity Throughout the Network and Environment Pillar*. Retrieved from NSA: https://media.defense.gov/2024/Mar/05/2003405462/-1/-1/0/CSI-ZERO-TRUST-NETWORK-ENVIRONMENT-PILLAR.PDF

Nelson, A., Rekhi, S., Souppaya, M., & Scarfone, K. (n.d.). *Special Publication 800-61r3 Incident Response Recommendationsand Considerations for CybersecurityRisk Management*. Retrieved from NIST: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r3.pdf

NIST. (14 Jan, 2025). *Cryptographic Standards and Guidelines*. Retrieved from NIST: https://csrc.nist.gov/projects/cryptographic-standards-and-guidelines

NVIDIA. (2025). *About NeMo Guardrails*. Retrieved from NVIDIA: https://docs.nvidia.com/nemo/guardrails/latest/index.html#

NVIDIA. (n.d.). *garak, LLM vulnerability scanner*. Retrieved from Github: https://github.com/NVIDIA/garak

OpenAI. (2025). *Guardrails*. Retrieved from OpenAI Agents SDK: https://openai.github.io/openai-agents-python/guardrails/

OpenAI. (n.d.). *Moderation*. Retrieved from https://platform.openai.com/docs/guides/moderation

OpenJS Foundation. (n.d.). *ESLint*. Retrieved from GitHub: https://github.com/eslint/eslint

OWASP. (22 Apr, 2025). *OWASP Gen AI Security Project - Multi-Agentic system Threat Modelling Guide*. Retrieved from https://genai.owasp.org/resource/multi-agentic-system-threat-modeling-guide-v1-0/

OWASP. (28 Jun, 2025). *OWASP Gen AI Security Project - Securing Agentic Applications Guide*. Retrieved from https://genai.owasp.org/resource/securing-agentic-applications-guide-1-0/

OWASP. (17 Feb, 2025). *OWASP Top 10 for LLMs - Agentic AI - Threats and Mitigations*. Retrieved from https://genai.owasp.org/resource/agentic-ai-threats-and-mitigations/

OWASP. (23 Jan, 2025). *OWASP Top 10 for LLMs - GenAI Red Teaming Guide*. Retrieved from https://genai.owasp.org/resource/genai-red-teaming-guide/

OWASP. (n.d.). *Authentication Cheat Sheet*. Retrieved from OWASP Cheat Sheet Series: https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html

OWASP. (n.d.). *Authorization Cheat Sheet*. Retrieved from OWASP Cheat Sheet Series: https://cheatsheetseries.owasp.org/cheatsheets/Authorization_Cheat_Sheet.html

OWASP. (n.d.). *Content Security Policy Cheat Sheet.* Retrieved from OWASP Cheat Sheet Series: https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html

OWASP. (n.d.). *Docker Security Cheat Sheet.* Retrieved from OWASP Cheat Sheet Series: https://cheatsheetseries.owasp.org/cheatsheets/Docker_Security_Cheat_Sheet.html

OWASP. (n.d.). *File Upload Cheat Sheet.* Retrieved from OWASP Cheat Sheet Series: https://cheatsheetseries.owasp.org/cheatsheets/File_Upload_Cheat_Sheet.html

OWASP. (n.d.). *Input Validation Cheat Sheet.* Retrieved from OWASP Cheat Sheet Series: https://cheatsheetseries.owasp.org/cheatsheets/Input_Validation_Cheat_Sheet.html

OWASP. (n.d.). *LLM Prompt Injection Prevention Cheat Sheet.* Retrieved from OWASP Cheat Sheet Series: https://cheatsheetseries.owasp.org/cheatsheets/LLM_Prompt_Injection_Prevention_Cheat_Sheet.html

OWASP. (n.d.). *Secrets Management Cheat Sheet.* Retrieved from OWASP Cheat Sheet Series: https://cheatsheetseries.owasp.org/cheatsheets/Secrets_Management_Cheat_Sheet.html

OWASP. (n.d.). *XSS Filter Evasion Cheat Sheet.* Retrieved from OWASP Cheat Sheet Series: https://cheatsheetseries.owasp.org/cheatsheets/XSS_Filter_Evasion_Cheat_Sheet.html

PagerDuty. (n.d.). *Incident Response.* Retrieved from PagerDuty: https://response.pagerduty.com/

Perrone, P. (15 Apr, 2025). *MCP Is a Security Nightmare — Here's How the Agent Security Framework Fixes It.* Retrieved from Medium: https://medium.com/data-science-collective/mcp-is-a-security-nightmare-heres-how-the-agent-security-framework-fixes-it-fd419fdfaf4e

Perrot, C., Tanke, M. L., Roy, M., & Sachs, R. (9 Apr, 2025). *Implement human-in-the-loop confirmation with Amazon Bedrock Agents.* Retrieved from AWS: https://aws.amazon.com/blogs/machine-learning/implement-human-in-the-loop-confirmation-with-amazon-bedrock-agents/

Personal Data Protection Commission Singapore. (1 Mar, 2024). *Advisory Guidelines on use of Personal Data in AI Recommendation and Decision Systems.* Retrieved from PDPC: https://www.pdpc.gov.sg/guidelines-and-consultation/2024/02/advisory-guidelines-on-use-of-personal-data-in-ai-recommendation-and-decision-systems

Personal Data Protection Commission Singapore. (24 Jul, 2024). *Guide to Basic Anonymisation*. Retrieved from PDPC: https://www.pdpc.gov.sg/-/media/files/pdpc/pdf-files/advisory-guidelines/guide-to-basic-anonymisation-(updated-24-july-2024).pdf

Personal Data Protection Commission Singapore. (14 Dec, 2024). *Guide to Data Protection Practices for ICT Systems*. Retrieved from PDPC: https://www.pdpc.gov.sg/-/media/files/pdpc/pdf-files/other-guides/tech-omnibus/guide-to-data-protection-practices-for-ict-systems.pdf

Promptfoo. (n.d.). *Promptfoo: LLM evals & red teaming*. Retrieved from Github: https://github.com/promptfoo/promptfoo

Python Code Quality Authority. (n.d.). *Bandit*. Retrieved from https://bandit.readthedocs.io/en/latest/

Rasmussen, P., Paliychuk, P., Beauvais, T., Ryan, J., & Chalef, D. (20 Jan, 2025). *Zep: A Temporal Knowledge Graph Architecture for Agent Memory*. Retrieved from arxiv: https://arxiv.org/abs/2501.13956

Rehberger, J. (2 May, 2025). *MCP: Untrusted Servers and Confused Clients, Plus a Sneaky Exploit*. Retrieved from Embrace The Red: https://embracethered.com/blog/posts/2025/model-context-protocol-security-risks-and-exploits/

Rehberger, J. (n.d.). *Trust No AI: Prompt Injection Along The CIA Security Triad*. Retrieved from https://arxiv.org/pdf/2412.06090

Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (Aug, 2020). *Special Publication 800-207 Zero Trust Architecture*. Retrieved from NIST: https://nvlpubs.nist.gov/nistpubs/specialpublications/NIST.SP.800-207.pdf

Sapkota, R., Roumeliotis, K. I., & Karkee, M. (28 May, 2025). *AI Agents vs. Agentic AI: A Conceptual Taxonomy, Applications and Challenges*. Retrieved from arxiv: https://arxiv.org/abs/2505.10468

Scarfone, K., Souppaya, M., Cody, A., & Orebaugh, A. (n.d.). *Special Publication 800-115 Technical Guide to Information Security Testing and Assessment*. Retrieved from NIST: https://csrc.nist.gov/pubs/sp/800/115/final

Semgrep, Inc. (n.d.). *Semgrep*. Retrieved from GitHub: https://github.com/semgrep/semgrep

Shah, D. (4 Jun, 2025). *Introduction to Training Data Poisoning: A Beginner's Guide*. Retrieved from Lakera: https://www.lakera.ai/blog/training-data-poisoning

Snyk. (Jun, 2025). *Snyk Open Source*. Retrieved from Snyk User Docs: https://docs.snyk.io/scan-with-snyk/snyk-open-source

Souppaya, M., Scarfone, K., & Dodson, D. (Feb, 2022). *Special Publication 800-218 Secure Software Development Framework (SSDF) Version 1.1*. Retrieved from NIST: https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-218.pdf

Stryker, C. (2025). *What is agentic AI?* Retrieved from IBM: https://www.ibm.com/think/topics/agentic-ai

Surapaneni, R., Jha, M., Vakoc, M., & Segal, T. (9 Apr, 2025). *Announcing the Agent2Agent Protocol (A2A)*. Retrieved from Google for Developers: https://developers.googleblog.com/en/a2a-a-new-era-of-agent-interoperability/

The Linux Foundation. (n.d.). *Supply-chain Levels for Software Artifacts*. Retrieved from SLSA: https://slsa.dev/

The OWASP Foundation. (July, 2017). *OWASP Code Review Guide*. Retrieved from OWASP: https://owasp.org/www-project-code-review-guide/

The Vulnerable MCP Project. (2025). *The Vulnerable MCP Project*. Retrieved from https://vulnerablemcp.info/

tl;dr sec. (Feb, 2025). *prompt-injection-defenses*. Retrieved from GitHub: https://github.com/tldrsec/prompt-injection-defenses

traceloop. (n.d.). *OpenLLMetry*. Retrieved from GitHub: https://github.com/traceloop/openllmetry

UK AI Secuity Institute, Arcadia Impact, Vector Institute. (n.d.). *Inspect Evals*. Retrieved from https://ukgovernmentbeis.github.io/inspect_evals/

UK National Cyber Security Centre. (12 Oct, 2023). *Supply Chain Guidance*. Retrieved from NCSC: https://www.ncsc.gov.uk/collection/supply-chain/guidance

UK National Cyber Security Centre. (7 Nov, 2024). *Vulnerability Disclosure Toolkit*. Retrieved from NCSC: https://www.ncsc.gov.uk/information/vulnerability-disclosure-toolkit

Ul Muram, F., Tran, H., & Zdun, U. (1 Apr, 2017). *Systematic Review of Software Behavioral Model Consistency Checking*. Retrieved from https://www.researchgate.net/publication/316938485_Systematic_Review_of_Software_Behavioral_Model_Consistency_Checking

University of the Sunshine Coast Australia. (n.d.). *What are credible sources?* Retrieved from https://libguides.usc.edu.au/credible/web

Wang, C. L., Singhal, T., Kelkar, A., & Tuo, J. (8 Aug, 2025). *MI9 - Agent Intelligence Protocol: Runtime Governance for Agentic AI Systems*. Retrieved from arxiv: https://arxiv.org/abs/2508.03858

Wickramasinghe, S. (18 Mar, 2025). *IT & System Availability + High Availability: The Ultimate Guide*. Retrieved from Splunk Blogs: https://www.splunk.com/en_us/blog/learn/availability.html

Xu, F. F., Song, Y., Li, B., Tang, Y., Jain, K., Bao, M., . . . Neubig, G. (19 May, 2025). *TheAgentCompany: Benchmarking LLM Agents on Consequential Real World Tasks*. Retrieved from https://the-agent-company.com/

Yuan, Y., Jiao, W., Wang, W., Huang, J.-t., Xu, J., Liang, T., . . . Tu, Z. (23 May, 2025). *Refuse Whenever You Feel Unsafe: Improving Safety in LLMs via Decoupled Refusal Training*. Retrieved from https://arxiv.org/abs/2407.09121

Zaharia, M. A., Chen, A., Davidson, A., Ghodsi, A., Hong, S. A., Konwinski, A., . . . Zumar, C. (2018). *Accelerating the Machine Learning Lifecycle with MLflow*. Retrieved from GitHub: https://github.com/mlflow/mlflow

Zhou, S., Xu, F. F., Zhu, H., Zhou, X., Lo, R., Sridhar, A., . . . Neubig, G. (16 Apr, 2024). *WebArena: A Realistic Web Environment for Building Autonomous Agents*. Retrieved from https://webarena.dev/

GUIDELINES ON **SECURING AI SYSTEMS**

CYBER SECURITY AGENCY OF SINGAPORE

OCTOBER 2024

CSA SINGAPORE

TABLE OF CONTENTS

1. INTRODUCTION 3
   1.1. PURPOSE AND SCOPE OF THIS DOCUMENT 4
2. UNDERSTANDING AI THREATS 5
3. SECURING AI 7
   3.1. TAKE A LIFECYCLE APPROACH 7
   3.2. START WITH A RISK ASSESSMENT 8
   3.3. GUIDELINES FOR SECURING AI SYSTEMS 10

GLOSSARY 14

ANNEX A 18

1. INTRODUCTION

Artificial Intelligence (AI) poses benefits for economy, society, and national security. It has the potential to drive efficiency and innovation in almost every sector – from commerce and healthcare to transportation and cybersecurity.

To reap the benefits of AI, users must have confidence that the AI will behave as designed, and outcomes are safe and secure. However, in addition to safety risks, AI systems can be vulnerable to adversarial attacks, where malicious actors intentionally manipulate or deceive the AI system. **The adoption of AI can introduce or exacerbate existing cybersecurity risks to enterprise systems. These can lead to risks such as data leakage or data breaches, or result in harmful or otherwise undesired model outcomes.**

As such, **as a key principle, AI should be secure by design and secure by default**, as with all software systems. This will enable system owners to manage security risks upstream. This will complement other controls and mitigation strategies that system owners may take to address the safety of AI, and other attendant considerations such as fairness or transparency, which are not addressed here.

The Cyber Security Agency of Singapore (CSA) has developed **Guidelines on Securing AI Systems** for system owners to secure the use of AI throughout its lifecycle. As AI is increasingly integrated into enterprise systems, security should be considered holistically at the system level. As such, these guidelines should be used alongside existing security best practices and requirements for IT environments. While these guidelines are not mandatory, we strongly encourage system owners to consider these key principles, so that they can make informed decisions on their adoption of AI vis-à-vis the potential risks.

AI security is a developing field of work, and mitigation controls continue to evolve. As such, CSA is also collaborating with AI and cybersecurity practitioners on the **Companion Guide on Securing AI Systems**. This is intended as a community-driven resource, and the Companion Guide complements the Guidelines as a useful reference containing practical measures and controls that system owners may consider as part of adopting the Guidelines, depending on their use case. The Companion Guide is not mandatory, prescriptive, or exhaustive. As the field of AI security continues to evolve rapidly, the Companion Guide will be updated to account for material developments in this space.

1.1. PURPOSE AND SCOPE OF THIS DOCUMENT

Purpose

These guidelines are designed to support systems owners that are adopting, or considering the adoption of AI systems. It identifies potential security risks associated with the use of AI and sets out guidelines for mitigating security risks at each stage of the AI lifecycle.

This document can be read together with the Companion Guide on Securing AI Systems, which provides an informative compilation of practical security control measures, that system owners may consider in implementing these guidelines.

Scope

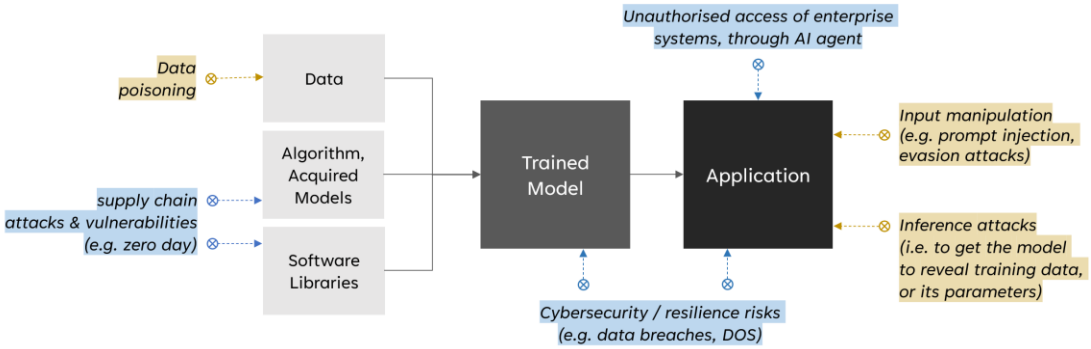
These guidelines address the cybersecurity risks to AI systems. It does not seek to address AI safety, or other common attendant considerations for AI such as fairness, transparency or inclusion, or cybersecurity risks introduced by AI systems, although some of the recommended actions may overlap. It also does not address the misuse of AI in cyberattacks (AI-enabled malware), mis/disinformation, and scams (deepfakes).

2. UNDERSTANDING AI THREATS

AI is a type of software system, and is itself vulnerable to cyber threats, while also posing a new attack surface for the broader enterprise system that it is integrated to, or interfaces with. As such, securing AI is in addition to practising good 'classical' cybersecurity hygiene.

Securing an AI system introduces new challenges that may be unfamiliar in traditional IT systems. In addition to classical cybersecurity risks, the AI itself is vulnerable to novel attacks such as Adversarial Machine Learning (ML) that set out to distort the model's behaviour. For more details on the security threats to AI, refer to **Annex A**.

Figure 1. Classical and AI-specific risks of AI systems– diagram adapted from OWASP[1]



[1] Threats overview - https://owaspai.org/docs/ai_security_overview/

CLASSICAL CYBERSECURITY RISKS TO AI SYSTEMS

AI systems require vast amounts of data for training; some also require importing external models and libraries. If inadequately secured, AI systems can be undermined by **supply chain attacks**, or may be **susceptible to intrusion or unauthorised access**, through vulnerabilities in the AI model or the underlying IT infrastructure. In addition, **organisations and users risk losing the ability to access and use AI tools** if there are disruptions to cloud services, data centre operations, or other digital infrastructure (e.g. through Denial of Service attacks), this could in turn **disable systems that depend on AI tools to function**.

ADVERSARIAL MACHINE LEARNING

Malicious actors may use novel Adversarial ML techniques to attack AI models and data, influencing machine learning models to produce inaccurate, biased, or harmful output; and/or reveal confidential information. Adversarial ML[2] attacks include: *data poisoning* (injecting malicious or corrupted data into training data sets) or *evasion attacks* (on trained models) to **distort outcomes**, *inference attacks* or *extraction attacks* (probing the model) to **expose sensitive or restricted data,** or to **steal the model.**

[2] A Taxonomy and Terminology of Attacks and Mitigations https://csrc.nist.gov/pubs/ai/100/2/e2023/final. The MITRE ATLAS is a useful reference to understand and situate classical cybersecurity risks from Adversarial ML.

3. SECURING AI

The security of AI is a widely cited concern, but this field of work is still relatively nascent. While practitioners continue to grow the body of research and resources on the security threats to AI, these guidelines lay out key considerations that system owners should take to support secure adoption of AI. Given the rapid speed of AI development, system owners should continue to apprise themselves on the latest developments in AI security, and refresh their risk management strategies accordingly.

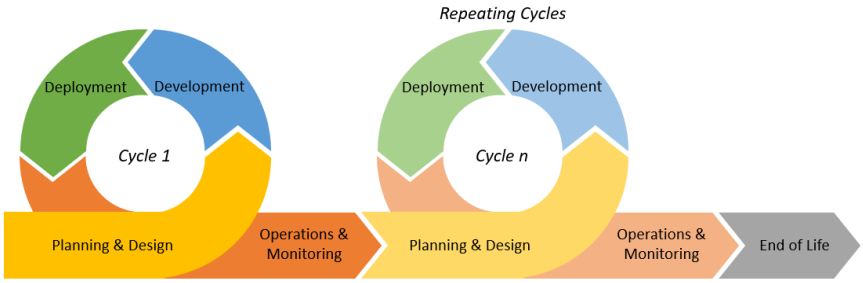
3.1. TAKE A LIFECYCLE APPROACH

There are **five key stages** – Planning and Design, Development, Deployment, Operations and Maintenance, and End of Life.

As with good cybersecurity practice, CSA recommends that system owners take a lifecycle approach to consider security risks. Hardening only the AI model is insufficient to ensure a holistic defence against AI related threats. All stakeholders involved across the lifecycle of an AI system should seek to better understand the security threats and their potential impact on the desired outcomes of the AI system, and what decisions or trade-offs will need to be made.

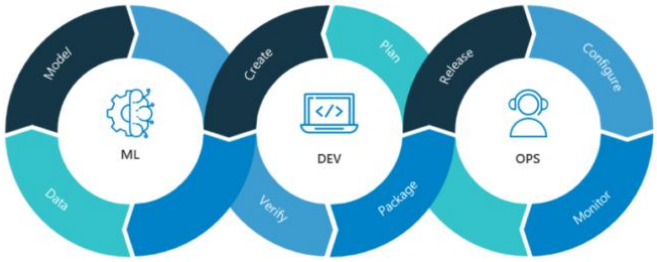
The AI lifecycle represents the iterative process of designing an AI solution to meet a business or operational need. As such, system owners will likely revisit the planning and design, development, and deployment steps in the lifecycle many times in the delivery of an AI solution.

Figure 2: AI System Development Lifecycle (AI SDLC)



Some organisations may have implemented the Machine Learning Operations (ML Ops) pipeline, which may not map exactly to the AI SDLC. Nonetheless, ML Ops teams that run a dev ops pipeline comprising ML Design, Development and Operation stages (similar to Figure 3), will find the guidelines across the AI SDLC's stages of Planning & Design, Development, Deployment and Operations relevant.

Figure 3: Example of ML-DevOps (source: Nvidia blog)



3.2. START WITH A RISK ASSESSMENT

Given the diversity of AI use cases, there is no one-size-fits-all solution to implementing security. As such, effective cybersecurity starts with conducting a risk assessment. This will enable organisations to identify potential risks, priorities, and subsequently, the appropriate risk management strategies.

A fundamental difference between AI and traditional software is that while traditional software relies on static rules and explicit programming, AI uses machine learning and neural networks to autonomously learn and make decisions without the need for detailed instructions for each task. As such, organisations should consider conducting risk assessments more frequently than for conventional systems, even if they generally base their risk assessment approach on existing governance and policies. These assessments may also be supplemented by continuous monitoring and a strong feedback loop.

We recommend these **four steps** to tailor a systematic defence plan that best addresses your organisation's highest priority risks – protecting the things you care about the most.

STEP 1

Conduct risk assessment, focusing on security risks to AI systems

Conduct a risk assessment, focusing on security risks related to AI systems, either based on best practices or your organisation's existing Enterprise Risk Assessment/Management Framework.

Risk assessment can be done with reference to CSA published guides, if applicable:

- Guide To Cyber Threat Modelling
- Guide To Conducting Cybersecurity Risk Assessment for Critical Information Infrastructure

STEP 2

Prioritise areas to address based on risk/impact/resources

Prioritise which risks to address, based on risk level, impact, and available resources.

STEP 3

Identify and implement the relevant actions to secure the AI system

Identify relevant actions and control measures to secure the AI system, such as by referencing those outlined in the **Companion Guide on Securing AI Systems** and implement these across the AI life cycle.

STEP 4

Evaluate residual risks for mitigation or acceptance

Evaluate the residual risk after implementing security measures for the AI system to inform decisions about accepting or addressing residual risks.

3.3. GUIDELINES FOR SECURING AI SYSTEMS

These guidelines apply across the various lifecycle stages of the AI system. System owners should read these as key issues to consider in securing their adoption of AI. In view of the diversity of use cases and developments in AI security, these guidelines do not provide prescriptive controls or requirements.

System owners should apply these to their specific context, and can reference the Companion Guide to Securing AI systems for potential controls.

1. PLANNING AND DESIGN

1.1. Raise awareness and competency on security risks

Organisations should understand the potential security risks posed by AI, in order to make informed decisions about adoption. Provide adequate training and guidance on the security risks of AI to all personnel, including developers, system owners and senior leaders.

1.2. Conduct security risk assessments

Risk management strategies should be informed by a security risk assessment, which will help to determine key risks and priorities. Apply a holistic process to model threats and risks to an AI system, in accordance with relevant industry standards/best practices.

2. DEVELOPMENT

2.1. Secure the supply chain

The AI supply chain includes (but is not limited to) the training data, models, APIs, and software libraries. Each of these components may introduce new vulnerabilities (e.g, models may carry malware encoded as model parameters that could enable attackers to extract and inject malicious software onto user machines). Assess and monitor potential security risks of the AI system's supply chain across its life cycle. Ensure that suppliers adhere to security policies and internationally recognised standards, or that risks are otherwise appropriately managed. Consider evaluating supply chain components (e.g. through Software Bills of Material [SBOM], code checking, or against vulnerability databases).

2.2. Consider security benefits and trade-offs when selecting the appropriate model to use

Different AI models (e.g. machine learning, deep learning, generative) pose unique characteristics and risks (e.g. LLMs can be vulnerable to input manipulation attacks) and as such require different security measures. When developing or selecting an appropriate AI model for your system, consider factors which may affect its security (such as complexity, explainability, interpretability, and sensitivity of training data).

2.3. Identify, track and protect AI-related assets

As AI systems become increasingly integrated into business operations, they will become part of an organisation's strategic assets and should be secured accordingly. Otherwise, sensitive data, intellectual property and organisational assets are at risk of potential threats and breaches. Understand the value of AI-related assets, including models, data, prompts, logs and assessments. Have processes to track, authenticate, version control, and secure assets.

2.4. Secure the AI development environment

AI models require access to large amounts of training data, and an insecure development environment can introduce risks of data breaches (e.g. exposure of Personally Identifiable Information or confidential business information). Insecure development can also make AI models vulnerable to attacks (e.g. poisoning) that result in compromised model behaviour, or expose models and other intellectual property to theft, unauthorised replication or misuse. Apply standard infrastructure security principles, such as implementing appropriate access controls and logging/monitoring, segregation of environments, and secure-by-default configurations.

3. DEPLOYMENT

3.1. Secure the deployment infrastructure and environment of AI systems

Similar considerations as with 2.4 “Secure the AI development environment”. Apply standard infrastructure security principles, such as access controls and logging/monitoring, segregation of environments, secure-by-default configurations, and firewalls.

3.2. Establish incident management procedures

AI systems are complex and adaptive, and this can sometimes result in unpredictable behaviour. Given the diversity in AI use cases, incidents can range from minor issues such as malfunctioning chat bots to critical outcomes such as disruption in the operation of critical infrastructure. System owners should put in place appropriate incident response, escalation and remediation plans.

3.3. Release AI systems responsibly

AI systems can be vulnerable to the risks described above, including misuse, data breaches, and model manipulation. These have impact on the trust and confidence of users, and may have reputational implications for organisations. A good practice is to release models, applications or systems only after subjecting them to appropriate and effective security checks and evaluation.

4. OPERATIONS AND MAINTENANCE

4.1. Monitor AI system inputs

AI systems are dynamic and adaptive to input. There have already been real-life incidents, in which users/ attackers have deliberately crafted input to trick AI systems into making incorrect or unintended decisions. AI system owners may wish to monitor and log inputs to the AI system, such as queries, prompts and requests, as third-party providers may not do so due to privacy reasons. Proper logging allows for compliance, audit, investigation and remediation.

4.2. Monitor AI system outputs and behaviour

AI systems can break or degrade in production phase. Monitoring models after deployment will make sure that they are performing as intended, and alert system owners to potential issues (whether caused by adversarial attacks or otherwise). Operators should monitor for anomalous behaviour that might indicate intrusions, compromise, or data drift.

4.3. Adopt a secure-by-design approach to updates and continuous learning

Changes to the data and model can lead to changes in behaviour. System owners should ensure that risks associated to model updates have been considered and appropriately managed.

4.4. Establish a vulnerability disclosure process

Even with monitoring mechanisms in place, the adaptive nature of AI can make it challenging to detect attacks and unintended behaviour. There should be a feedback process for users to share any findings of concern, which might uncover potential vulnerabilities to the system.

5. END OF LIFE

5.1. Ensure proper data and model disposal

As models are trained on large amounts of training data (incl. potentially confidential information), improper disposal can lead to incidents such as data breaches. There should be proper and secure disposal/destruction of data and models in accordance with relevant industry standards or regulations.

GLOSSARY

Term	Brief description
AI system	Artificial Intelligence. A machine-based system that for explicit or implicit objectives, infers, from the input it receives, how to generate outputs such as predictions, content, recommendations, or decisions that can influence physical or virtual environments. Different AI systems vary in their levels of autonomy and adaptiveness after deployment.
Adversarial Machine Learning	The process of extracting information about the behaviour and characteristics of an ML system and/or learning how to manipulate the inputs into an ML system in order to obtain a preferred outcome.
Anomaly Detection	The identification of observations, events or data points that deviate from what is usual, standard, or expected, making them inconsistent with the rest of data.
API	Application Programming Interface. A set of protocols that determine how two software applications will interact with each other.
Backdoor attack	A backdoor attack is when an attacker subtly alters AI models during training, causing unintended behaviour under certain triggers.
Chatbot	A software application that is designed to imitate human conversation through text or voice commands
Computer Vision	An interdisciplinary field of science and technology that focuses on how computers can gain understanding from images and videos.
Data Breach	Data Breach occurs when a threat actor gains unauthorised access to sensitive/confidential data.
Data Integrity	The property that data has not been altered in an unauthorised manner. Data integrity covers data in storage, during processing, and while in transit.

Data Leakage	Unintentional exposure of sensitive, protected, or confidential information outside its intended environment.
Data Loss Prevention	A system's ability to identify, monitor, and protect data in use (e.g., endpoint actions), data in motion (e.g., network actions), and data at rest (e.g., data storage) through deep packet content inspection, and contextual security analysis of transaction (e.g., attributes of originator, data object, medium, timing, recipient/destination, etc.) within a centralised management framework.
Data Poisoning	Control a model with training data modifications.
Data Science	An interdisciplinary field of technology that uses algorithms and processes to gather and analyse large amounts of data to uncover patterns and insights that inform business decisions.
Deep Learning	A function of AI that imitates the human brain by learning from how it structures and processes information to make decisions. Instead of relying on an algorithm that can only perform one specific task, this subset of machine learning can learn from unstructured data without supervision.
Defence-in-Depth	Defence in depth is a strategy that leverages multiple security measures to protect an organization's assets. The thinking is that if one line of defence is compromised, additional layers exist as a backup to ensure that threats are stopped along the way.
Evasion attack	Crafting input to AI in order to mislead it into performing its task incorrectly.
Extraction attack	Copy or steal an AI model by appropriately sampling the input space and observing outputs to build a surrogate model that behaves similarly.
Generative AI	A type of machine learning that focuses on creating new data, including text, video, code and images. A generative AI system is trained using large amounts of data, so that it can find patterns for generating new content.
Guardrails	Restrictions and rules placed on AI systems to make sure that they handle data appropriately and don't generate unethical content.

Hallucination	An incorrect response from an AI system, or false information in an output that is presented as factual information.
Image Recognition	Image recognition is the process of identifying an object, person, place, or text in an image or video.
LLM	Large Language Model. A type of AI model that processes and generates human-like text. LLMs are specifically trained on large data sets of natural language to generate human-like output.
ML	Machine Learning. A subset of AI that incorporates aspects of computer science, mathematics, and coding. Machine learning focuses on developing algorithms and models that can learn from data, and make predictions and decisions about new data.
Membership Inference attack	Data privacy attacks to determine if a data sample was part of the training set of a machine learning model.
NLP	Natural Language Processing. A subset of AI that enables computers to understand spoken and written human language. NLP enables features like text and speech recognition on devices.
Neural Network	A deep learning technique designed to resemble the human brain's structure. Neural networks require large data sets to perform calculations and create outputs, which enables features like speech and vision recognition.
Overfitting	Occurs in machine learning training when the algorithm can only work on specific examples within the training data. A typical functioning AI model should be able to generalise patterns in the data to tackle new tasks.
Prompt	A prompt is a natural language input that a user feeds to an AI system in order to get a result or output.
Reinforcement Learning	A type of machine learning in which an algorithm learns by interacting with its environment and then is either rewarded or penalised based on its actions.

SDLC	Software Development Life Cycle The process of integrating security considerations and practices into the various stages of software development. This integration is essential to ensure that software is secure from the design phase through deployment and maintenance.
Training data	Training data is the information or examples given to an AI system to enable it to learn, find patterns, and create new content.

ANNEX A

UNDERSTANDING AI THREATS

Adversarial threats are caused by threat actors with deliberate intention to cause harm. Typically, these threat actors are referred to as attackers or adversaries.

To understand these threats, system owners can refer to resources such as the OWASP Top 10 for Large Language Model Applications, or OWASP Machine Learning Security Top 10, or the MITRE ATLAS™ (Adversarial Threat Landscape for Artificial-Intelligence Systems). The MITRE ATLAS in particular provides **a structured knowledge base** for AI and cybersecurity professionals to understand and defend against AI cyber threats. It compiles adversary tactics, techniques, and case studies for AI systems based on real-world observations, demonstrations from ML red teams and security groups, as well as state-of-the-possible from academic research.

> Any attempt to secure an AI system should be on top of the 'traditional' good cybersecurity hygiene, such as implementing the principle of least privileges, multi-factor authentication, continuous security monitoring and auditing.

The ATLAS[3] Matrix (see **Table A1)** covers 2 types of adversarial 'techniques'.

- Techniques specific to AI/ML systems (indicated in orange boxes), and
- Techniques that are conventional cybersecurity offensive techniques, but applicable to both AI and non-AI systems and come directly from the MITRE Enterprise ATT&CK Matrix (indicated in white boxes).

System owners should continue to build their awareness of security threats using these resources, to better understand emerging risks that may have implications on their adoption of AI. As this space continues to evolve, such resources will aid both AI and cyber teams in their security risk assessment and management activities.

[3] MITRE ALTAS Framework: https://atlas.mitre.org/. It leverages the same core principles and structure of the well-known MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) framework, which is widely used by cyber defenders to map the terminologies of cybersecurity attacks. The ATLAS adapts these to the unique context of AI systems and potential adversarial attacks.

Table A1: MITRE ATLAS Matrix

Reconnaissance	Search for Victim's Publicly Available Research Materials	Search for Publicly Available Adversarial Vulnerability Analysis	Search Victim-Owned Websites	Search Application Repositories	Active Scanning		
Resource Development	Acquire Public ML Artifacts	Obtain Capabilities	Develop Capabilities	Acquire Infrastructure	Publish Poison Datasets	Poison Training Data	Establish Accounts
Initial Access	ML Supply Chain Compromise	Valid Accounts	Evade ML Model	Exploit Public-Facing Application	LLM Prompt Injection	Phishing	
ML Model Access	ML Model Inference API Access	ML-Enabled Product or Service	Physical Environment Access	Full ML Model Access			
Execution	User Execution	Command and Scripting Interpreter	LLM Plugin Compromise				
Persistence	Poison Training Data	Backdoor ML Model	LLM Prompt Injection				
Privilege Escalation	LLM Prompt Injection	LLM Plugin Compromise	LLM Jailbreak				
Defence Evasion	Evade ML Model	LLM Prompt Injection	LLM Jailbreak				
Credential Access	Unsecured Credentials						
Discovery	Discover ML Model Ontology	Discover ML Model Family	Discover ML Artifacts	LLM Meta Prompt Extraction			
Collection	ML Artifact Collection	Data from Information Repositories	Data from Local System				
ML Attack Staging	Create Proxy ML Model	Backdoor ML Model	Verify Attack	Craft Adversarial Data			
Exfiltration	Exfiltration via ML Inference API	Exfiltration via Cyber Means	LLM Meta Prompt Extraction	LLM Data Leakage			
Impact	Evade ML Model	Denial of ML Service	Spamming ML System with Chaff Data	Erode ML Model Integrity	Cost Harvesting	External Harms	

COMPANION GUIDE ON **SECURING AI SYSTEMS**

OCTOBER 2024

This document is meant as a community-driven resource with contribution from the AI and cybersecurity practitioner communities. It puts together available and practical mitigation measures and practices. This document is intended for informational purposes only and is not mandatory, prescriptive nor exhaustive.

System owners should refer to this Companion Guide as a resource, alongside other available resources in observing the Cyber Security Agency of Singapore's (CSA) Guidelines on Securing AI systems. This Companion Guide is a living document that will be continually updated to address material developments in this space.

DEVELOPED IN CONSULTATION WITH

This document is published by the CSA, developed with partners across the AI and Cyber communities, including:

Accenture
Artificial Intelligence Technical Committee, Information Technology Standards Committee (AITC, ITSC)
Association of Information Security Professionals (AiSP)'s Artificial Intelligence Special Interest Group (AI SIG)
Alibaba Cloud (Singapore) Pte Ltd
Amazon Web Services Singapore
Amaris.AI
BSA | The Software Alliance
Ensign InfoSecurity Pte Ltd
F5
Google Asia Pacific Pte Ltd
Huawei International Pte Ltd
Information Technology Industry Council (ITI)
Kaspersky Lab Singapore Pte Ltd
KPMG in Singapore
Microsoft Singapore
Pricewaterhouse Coopers Risk Services Pte Ltd
Rajah & Tann Cybersecurity Pte. Ltd.
Rajah & Tann Technologies Pte. Ltd.
Resaro.AI
SPH Media Limited
US-ASEAN Business Council
AI & Cyber practitioners across the Singapore Government

DISCLAIMER

These organisations provided views and suggestions on the security controls, descriptions of the security control(s), and technical implementations included in this Companion Guide. CSA and its partners shall not be liable for any inaccuracies, errors and/or omissions contained herein nor for any losses or damages of any kind (including any loss of profits, business, goodwill, or reputation, and/or any special, incidental, or consequential damages) in connection with any use of this Companion Guide. Organisations are advised to consider how to apply the controls within to their specific circumstances, in addition to other additional measures relevant to their needs.

VERSION HISTORY

VERSION	DATE RELEASED	REMARKS
0.1	29 July 2024	Draft release of Companion Guide
1.0	15 Oct 2024	First release

TABLE OF CONTENTS

1. INTRODUCTION 8
   1.1. PURPOSE AND SCOPE 9
2. USING THE COMPANION GUIDE 10
   2.1. START WITH A RISK ASSESSMENT 11
   2.2. IDENTIFY THE RELEVANT MEASURES/CONTROLS 12
      2.2.1. PLANNING AND DESIGN 13
      2.2.2. DEVELOPMENT 16
      2.2.3. DEPLOYMENT 31
      2.2.4. OPERATIONS AND MAINTENANCE 39
      2.2.5. END OF LIFE 42
3. USE CASE EXAMPLES 44
   3.1. DETAILED WALKTHROUGH EXAMPLE 44
      3.1.1. RISK ASSESSMENT EXAMPLE 45
      3.1.2. WALKTHROUGH OF TABULATED MEASURES/CONTROLS 46
   3.2. STREAMLINED IMPLEMENTATION EXAMPLE 56
      3.2.1. RISK ASSESSMENT EXAMPLE – EXTRACT ON PATCH ATTACK 57
      3.2.2. RELEVANT TREATMENT CONTROLS FROM COMPANION GUIDE 58

GLOSSARY 59

ANNEX A 63
   LIST OF AI TESTING TOOLS 66
      OFFENSIVE AI TESTING TOOLS 67
      DEFENSIVE AI TESTING TOOLS 70
      AI GOVERNANCE TESTING TOOLS 71

ANNEX B 74

REFERENCES 80

QUICK REFERENCE TABLE

Stakeholders in specific roles may use the following table to quickly reference relevant controls in section "2.2 IDENTIFY THE RELEVANT MEASURES/CONTROLS"

The roles defined below are included to guide understanding of this document and are not intended to be authoritative.

Decision Makers:

Responsible for overseeing the strategic and operational aspects of AI implementation for the AI system. They are responsible for setting the vision and goals for AI initiatives, defining product requirements, allocating resources, ensuring compliance, and evaluating risks and benefits.

Roles Included: Product Manager, Project Manager

AI Practitioners:

Responsible for the practical application (i.e. designing, developing, and implementing AI models and solutions) across the life cycle. This includes collecting or procuring and analysing data that goes into systems, building the AI system architecture and infrastructure, building and optimising the AI system to deliver the required functions, as well as conducting rigorous testing and validation of AI models to ensure their accuracy, reliability, and performance. In cases where the AI system utilizes a third-party AI system, AI Practitioners include the third-party provider responsible for these activities, e.g. as contracted through a Service Level Agreement (SLA). AI practitioners would be in charge of implementing the required controls across the entire system.

Roles Included: AI/ML Developer, AI/ML Engineer, Data Scientist

Cybersecurity Practitioners:

Responsible for ensuring the security and integrity of AI systems. This includes implementing security measures to protect AI systems in collaboration with AI Practitioners, monitoring for potential threats, ensuring compliance with cybersecurity regulations.

Roles Included: IT Security Practitioner, Cybersecurity Expert

The following sections may be relevant to **Decision Makers**:	The following sections may be relevant to **AI Practitioners**:	The following sections may be relevant to **Cybersecurity Practitioners**:
1.1 Team competency on threats and risks 1.2 Conduct security risk assessment	1.1 Team competency on threats and risks 1.2 Conduct security risk assessment	1.1 Team competency on threats and risks 1.2 Conduct security risk assessment
2.1 Secure the supply chain	2.1 Secure the supply chain 2.2 Model development 2.3 Identify, track and protect assets 2.4 Secure the AI development environment	2.1 Secure the supply chain 2.3 Identify, track and protect assets 2.4 Secure the AI development environment
3.1 Secure the deployment infrastructure and environment 3.2 Have well developed incident management procedures	3.1 Secure the deployment infrastructure and environment 3.2 Have well developed incident management procedures 3.3 Release AI responsibly	3.1 Secure the deployment infrastructure and environment 3.2 Have well developed incident management procedures 3.3 Release AI responsibly
4.4 Vulnerability disclosure process	4.1 Monitor system outputs and behaviour 4.2 Monitor system inputs 4.3 Have a secure-by-design approach to updates and continuous learning 4.4 Vulnerability disclosure process	4.1 Monitor system outputs and behaviour 4.2 Monitor system inputs 4.4 Vulnerability disclosure process
5.1 Proper data and model disposal	5.1 Proper data and model disposal	5.1 Proper data and model disposal

Table 1: User Quick Reference Table

1. INTRODUCTION

Artificial Intelligence (AI) poses benefits for economy, society, and national security. It has the potential to drive efficiency and innovation in almost every sector – from commerce and healthcare to transportation and cybersecurity.

To reap the benefits, users must have confidence that the AI will behave as designed, and outcomes are safe, secure, and responsible manner. However, in addition to safety risks, AI systems can be vulnerable to adversarial attacks, where malicious actors intentionally manipulate or deceive the AI system. **The adoption of AI can introduce or exacerbate existing cybersecurity risks to enterprise systems. These can lead to risks such as data leakage or data breaches, or result in harmful, unfair, or otherwise undesired model outcomes.** As such, the Cyber Security Agency of Singapore (CSA) has released the **Guidelines on Securing AI Systems** to advise **system owners on securing their adoption of AI.**

Nonetheless, **AI security is a developing field of study, and understanding of the security risks associated with AI continues to evolve internationally. As such, government agencies, our industry partners, AI and cybersecurity practitioners have put together this Companion Guide on Securing AI Systems**. The Companion Guide is a community-driven resource. It puts together available and practical mitigation measures and practices, drawing from industry and academia, as well as key resources such as the MITRE ATLAS database and OWASP Top 10 for Machine Learning and for Generative AI. System owners can refer to this Companion Guide as a resource, alongside other available resources in observing the Guidelines. This document is **intended for informational purposes only and is not mandatory, prescriptive nor exhaustive**. They should not be construed as comprehensive guidance or definitive recommendations.

> This Companion Guide is a living document that will be continually updated to address material developments in this space.

1.1. PURPOSE AND SCOPE

Purpose

This Companion Guide curates practical treatment measures and controls that system owners of AI systems may consider to secure their adoption of AI systems. **These measures/controls are voluntary, and not all the treatment measures/controls listed in this Companion Guide will be directly applicable** to all organisations or environments. Organisations may also be at different stages of development and release (e.g. POC, pilot, beta release). Organisations should consider relevance to their use cases/applications.

The Companion Guide is also meant as a resource to support system owners in addressing CSA's **Guidelines on Securing AI Systems**.

Scope

The controls within the Companion Guide primarily address the cybersecurity risks to AI systems. It does not address AI safety, or other common attendant considerations for AI such as fairness, transparency or inclusion, or cybersecurity risks introduced by AI systems, although some of the recommended controls may overlap. It also does not cover the misuse of AI in cyberattacks (AI-enabled malware), mis/disinformation, and scams (deepfakes).

2. USING THE COMPANION GUIDE

The Companion Guide puts together potential treatment measures/ controls that can support secure adoption of AI. However, not all of these controls might apply to your organisation.

Our goal is to put together a comprehensive set of treatment measures that system owners can consider for their respective use cases across the AI system lifecycle. These span the categories of People, Process and Technology.

There are two categories of measures/controls: (1) based on classical cybersecurity practices, which continue to be relevant to AI systems; and (2) others unique to AI systems. Measures/controls marked with an asterisk (*) next to their number indicates that they are unique to AI systems.

Each measure/control is **designed to be used independently**, to offer flexibility in customising which measures to evaluate and what mitigations to adopt, based on the specific needs of your organisation.

2.1. START WITH A RISK ASSESSMENT

As in CSA's Guidelines for Securing AI Systems, system owners should consider starting with a risk assessment. This will enable organisations to identify potential risks, priorities, and subsequently, the appropriate risk management strategies (including what measures and controls are appropriate).

You can consider the following **four steps** to tailor a systematic defence plan that best addresses your organisation's highest priority risks – protecting the things you care about the most.

STEP 1

Conduct risk assessment, focusing on security risks to AI systems

Conduct a risk assessment, focusing on security risks related to AI systems, either based on best practices or your organisation's existing Enterprise Risk Assessment/Management Framework.

Risk assessment can be done with reference to CSA published guides, if applicable:

- Guide To Cyber Threat Modelling
- Guide To Conducting Cybersecurity Risk Assessment for Critical Information Infrastructure

STEP 2

Prioritise areas to address based on risk/impact/resources

Prioritise which risks to address, based on risk level, impact, and available resources.

STEP 3

Identify and implement the relevant actions to secure the AI system

Identify relevant actions and control measures to secure the AI system, such as by referencing those outlined in the **Companion Guide on Securing AI Systems** and implement these across the AI life cycle.

STEP 4

Evaluate residual risks for mitigation or acceptance

Evaluate the residual risk after implementing security measures for the AI system to inform decisions about accepting or addressing residual risks.

2.2. IDENTIFY THE RELEVANT MEASURES/CONTROLS

Based on the risk assessment, system owners can identify the relevant measures/controls from the following tables. Each treatment measure/ control plays a different role, and should be assessed for relevance and priority in addressing the security risks specific to your AI system and context (Refer to section "2.1 START WITH A RISK ASSESSMENT").

Checkboxes are included to help users of this document to keep track of which measures/controls are applicable, and have (or have not) been implemented.

Related risks and Associated MITRE ATLAS Techniques[1] indicated serve as examples and are not exhaustive. They might differ based on your organisation's use case.

Example implementations are included for each measure/control as a more tangible elaboration on how they can be applied. These are also not exhaustive.

Additional **references and resources** are provided for users of this document to obtain further details on applying the treatment measure/control if required.

Asterisks (*) indicate measures/controls that are unique to AI systems (those without an asterisk indicate more classical cyber practices).

[1] MITRE ATLAS Framework offer a structured way to understand cyber threats in relation to AI systems (see Annex A)

2.2.1. PLANNING AND DESIGN

1.1	**Raise awareness and competency on security risks** Security is everyone's responsibility. Staff are provided with proper training and guidance.						
	Suggested Treatment Measures/Controls for consideration	**Yes**	**No**	**NA**	**Related Risks**	**Example Implementation**	**Reference or Resource**
1.1.1*	Ensure system owners and senior leaders understand threats to secure AI and their mitigations. **Responsible parties:** Decision Makers				• Incidents occurring due to poor cyber hygiene and/or knowledge	Attending seminars on AI threats, policies, and compliance and get exposed to case studies to appreciate the many AI potential and associated risks. Internal workshops and eLearning courses can inform employees on AI basics, responsible use, and relevant regulations. Integrate regular security training as part of the company's AI innovation training for a balanced approach. Online resources, e.g. electronic newsletters and YouTube videos could provide a means to track AI security developments that are emerging almost daily. Documentary evidence that team members have relevant security knowledge and training. These can include, where applicable: • Training records • Attendance records • Assessments • Certifications Establish the right cross-functional team to ensure that security, risk, and compliance considerations are included from the start.	• Principles for the Security of Machine Learning (UK NCSC) • Secure by Design - Shifting the Balance of Cybersecurity Risk: Principles and Approaches for Secure by Design Software • Failure modes in Machine Learning (Microsoft) • OWASP AI Exchange • Advisory Guidelines on use of Personal Data in AI Recommendation and Decision Systems
1.1.2*	Provide guidance to staff on Security by Design and Security by Default principles as well as unique AI security risks and failure modes as part of InfoSec training. e.g. LLM security matters, common AI weaknesses and attacks. **Responsible parties:** Decision Makers, AI Practitioners, Cybersecurity Practitioners						
1.1.3	Train developers in secure coding practices and good practices for the AI lifecycle. **Responsible parties:** Decision Makers, AI Practitioners				• Code vulnerabilities that could be exploited		

1.2	**Conduct security risk assessments** Apply a holistic process to model threats to the system.						
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Related Risks**	**Example Implementation**	**Reference or Resource**
1.2.1*	Understand AI governance and legal requirements, the impact to the system, users, organisation, if an AI component is compromised or has unexpected behaviour or there is an attack that affected AI privacy. Plan for an attack and its mitigation, using the principles of CIA. **Responsible parties:** Decision Makers, AI Practitioners, Cybersecurity Practitioners				• No triage, leading to confusion and locked or overloaded resources in the event of an AI security incident • Slow incident response, leading to large damage done • Slow remediation, leading to prolonged operational outage • Slow response means that attackers could do more damage, cover their tracks e.g. using anti-forensics	Perform a security risk assessment to determine the consequences and impact to the various stakeholders, and if the AI component does not behave as intended. Understand the AI inventory of systems used and their implications and interactions.	• Reference the case studies in this document. • Singapore Model Governance Framework for Generative AI • NIST AI Risk Management Framework • ISO 31000: Risk Management • MITRE ATLAS • NCSC Risk Management Guidance • OWASP Threat Modelling • OWASP Machine Learning Security Top Ten • Threats to AI using Microsoft STRIDE • Advisory Guidelines on use of Personal Data in AI Recommendation and Decision Systems • Model Artificial Intelligence Governance Framework
1.2.2*	Assess AI-related attacks and implement mitigating steps. **Responsible parties:** AI Practitioners, Cybersecurity Practitioners					Having/Developing a play book and AI incident handling procedures that will shorten the time to remediate and reduce resources wasted on unnecessary steps. Document the decision-making process of assessing potential AI threats and possible attack surfaces, as well as steps to mitigate these threats. This can be done through a threat risk assessment. Project risks may extend beyond security, e.g. newer AI models could obsolete	

1.2	**Conduct security risk assessments** Apply a holistic process to model threats to the system.						
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Related Risks**	**Example Implementation**	**Reference or Resource**
						the entire use case and business assumptions.	
1.2.3	Conduct a risk assessment in accordance with the relevant industry standards/best practices. **Responsible parties:** Decision Makers, AI Practitioners, Cybersecurity Practitioners				• Failure to comply with industry standards/best practices may lead to insufficient, inefficient or ineffective mitigations	Refer to the industry standards and best practices when performing risk assessment.	

2.2.2. DEVELOPMENT

2.1	**Secure the Supply Chain** Assess and monitor the security of the supply chain across the system's life cycle.						
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Related Risks**	**Example Implementation**	**Reference or Resource**
2.1.1	Implement Secure Coding and Development Lifecycle. **Responsible parties:** Decision Makers, AI Practitioners				• Introduction of bugs, vulnerabilities or unwanted and malicious active content, such as AI poisoning and model backdoors Associated MITRE ATLAS Techniques: • AML.T0018.000 Backdoor ML Model • AML.T0020.000 Poison Training Data • AML.T0010 ML Supply Chain Compromise	Adopt Security by Design. Apply software development lifecycle (SDLC) process. Use software development tools to check for insecure coding practices. Consider implementing zero trust principles in system design.	• CSA Critical Information Infrastructure Supply Chain Programme • NCSC Supply Chain Guidance • Supply-chain Levels for Software Artifacts (SLSA) • MITRE Supply Chain Security Framework • OWASP Top 10 LLM Applications • MITRE Supply Chain Security Framework • NIST Secure Software Development Framework for Generative AI and for Dual Use Foundation Models Virtual Workshop
2.1.2	Supply Chain Security: Ensure data, models, compilers, software libraries, developer tools and applications from trusted sources. **Responsible parties:** Decision Makers, AI Practitioners, Cybersecurity Practitioners					If procuring any AI System or component from a vendor, check/ensure suppliers adhere to policy and the equivalent security standards as your organisation. This could be done by establishing a Service Level Agreement (SLA) with the vendor. If the above is not plausible, consider using software components only from trusted sources. Verify object integrity e.g. hashes before using, opening, or running any files. Associated MITRE Mitigations: • AML.M0016 Vulnerability Scanning • AML.M0013 Code Signing • AML.M0007 Sanitize Training Data	

2.1	**Secure the Supply Chain** Assess and monitor the security of the supply chain across the system's life cycle.						
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Related Risks**	**Example Implementation**	**Reference or Resource**
						• AML.M0014 Verify ML Artifacts • AML.M0008 Validate ML Model	
2.1.3*	Protect the integrity of data that will be used for training the model. **Responsible parties:** AI Practitioners				• Data poisoning attacks • Exposure of sensitive and classified data in the AI training Data Associated MITRE ATLAS Techniques: • AML.T0020.000 Poison Training Data • AML.T0019 Publish Poison Dataset	Use automated data discovery tools to identify sensitive data across various environments, including databases, data lakes, and cloud storage. Implement secure workflow and data flow to ensure the integrity of the data used. When viable, have humans look at each data input and generate notifications where labels differ. Use statistical and automated methods to check for abnormalities. Associated MITRE Mitigations: • AML.M0007 Sanitize Training Data • AML.M0014 Verify ML Artifacts	• ETSI AI Data Supply Chain Security • DSTL Machine Learning with Limited Data
2.1.4*	Consider the trade-offs when deciding to use an untrusted 3rd party model (with or without fine tuning). **Responsible parties:** Decision Makers, AI Practitioners, Cybersecurity Practitioners				• Model backdoors • Remote code execution Associated MITRE ATLAS Techniques: • AML.T0018 Backdoor ML Model • AML.T0043 Craft Adversarial Data • AML.T0050 Command and Scripting Interpreter	Untrusted 3rd party models are models obtained from public/private repositories, whose publisher's origins cannot be verified. While there are benefits to relying on 3rd party models, possible risks	

2.1	**Secure the Supply Chain** Assess and monitor the security of the supply chain across the system's life cycle.						
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Related Risks**	**Example Implementation**	**Reference or Resource**
						include less control and visibility of model development. This reduced visibility may introduce backdoors injected by malicious actors. Consider the trade-offs based on your application's requirements Associated MITRE Mitigations: • AML.M0018 User Training • AML.M0013 Code Signing	
2.1.5*	Consider sandboxing untrusted models or serialised weight files where relevant. **Responsible parties:** AI Practitioners, Cybersecurity Practitioners					Running the model within a virtual machine or isolated environment away from production environment. Associated MITRE Mitigations: • AML.M0008 Validate ML Model • AML.M0018 User Training • AML.M0013 Code Signing	
2.1.6*	Scan models or serialised weight files. **Responsible parties:** AI Practitioners, Cybersecurity Practitioners					Use scanning tools such as Picklescan, Modelscan, on model files from an external source on a separate platform/system where the production system is on. Associated MITRE Mitigations: • AML.M0016 Vulnerability Scanning • AML.M0008 Validate ML Model	• Pickle Scanning (Hugging Face) • Stable Diffusion Pickle Scanner GUI • Also see Annex A – Technical Testing and System Validation

2.1	**Secure the Supply Chain** Assess and monitor the security of the supply chain across the system's life cycle.						
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Related Risks**	**Example Implementation**	**Reference or Resource**
2.1.7	Consider the trade-offs associated with using sensitive data for model training or inference. **Responsible parties:** Decision Makers, AI Practitioners				• Data leaks • Compromised privacy Associated MITRE ATLAS Techniques: • AML.T0024 Exfiltration via ML Inference API • AML.T0057 LLM Data Leakage • AML.T0056 LLM Meta Prompt Extraction • AML.T0040 ML Model Inference API Access • AML.T0047 ML Model Product or Service • AML.T0049 Exploit Public Facing Application	Check that data uploaded is non-sensitive or protected before submitting to the external model according to enterprise data protection policy/requirements. Organisations may explore various risk mitigation measures to secure their non-public sensitive data, such as anonymisation and privacy-enhancing technologies, before making decision on the use of sensitive data for model training. Pay specific attention to supplier policies on the confidentiality of user data, most notably ensure that suppliers commit that user inputs and model outputs are not subsequently used for model training. If necessary, consider techniques such as anonymisation, before deciding to use sensitive data for training. Associated MITRE Mitigations: • AML.M0012 Encrypt Sensitive Information • AML.M0016 Vulnerability Scanning	• Advisory Guidelines on use of Personal Data in AI Recommendation and Decision Systems

2.1	**Secure the Supply Chain** Assess and monitor the security of the supply chain across the system's life cycle.						
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Related Risks**	**Example Implementation**	**Reference or Resource**
2.1.8	Apply appropriate controls for data being sent out of the organisation. **Responsible parties:** AI Practitioners, Cybersecurity Practitioners				• Data leaks • Compromised privacy Associated MITRE ATLAS Techniques: • AML.T0024 Exfiltration via ML Inference API • AML.T0057 LLM Data Leakage • AML.T0056 LLM Meta Prompt Extraction • AML.T0040 ML Model Inference API Access • AML.T0047 ML Model Product or Service • AML.T0049 Exploit Public Facing Application • Insecure or vulnerable libraries, which can introduce unexpected attack surfaces • Model Subversion Associated MITRE ATLAS Techniques: • AML.T0016 Obtain Capabilities	Implement an automated Data Loss Prevention, exfiltration countermeasures, alert triggers and possibly human intervention e.g. added confirmation via login and input confirmation. Associated MITRE Mitigations: • AML.M0012 Encrypt Sensitive Information • AML.M0004 Restrict Number of ML Model Queries • AML.M0019 Control Access to ML Models and Data in Production.	
2.1.9	Consider evaluation of dependent software libraries, open-source models and when possible, run code checking. **Responsible parties:** AI Practitioners, Cybersecurity Practitioners					For example, ensure the library does not have arbitrary code execution when being imported or used. This can be done by using AI code checking, a vulnerability scanning tool, or checking against a database with vulnerability information. Associated MITRE Mitigations: • AML.M0008 Validate ML Model • AML.M0011 Restrict Library Loading • AML.M0004 Restrict Number of ML Model Queries • AML.M0008 Validate ML Model • AML.M0014 Verify ML Artifacts	• CVE List • Open-source Insights • OSS Insight

2.1	**Secure the Supply Chain** Assess and monitor the security of the supply chain across the system's life cycle.						
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Related Risks**	**Example Implementation**	**Reference or Resource**
						• AML.M0011 Restrict Library Loading	
2.1.10	Use software and libraries that does not have known vulnerabilities. **Responsible parties:** AI Practitioners, Cybersecurity Practitioners				• Insecure or vulnerable libraries, which can introduce unexpected attack surfaces • Model Subversion Associated MITRE ATLAS Techniques: • AML.T0016 Obtain Capabilities	Update to the latest secure patch in a timely manner. Associated MITRE Mitigations: • AML.M0008 Validate ML Model • AML.M0014 Verify ML Artifacts	• CVE List • Open-source Insights • OSS Insight

2.2	**Consider security benefits and trade-offs when selecting the appropriate model to use**						
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Related Risks**	**Example Implementation**	**Reference or Resource**
2.2.1*	Assess the need to use sensitive data for training the model, or directly referenced by the model. **Responsible parties:** AI Practitioners				• Privacy compromise • Attackers may be able to extract data used for training or from vector stores via malicious queries and prompt injections Associated MITRE ATLAS Techniques: • AML.T0057 LLM Data Leakage	Classify your organisation data based on sensitivity and/or enterprise data policy. Consider the need to use PII or sensitive data to generate vector databases that will be referenced by the model e.g. when using Retrieval Augmented Generation (RAG). Consider the trade-offs associated with using sensitive data for model training. Organisations may wish to explore various risk mitigation measures to secure their non-public sensitive data, such as anonymisation and privacy-enhancing technologies, before they decide whether to use such sensitive data for model training. Associated MITRE Mitigations: • AML.M0018 User Training	• Advisory Guidelines on use of Personal Data in AI Recommendation and Decision Systems (PDPC) • Generative AI Scoping Matrix • OWASP Machine Learning Security Top 10 (2023 edition) - Draft release v0.3 • OWASP Top 10 for Large Language Model Applications
2.2.2*	Consider Model hardening if appropriate. **Responsible parties:** AI Practitioners				• Input-based attacks • Prompt Injection • Adversarial Attacks • Model overfitting • Privacy compromise Associated MITRE ATLAS Techniques:	Apply data augmentation and adversarial training to reduce the effect of adversarial robustness attacks. Adversarial training: Inject adversarial text or image transformations (e.g. random flips, crops, rotation). This might	

2.2	**Consider security benefits and trade-offs when selecting the appropriate model to use**						
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Related Risks**	**Example Implementation**	**Reference or Resource**
					• AML.T0043 Craft Adversarial Data • AML.T0015 Evade ML Model • AML.T0024 Exfiltration via ML Inference API • AML.T0051 LLM Prompt Injection • AML.T0057 LLM Data Leakage • AML.T0054 LLM Jailbreak	impact the effectiveness of the model. For LLMs, prompt engineering best practices such as usage of guardrails and wrapping instructions in a single pair of salted sequence tags can be methods to further ground the model. Overfitting can increase the chance of adversarial attacks through model inversion. Associated MITRE Mitigations: • AML.M0003 Model Hardening • AML.M0006 Use Ensemble Methods • AML.M0010 Input Restoration • AML.M0015 Adversarial Input Detection • AML.M0004 Restrict Number of ML Model Queries	
2.2.3*	Consider implementing techniques to strengthen/harden the system apart from strengthening the model itself. **Responsible parties:** AI Practitioners				• Adversarial attacks on the model Associated MITRE ATLAS Techniques: • AML.T0015 Evade ML Model • Infrastructure Attacks • AML.T0029 Denial of ML Service • Attacker Recon activities • AML.TA002 ATLAS Tactic Recon	Supporting Countermeasures: • Cyber threat Intelligence to analyse and predict attacks. • Involve beta users (better red teaming) to test, exploit the wisdom of the crowds. • Anti-recon measures via hiding, disinformation, deception (honeypots).	

2.2	Consider security benefits and trade-offs when selecting the appropriate model to use						
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Related Risks**	**Example Implementation**	**Reference or Resource**
						• High quality datasets to improve model performance. • Data security controls for data collection, data storage, data processing, and data use as well as code and model security. • For LLMs, implement guardrails or input validation. • Implement endpoint security. • Consider implementing Zero Trust Principles for the system. Associated MITRE Mitigations: • AML.M0003 Model Hardening • AML.M0006 Use Ensemble Methods • AML.M0010 Input Restoration • AML.M0015 Adversarial Input Detection • AML.M0004 Restrict Number of ML Model Queries • AML.M0019 Control Access to ML Models and Data in Production	

2.3	**Identify, track and protect AI-related assets** Understand the value of AI-related assets, including models, data, prompts, logs, and assessments. Have processes to track, authenticate, version control, and secure assets.						
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Related Risks**	**Example Implementation**	**Reference or Resource**
2.3.1	Establishing a data lineage and software license management process. This includes documenting the data, codes, test cases and model, including any changes made and by whom. **Responsible parties:** AI Practitioners, Cybersecurity Practitioners				• Loss of data integrity • Unauthorised changes to data, model or system • Insider threats • Ransomware attacks • Loss of intellectual property Associated MITRE ATLAS Techniques: • AML.T0018.000 Backdoor ML Model • AML.T0020.000 Poison Training Data • AML.T0011 User Execution	Model cards, Data cards, and Software Bill of Materials (SBOMs) may be used. Associated MITRE Mitigations: • AML.M0016 Vulnerability Scanning • AML.M0013 Code Signing • AML.M0007 Sanitize Training Data • AML.M0014 Verify ML Artifacts • AML.M0008 Validate ML Model • AML.M0005 Control Access to ML Models and Data at Rest • AML.M0018 User Training	• Cybersecurity Code of Practice for Critical Information Infrastructure (CSA) • ISO 27001: Information security, cybersecurity and privacy protection
2.3.2	Secure data at rest, and data in transit. **Responsible parties:** AI Practitioners, Cybersecurity Practitioners				• Data loss and leaks. • Loss of data integrity. • Ransomware encryption. Associated MITRE ATLAS Techniques: • AML.T0024 Exfiltration via ML Inference API • AML.T0025 Exfiltration via Cyber Means • AML.T0054 LLM Jailbreak	Sensitive data (model weight and python code) is stored encrypted and transferred with proper encryption protocols, and secure key management. Consider saving model weights in secure formats such as safetensor, etc. Associated MITRE Mitigations: • AML.M0012 Encrypt Sensitive Information • AML.M0005 Control Access to ML Models and Data at Rest • AML.M0019 Control Access to ML Models and Data in Production	

2.3	**Identify, track and protect AI-related assets** Understand the value of AI-related assets, including models, data, prompts, logs, and assessments. Have processes to track, authenticate, version control, and secure assets.						
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Related Risks**	**Example Implementation**	**Reference or Resource**
2.3.3	Have regular backups in event of compromise. **Responsible parties:** AI Practitioners, Cybersecurity Practitioners					Identify essential data to backup more frequently. Implement a regular backup schedule. Have redundancy to ensure availability. Associated MITRE Mitigations: • AML.M0014 Verify ML Artifacts • AML.M0005 Control Access to ML Models and Data at Rest • AML.M0019 Control Access to ML Models and Data in Production	
2.3.4*	Implement controls to limit what AI can access and generate, based on sensitivity of the data. **Responsible parties:** AI Practitioners, Cybersecurity Practitioners				• Data leaks • Privacy attacks Associated MITRE ATLAS Techniques: • AML.T0036 Data from Information Repositories • AML.T0037 Data from Local System • AML.T0057 LLM Data Leakage	For sensitive data such as PII, explore various risk mitigation measures to secure non-public sensitive data, such as data anonymisation and privacy-enhancing techniques, before input into the AI. Have filters at the output to prevent sensitive information from being leaked. Associated MITRE Mitigations: • AML.M0012 Encrypt Sensitive Information • AML.M0019 Control Access to ML Models and Data in Production • AML.M0014 Verify ML Artifacts	• Advisory Guidelines on use of Personal Data in AI Recommendation and Decision Systems

2.3	**Identify, track and protect AI-related assets** Understand the value of AI-related assets, including models, data, prompts, logs, and assessments. Have processes to track, authenticate, version control, and secure assets.						
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Related Risks**	**Example Implementation**	**Reference or Resource**
						• AML.M0005 Control Access to ML Models and Data at Rest	
2.3.5	For very private data, consider if privacy enhancing technologies may be used. **Responsible parties:** AI Practitioners, Cybersecurity Practitioners				• Data leaks Associated MITRE ATLAS Techniques: • AML.T0024 Exfiltration via ML Inference API	Examples include having a Trusted Execution Environment, differential privacy or homomorphic encryption. Associated MITRE Mitigations: • AML.M0012 Encrypt Sensitive Information	

2.4	**Secure the AI development environment** Apply good infrastructure security principles.						
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Related Risks**	**Example Implementation**	**Reference or Resource**
2.4.1	Implement appropriate access controls to APIs, models and data, logs, and the environments that they are in. **Responsible parties:** AI Practitioners, Cybersecurity Practitioners				• Unauthorised access to system, data and models • Data breaches • Model/system compromise • Loss of intellectual property Associated MITRE ATLAS Techniques: • AML.T0024 Exfiltration via ML Inference API • AML.T0025 Exfiltration via Cyber Means • AML.T0036 Data from Information Repositories • AML.T0037 Data from Local System • AML.T0012 Valid Accounts • AML.T0057 LLM Data Leakage • AML.T0053 LLM Plugin Compromise • AML.T0054 LLM Jailbreak • AML.T0044 Full ML Model Access • AML.T0055 Unsecured Credentials • AML.T0013 Discover ML Ontology • AML.T0014 Discover ML Family • AML.T0007 Discover ML Artifacts • AML.T0035 ML Artifact Collection	Have secure authentication processes. Rule and role-based access controls to the development environment, based on the principles of least privilege. Have periodic reviews for role conflicts or violations of segregation of duties, and documentation should be retained including remediation actions. Access should be promptly revoked for terminated users or when the employee no longer requires access. Associated MITRE Mitigations: • AML.M0005 Control Access to ML Models and Data at Rest • AML.M0019 Control Access to ML Models and Data in Production • AML.M0012 Encrypt Sensitive Information • AML.M0014 Verify ML Artifacts	• Cybersecurity Code of Practice for Critical Information Infrastructure (CSA) • ISO 27001: Information security, cybersecurity and privacy protection • Advisory Guidelines on use of Personal Data in AI Recommendation and Decision Systems

2.4	**Secure the AI development environment** Apply good infrastructure security principles.						
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Related Risks**	**Example Implementation**	**Reference or Resource**
2.4.2	Implement access logging and monitoring. **Responsible parties:** AI Practitioners, Cybersecurity Practitioners				• Anomalies and suspicious activities not detectable • Failed compliance and audit. • Poor transparency and accountability • Insider threats Associated MITRE ATLAS Techniques: • AML.T0024 Exfiltration via ML Inference API • AML.T0025 Exfiltration via Cyber Means • AML.T0040 ML Model Inference API Access • AML.T0020.000 Poison Training Data	Log access with timestamps. Track changes to the data and model or configuration changes. Protect logs from being attacked (deleted, or tampered) Associated MITRE Mitigations: • AML.M0005 Control Access to ML Models and Data at Rest • AML.M0019 Control Access to ML Models and Data in Production	
2.4.3	Segregate production/ development environments. **Responsible parties:** AI Practitioners, Cybersecurity Practitioners				• Data integrity and confidentiality being compromised • Limit the impact of potential attacks • Risk of disruptions or conflicts between different functions/ models • Insider attacks Associated MITRE ATLAS Techniques: • AML.T0024 Exfiltration via ML Inference API • AML.T0025 Exfiltration via Cyber Means	Consider keeping different project environments separate from each other. E.g. development separated from production. If you are using cloud services, consider compartmentalizing your projects using VPCs, VMs, VPNs, enclaves, and containers Associated MITRE Mitigations: • AML.M0005 Control Access to ML Models and Data at Rest • AML.M0019 Control Access to ML Models and Data in Production	

2.4	**Secure the AI development environment** Apply good infrastructure security principles.						
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Related Risks**	**Example Implementation**	**Reference or Resource**
2.4.4	Ensure configurations are secure by default. **Responsible parties:** AI Practitioners, Cybersecurity Practitioners				• Unauthorized access and data breaches • Insider threats Associated MITRE ATLAS Techniques: • AML.T0024 Exfiltration via ML Inference API • AML.T0025 Exfiltration via Cyber Means	Default option should be secure against common threats. E.g. Implicitly deny access to sensitive data. Associated MITRE Mitigations: • AML.M0005 Control Access to ML Models and Data at Rest • AML.M0019 Control Access to ML Models and Data in Production	

2.2.3. DEPLOYMENT

3.1	**Secure the deployment infrastructure and environment of AI systems** Apply good infrastructure security principles.						
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Related Risks**	**Example Implementation**	**Reference or Resource**
3.1.1	Ensure contingency plans are in place to mitigate disruption or failure of AI services. **Responsible parties:** Decision Makers, AI Practitioners, Cybersecurity Practitioners				• Extended downtime to availability Associated MITRE ATLAS Techniques: • AML.T0029 Denial of ML Service	Having a manual or secondary system as a fail-over/fail-safe if the AI service becomes unavailable.	• Cybersecurity Code of Practice for Critical Information Infrastructure (CSA) • ISO 27001: Information security, cybersecurity and privacy protection • Advisory Guidelines on use of Personal Data in AI Recommendation and Decision Systems • NSA Guidance for Strengthening AI System Security
3.1.2	Implement appropriate access controls to APIs, models and data, logs, configuration files and the environments that they are in. **Responsible parties:** AI Practitioners, Cybersecurity Practitioners				• Unauthorized access to sensitive AI models and data • Data breaches • Loss of model integrity • Loss of intellectual property Associated MITRE ATLAS Techniques: • AML.T0024 Exfiltration via ML Inference API • AML.T0025 Exfiltration via Cyber Means • AML.T0040 ML Model Inference API Access • AML.T0020.000 Poison Training Data	Have secure authentication processes. Rule and role-based access controls to the deployment environment, based on the principles of least privilege. Have periodic reviews for role conflicts or violations of segregation of duties, and documentation should be retained including remediation actions. Access should be removed timely for terminated users or when the employee no longer requires access. Associated MITRE Mitigations: • AML.M0005 Control Access to ML Models and Data at Rest • AML.M0019 Control Access to ML Models and Data in Production	

3.1	**Secure the deployment infrastructure and environment of AI systems** Apply good infrastructure security principles.						
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Related Risks**	**Example Implementation**	**Reference or Resource**
3.1.3	Implement access logging, monitoring and policy management **Responsible parties:** AI Practitioners, Cybersecurity Practitioners				• Unauthorized access to deployment infrastructure and environment • Undetected Anomalies and suspicious activities • Nonadherence to compliance and audit requirements • Data integrity and accountability • Insider threats Associated MITRE ATLAS Techniques: • AML.T0024 Exfiltration via ML Inference API • AML.T0025 Exfiltration via Cyber Means • AML.T0040 ML Model Inference API Access	Keep a record of access to the model, inputs to the model, and output behaviour of the model. If necessary, track all AI applications, models and data. Have the ability to discover all AI apps, models, and data across the system, and who they are used by. Define and enforce data security policies across their environments. Associated MITRE Mitigations: • AML.M0005 Control Access to ML Models and Data at Rest • AML.M0019 Control Access to ML Models and Data in Production	
3.1.4	Implement segregation of environments. **Responsible parties:** AI Practitioners, Cybersecurity Practitioners				• Data integrity and confidentiality being compromised • Limit the impact of potential attacks, Risk of disruptions or conflicts between different functions/models Associated MITRE ATLAS Techniques: • AML.T0029 Denial of ML Service • AML.T0025 Exfiltration via Cyber Means	Keep different project environments separate from each other. E.g. when working on the cloud, have a separate VPC. Keep the development and operational environment apart. Associated MITRE Mitigations: • AML.M0019 Control Access to ML Models and Data in Production	

3.1	**Secure the deployment infrastructure and environment of AI systems** Apply good infrastructure security principles.						
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Related Risks**	**Example Implementation**	**Reference or Resource**
					• AML.T0031 Erode ML Model Integrity		
3.1.5	Ensure configurations are secure by default. **Responsible parties:** AI Practitioners, Cybersecurity Practitioners				• Vulnerability exploitation, Unauthorized access, Data breaches • Insider threats Associated MITRE ATLAS Techniques: • AML.T0024 Exfiltration via ML Inference API • AML.T0025 Exfiltration via Cyber Means • AML.T0031 Erode ML Model Integrity	Default option should be secure against common threats. E.g. Implicitly deny access to sensitive data. Associated MITRE Mitigations: • AML.M0019 Control Access to ML Models and Data in Production	
3.1.6	Consider implementing firewalls. **Responsible parties:** Cybersecurity Practitioners				• Unauthorized access to AI systems, models, and data • Network-based attacks, such as denial-of-service (DoS) attacks. • Malware and intrusion attempts • Unauthorized access to specific components of the AI systems Associated MITRE ATLAS Techniques: • AML.T0029 Denial of ML Service • AML.T0046 Spamming ML System with Chaff Data	Consider implementing Firewalls if the model is accessible to users online. Associated MITRE Mitigations: • AML.M0005 Control Access to ML Models and Data at Rest • AML.M0019 Control Access to ML Models and Data in Production	

3.1	**Secure the deployment infrastructure and environment of AI systems** Apply good infrastructure security principles.						
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Related Risks**	**Example Implementation**	**Reference or Resource**
3.1.7	Implement any other relevant security controls based on cybersecurity best practice, which has not been stated above. **Responsible parties:** Decision Makers, AI Practitioners, Cybersecurity Practitioners					Implement any other relevant security control based on best practice, such as ISO 27001.	

3.2	**Establish incident management procedures** Ensure proper incident response, escalation, and remediation plans.						
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Related Risks**	**Example Implementation**	**Reference or Resource**
3.2.1	Have plans to address different attack and outage scenarios. Implement measures to assist investigation. **Responsible parties:** Decision Makers, AI Practitioners, Cybersecurity Practitioners				• Failed Incident Response • Disruption to business continuity Associated MITRE ATLAS Techniques: • AML.T0029 Denial of ML Service	Have different incident response plans that address different types of outages and the potential attack scenarios, which may be blended with DOS. Implement forensics support and protect against erasure of evidence. Use cyber threat intelligence to support investigation. Associated MITRE Mitigations: • AML.M0018 User Training	• CSA Incident Response Checklist
3.2.2	Regularly reassess incident response plans as the system changes. **Responsible parties:** Decision Makers, AI Practitioners, Cybersecurity Practitioners				• Failed Incident Response • Disruption to business continuity Associated MITRE ATLAS Techniques: • AML.T0029 Denial of ML Service	Assess how changes to the system and AI will affect the attack surfaces. Associated MITRE Mitigations: • AML.M0018 User Training	
3.2.3	Have regular backups in event of compromise. **Responsible parties:** AI Practitioners, Cybersecurity Practitioners				• Data Loss • Ransomware attacks • Operational Disruptions • Data Integrity Associated MITRE ATLAS Techniques: • AML.T0029 Denial of ML Service • AML.T0031 Erode ML Model Integrity	Store critical data assets in offline backups. Associated MITRE Mitigations: • AML.M0014 Verify ML Artifacts	

3.2	**Establish incident management procedures** Ensure proper incident response, escalation, and remediation plans.						
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Related Risks**	**Example Implementation**	**Reference or Resource**
3.2.4	When an alert has been raised or investigation has confirmed an incident, report to the relevant stakeholders **Responsible parties:** Decision Makers, AI Practitioners, Cybersecurity Practitioners				• Regulatory non-Compliance • Increased cost and damages to the enterprise	Use threat hunting to determine full extent of attack and investigate attribution.	

3.3	**Release AI systems responsibly** Release models, applications, or systems only after subjecting them to appropriate and effective security checks and evaluation						
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Related Risks**	**Example Implementation**	**Reference or Resource**
3.3.1*	Verify models with hashes/signatures of model files and datasets before deployment or periodically, according to enterprise policy. **Responsible parties:** AI Practitioners				• Model Tampering/Poisoning • Data Poisoning • Backdoor/ Trojan model Associated MITRE ATLAS Techniques: • AML.T0018.000 Backdoor ML Model • AML.T0020.000 Poison Training Data	Compute and share model and dataset hashes/signatures when creating new models or data and update the relevant documentation e.g. model cards. Associated MITRE Mitigations: • AML.M0014 Verify ML Artifacts	
3.3.2*	Benchmark and test the AI models before release. **Responsible parties:** AI Practitioners				• Failure to achieve trust and reliability • Adversarial Attacks • Lack of accountability • Model Robustness Associated MITRE ATLAS Techniques: • AML.T0048 External Harm • AML.T0043 Craft Adversarial Data • AML.T0031 Erode ML Model Integrity	Models have been validated and achieved performance targets before deployment. Consider using an adversarial test set to validate model robustness, where possible. Conduct AI Red-Teaming. Associated MITRE Mitigations: • AML.M0008 Validate ML Model • AML.M0014 Verify ML Artifacts	• Adversarial Robustness Toolbox (IBM) • CleverHans (University of Toronto) • TextAttack (University of Virginia) • Prompt Bench (Microsoft) • Counterfit (Microsoft) • AI Verify (Infocomm Media Development Authority, Singapore) • Moonshot (Infocomm Media Development Authority, Singapore)

3.3	**Release AI systems responsibly** Release models, applications, or systems only after subjecting them to appropriate and effective security checks and evaluation						
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Related Risks**	**Example Implementation**	**Reference or Resource**
3.3.3	Consider the need to conduct security testing on the AI systems. **Responsible parties:** AI Practitioners, Cybersecurity Practitioners				• Security Vulnerabilities Associated MITRE ATLAS Techniques: • AML.T0048 External Harm • AML.T0031 Erode ML Model Integrity	Perform VAPT/security testing on AI systems. Prioritise and focus on the most realistic and practical attacks, based on the risk assessment during the planning phase. System owner and project teams to follow up on findings from security testing/red team, by assessing the criticality of vulnerabilities uncovered, apply additional measures and if necessary, seek approval from relevant entity e.g. CISO, for acceptance of residual risks, according to their enterprise risk management/cybersecurity policies. Create a feedback loop to maximise the impact of the findings from security tests. Associated MITRE Mitigations: • AML.M0003 Model Hardening • AML.M0006 Use Ensemble Methods • AML.M0016 Vulnerability Scanning	• OWASP Top 10 for Large Language Model Applications • Web LLM attacks (Portswigger)

2.2.4. OPERATIONS AND MAINTENANCE

4.1	**Monitor AI system inputs** Monitor and log inputs to the system, such as queries, prompts and requests. Proper logging allows for compliance, audit, investigation and remediation.						
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Related Risks**	**Example Implementation**	**Reference or Resource**
4.1.1*	Validate/Monitor inputs to the model and system for possible attacks and suspicious activity. **Responsible parties:** AI Practitioners, Cybersecurity Practitioners				• Adversarial Attacks • Data exfiltration Associated MITRE ATLAS Techniques: • AML.T0043 Craft Adversarial Data • AML.T0025 Exfiltration via Cyber Means	AI System owners may consider to monitor and validate input prompts, queries or API requests for attempts to access, modify or exfiltrate information deemed confidential by the organisation. Consider use of classifiers to detect malicious inputs and log them for future review to identify potential vulnerabilities. Note: Implementor should consider the current privacy regulations/guidelines when logging inputs. Associated MITRE Mitigations: AML.M0015 Adversarial Input Detection	• Introduction to Logging for Security Purpose (NCSC) • OpenAI usage policies • Advisory Guidelines On use of Personal Data In AI Recommendation and Decision Systems (PDPC)
4.1.2	Monitor/Limit the rate of queries. **Responsible parties:** AI Practitioners, Cybersecurity Practitioners				• Denial of Service (DoS) Attacks Associated MITRE ATLAS Techniques: • AML.T0029 Denial of ML Service • AML.T0034 Cost Harvesting	If possible, prevent users from continuously querying the model with a high frequency e.g. API throttling. This mitigates the potential for membership-inference and extraction attacks. Associated MITRE Mitigations: • AML.M0004 Restrict Number of ML Model Queries	

4.2	**Monitor AI system outputs and behaviour** Monitor for anomalous behaviour that might indicate intrusions or compromise.						
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Related Risks**	**Example Implementation**	**Reference or Resource**
4.2.1*	Monitor model outputs and model performance. **Responsible parties:** AI Practitioners, Cybersecurity Practitioners				• Adversarial Attacks • Operational Impact Associated MITRE ATLAS Techniques: • AML.T0031 Erode ML Model Integrity • AML.T0020.000 Poison Training Data • AML.T0029 Denial of ML Service • AML.T0048 External Harms	Implement an alert system that monitors for anomalous or unwanted output. E.g. a customer facing chatbot that is safe for work begins to output profanity instead. Associated MITRE Mitigations: • AML.M0008 Validate ML Model	
4.2.2*	Ensure adequate human oversight to verify model output, when viable or appropriate. **Responsible parties:** AI Practitioners, Cybersecurity Practitioners				• False Positives from the model • Misinterpretation of Context • Adverse Impact on Operations Associated MITRE ATLAS Techniques: • AML.T0029 Denial of ML Service • AML.T0048 External	Manual investigation of unusual or anomalous alert notifications. For critical systems, ensure human oversight to verify decisions recommended by the model. Associated MITRE Mitigations: • AML.M0018 User Training • AML.M0015 Adversarial Input Detection	

4.3	**Adopt a secure-by-design approach to updates and continuous learning.** Ensure risks associated to model updates have been considered. Changes to the data and model can lead to changes in behaviour.						
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Related Risks**	**Example Implementation**	**Reference or Resource**
4.3.1*	Treat major updates as new versions and integrate software updates with model updates and renewal. **Responsible parties:** AI Practitioners				• Model Tampering/Poisoning • Backdoor/ Trojan model Associated MITRE ATLAS Techniques: • AML.T0020.000 Poison Training Data • AML.T0018.000 Backdoor ML Model • AML.T0031 Erode ML Model Integrity • AML.T0010 ML Supply Chain Compromise	New models to be validated, benchmarked, and be tested before release. Associated MITRE Mitigations: • AML.M0008 Validate ML Model • AML.M0014 Verify ML Artifacts	• Principles for the Security of Machine Learning (UK NCSC)
4.3.2*	Treat new input data used for training as new data. **Responsible parties:** AI Practitioners				• Data Poisoning • Poison/Backdoor/Trojan model Associated MITRE ATLAS Techniques: • AML.T0020.000 Poison Training Data • AML.T0018.000 Backdoor ML Model • AML.T0010 ML Supply Chain Compromise	Subject new input to the same verification and validation as new data. Associated MITRE Mitigations: • AML.M0007 Sanitize Training Data	

4.4	**Establish a vulnerability disclosure process** Have a feedback process for users to share any findings of concern, which might uncover potential vulnerabilities to the system.						
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Possible Risk Mitigated**	**Example Implementation**	**Reference or Resource**
4.4.1	Maintain open lines of communication. **Responsible parties:** Decision Makers, AI Practitioners, Cybersecurity Practitioners				• Regulatory non-Compliance	Set up channels to allow users to provide feedback on security and usage.	• SingCERT Vulnerability Disclosure Policy (CSA) • UK NCSC Vulnerability Disclosure Toolkit • CVE List • AI CWE List • ATLAS Case Studies
4.4.2	Share findings with appropriate stakeholders. **Responsible parties:** Decision Makers, AI Practitioners, Cybersecurity Practitioners				• Regulatory non-Compliance	Share discoveries of vulnerabilities to relevant stakeholders such as the company CISO.	

2.2.5. END OF LIFE

5.1	**Ensure proper data and model disposal**						
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Related Risks**	**Example Implementation**	**Reference or Resource**
5.1.1	Ensure proper and secure disposal/destruction of data and models in accordance with data privacy standards and/or relevant rules and regulations. **Responsible parties:** Decision Makers, AI Practitioners, Cybersecurity Practitioners				• Regulatory non-Compliance • Sensitive data loss	Examples include crypto shredding or degaussing	• Personal Data Protection Act (PDPA) • Advisory Guidelines on use of Personal Data in AI Recommendation and Decision Systems

3. USE CASE EXAMPLES

3.1. DETAILED WALKTHROUGH EXAMPLE

Case Study: Implementing Companion Guide on LLM-based Chatbot

- Company A is currently testing out an LLM to implement as their customer service chatbot, known as *SuperResponder*.
- The model is an LLM that is downloaded from an open-source model hosting website (Hugging Face) and further developed in-house on a cloud environment.
- The data is sourced from manually curated FAQs from customer service conversations, which will be converted to a vector database to implement Retrieval Augmented Generation (RAG) with the downloaded LLM model.

Supply Chain Attacks

In this example, Company A relies heavily on third party AI software components to develop *SuperResponder*.

The integrity and security of AI supply chains are essential for ensuring the reliability and trustworthiness of AI systems. AI vulnerabilities in the supply chain refer to weaknesses or exploitable points within the processes of acquiring, integrating, and deploying AI technologies. These vulnerabilities can stem from malicious or compromised components, including datasets, models, algorithms, and software libraries, which may introduce security risks and threats to AI systems[2].

[2] https://vulcan.io/blog/understanding-the-hugging-face-backdoor-threat/

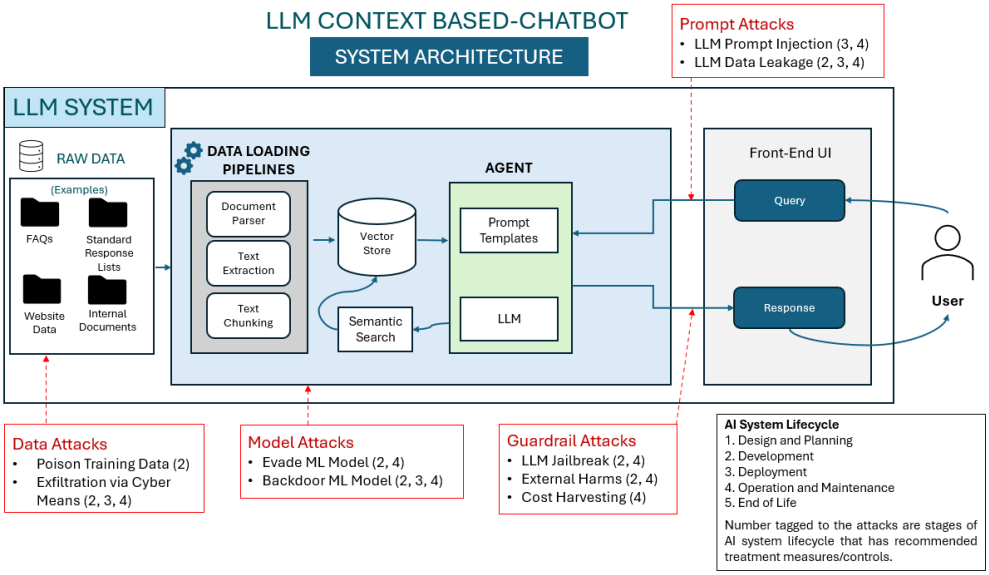


Figure 2. LLM Context Based-chatbot System Architecture

3.1.1. RISK ASSESSMENT EXAMPLE

Company A performed a risk assessment to identify and address potential risks to confidentiality, integrity, availability of their AI system. If the risks are not mitigated, there is a potential for an attacker to exploit the list of vulnerabilities, causing *SuperResponder* to be compromised. This could result in widespread customer dissatisfaction and damage to the company's reputation.

The hypothetical risk assessment* is as follows:

Risk Scenarios	Impact	Likelihood	Proposed Mitigations	Risk Level
Prompt injection attack Crafted input can be executed to instruct LLM to retrieve private customer information.	**Confidentiality: High** Confidential information such as PII data of customers may be leaked.	**Likelihood: Medium** Chatbot interface is public facing. Attack can be performed easily without privileged access and be repeated continuously.	Use automated tools to remove PIIs from datasets used. In addition, use data protection measures and output sanitisation mechanisms.	Initial Risk Level: **Medium** Residual Risk Level: **Low**
Supply Chain Vulnerabilities. Use of compromised pre-trained LLM can introduce other vulnerabilities such as model backdoor.	**Integrity: High** The chatbot may be prompted to regularly output the wrong answer or advice to customers.	**Likelihood: Medium** It is possible to upload compromised models onto public model hosting platforms. These models are downloaded and used to develop the chatbot.	Scanning the model. Sandboxing the model. Download models from trusted model developers or sources.	Initial Risk Level: **Medium** Residual Risk Level: **Low**
Model Denial of Service. Chatbot at risk of volumetric and continuous querying, consuming a large amount of resource.	**Availability: Medium** The chatbot service can be overwhelmed by a large volume of requests and become unavailable to other users.	**Likelihood: Medium** Volumetric and continuous querying of the chatbot can be performed with some scripting knowledge or automated tools.	API throttling.	Initial Risk Level: **Medium** Residual Risk Level: **Low**

* The above table is not exhaustive and is meant as an example of a risk assessment done.

3.1.2. WALKTHROUGH OF TABULATED MEASURES/CONTROLS

Following the risk assessment, Company A promptly referenced the CSA Guidelines for Securing AI Systems and the Companion Guide to mitigate the risks. The list of implemented actions are as follows:

3.1.2.1. PLANNING AND DESIGN STAGE

1.1	**Raise awareness and competency on security risks**				
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Implementation done / reason not done**
1.1.1	Ensure system owners and senior leaders understand threats to secure AI and their mitigations.	✓			System owners have attended seminars on AI security and understood potential risks associated with AI systems.
1.1.2	Provide guidance to staff on Security by Design and Security by Default principles as well as unique AI security risks and failure modes as part of InfoSec training. e.g. LLM security matters, common AI weaknesses and attacks.	✓			Trained staff on AI security and risks, e.g. attack vectors, and countermeasures (practical defence strategies). Developers were sent to attend a 3-day course on AI & Cybersecurity covering adversarial machine learning at a local tertiary institution. They also referred to online courses from Udemy on AI security essentials and AI risk management.
1.1.3	Train developers are trained in secure coding practices and good practices for the AI lifecycle.	✓			Developers have attended certified workshops on how to maintain secure coding practices when developing the model.

1.2	Conduct security risk assessments				
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Implementation done / reason not done**
1.2.1	Understand AI governance and legal requirements, the impact to the system, users, organisation, if an AI component is compromised or has unexpected behaviour or there is an attack that affected AI privacy. Plan for an attack and its mitigation, using the principles of CIA.	✓			Understood AI Verify framework, PDPA Guidance for AI and User Data, Model Governance Framework for Generative AI (IMDA)
1.2.2	Assess AI-related attacks and implement mitigating steps.	✓			Threat Modelling: Identify and assess potential attack vectors from adversarial attacks such as prompt injection, membership inference, data poisoning and backdoor attacks using mitre atlas framework
1.2.3	Conduct risk assessment is done in accordance with the relevant industry standards/best practices.	✓			Risk assessment was conducted in accordance with company risk management policy

3.1.2.2. DEVELOPMENT

2.1	Secure the Supply Chain				
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Example Implementation of Action**
2.1.1	Implement Secure Coding and Development Lifecycle.	✓			Attended secure coding courses and adopt secure coding practices for development of the LLM.
2.1.2	Supply Chain Security: Ensure data, models, compilers, software libraries, developer tools and applications from trusted sources.	✓			For the pre-trained 3rd party LLM model – Applied Source verification to ensure data and models obtained are from trusted and reputable sources. Verified the authenticity and integrity of the sources before incorporating them into the system (digital signatures).
2.1.3	Protect the integrity of data that will be used for training the model.	✓			Data to support Retrieval Augmented Generation (RAG) is sourced from company's own customer service conversations and internal FAQ documents.
2.1.4	Consider the trade-offs when deciding to use an untrusted 3rd party model (with or without fine tuning).	✓			Examples of risks considered: Data Breaches, Data Privacy Leakage, Service Disruptions, Model backdoor. Compensatory measures such as prompt filters and prompt engineering to mitigate adversarial attacks.
2.1.5	Consider sandboxing untrusted models or serialised weight files where relevant.	✓			Implemented virtual machines (VMs), to isolate and restrict execution environment of these components.
2.1.6	Scan models or serialised weight files.	✓			Scanned model files with Picklescan
2.1.7	Consider the trade-offs associated with using sensitive data for model training or inference			✓	Not using external APIs during development
2.1.8	Apply appropriate controls for data being sent out of the organisation.			✓	Not required as model is hosted locally and not a SaaS.

2.1.9	Consider evaluation of dependent software libraries, open-source models and when possible, run code checking.	✓			Used a vulnerability scanner to ensure safety of third-party libraries from known CVEs
2.1.10	Use software and libraries that does not have known vulnerabilities.	✓			Use of updated software and libraries with no known vulnerability in accordance with company IT policy

2.2	**Consider security benefits and trade-offs when selecting the appropriate model to use**				
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Implementation done / reason not done**
2.2.1	Assess the need to use sensitive data for training the model, or directly referenced by the model.	✓			Sensitive data not used for vector database. Training data has been carefully sanitised by sensitive data redaction methods to counter inference attacks.
2.2.2	Consider Model hardening if appropriate.	✓			Prompt engineering to prevent the model from producing output beyond what is intended. Implemented guardrails to ensure sensitive data is not disclosed.
2.2.3	Consider implementing techniques to strengthen/harden the system apart from strengthening the model itself.	✓			Added input prompt filters and output filters for unwanted topics, to mitigate against prompt injections.

2.3	Identify, track and protect AI-related assets				
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Implementation done / reason not done**
2.3.1	Establishing a data lineage and software license management process. This includes documenting the data, codes, test cases and model, including any changes made and by whom.	✓			Maintained documentation of the changes made to newer model versions on Model cards and verified it.
2.3.2	Secure data at rest, and data in transit.	✓			Encryption algorithms approved by enterprise security policy is used for data at rest and transit.
2.3.3	Have regular backups in event of compromise.	✓			Used git to maintain version control of the codebase and model artifacts.
2.3.4	Implement controls to limit what AI can access and generate, based on sensitivity of the data.	✓			Prompt engineering to ensure that the model is less likely to generate any unwanted topics.
2.3.5	For very private data, privacy enhancing technologies may be used.			✓	No private data used

2.4	Secure the AI development environment				
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Implementation done / reason not done**
2.4.1	Appropriate access controls to APIs, models and data, logs, and the environments that they are in.	✓			Rule and role-based access controls implemented for developers
2.4.2	Implement access logging and monitoring.	✓			Turned on cloud native logging.
2.4.3	Segregation production/ development environments.	✓			Developer environment is in a different VPC from the deployment environment.
2.4.4	Ensure configurations are secure by default.	✓			Implicit deny access to unauthorised users via cloud native identity and access management.

3.1.2.3. DEPLOYMENT

3.1	Secure the deployment infrastructure and environment of AI systems				
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Implementation done**
3.1.1	Ensure contingency plans are in place to mitigate disruption or failure of AI services.	✓			Deployed a backup availability zone to ensure availability of service.
3.1.2	Implement appropriate access controls to APIs, models and data, logs, configuration files and the environments that they are in.	✓			General users only have access to the LLM interface via the frontend chatbot, no access to the backend environment.
3.1.3	Implement access logging, monitoring and policy management	✓			Turned on cloud native logging.
3.1.4	Implementation segregation of environments.	✓			Deployment environment is in a different VPC from the development environment.
3.1.5	Ensure configurations are secure by default.	✓			Implicit deny access to unauthorised users via cloud native identity and access management.
3.1.6	Consider implementing firewalls.	✓			Configured firewalls in between access to environment and model
3.1.7	Implement any other relevant security controls based on cybersecurity best practice, which has not been stated above.			✓	Current controls are in line with company cybersecurity policy.

3.2	Establish incident management procedures				
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Implementation done / reason not done**
3.2.1	Have plans to depict different attack and outage scenarios. Implement measures to assist investigation.	✓			Conducted exercise to simulate outage of AI chatbot and fail over to another availability zone.
3.2.2	Regularly reassess incident response plans as the system changes.	✓			Will reassess system every 12 months or whenever there is an update to the system, according to company cybersecurity policy.
3.2.3	Have regular backups in event of compromise.	✓			Weekly backups in place, according to company IT policy.
3.2.4	When an alert has been raised or investigation has confirmed incident, to report to the relevant stakeholders.	✓			Procedure in place to report to CISO, in accordance with incident response standard operating procedure.

3.3	Release AI systems responsibly				
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Implementation done / reason not done**
3.3.1	Verify models with hashes/signatures of model files and datasets before deployment or periodically, according to enterprise policy.	✓			Models are validated with hashes before deployment.
3.3.2	Benchmark and test the AI models before release.	✓			Prepared a golden dataset to validate and benchmark model. Conducted red teaming on the LLM model before release; incorporating test cases on prompt injection and supply chain attacks, which were identified during the security risk assessment.
3.3.3	Consider need to conduct security testing on the AI systems.	✓			Performed VAPT/security testing on LLM systems. The system owner followed up on findings from the red team, assessed the criticality of vulnerabilities uncovered, applied additional measures, and sought approval from CISO for acceptance of vulnerabilities that cannot be rectified.

3.1.2.4. OPERATIONS AND MAINTENANCE

4.1	Monitor AI system inputs				
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Implementation done / reason not done**
4.1.1	Validate/monitor inputs to the model and system for possible attacks and suspicious activity.	✓			All inputs to the LLM that has guardrails triggered are logged for future review and to identify potential vulnerabilities in prompt design.
4.1.2	Monitor/Limit the rate of queries.	✓			API throttling is in place to limit rate on queries to model.

4.2	Monitor AI system outputs and behaviour				
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Implementation done / reason not done**
4.2.1	Monitoring of model outputs and model performance.	✓			Implement a monitoring system to detect anomalous behaviour or outputs from the LLM system that could indicate an attack or vulnerability.
4.2.2	Ensure adequate human oversight to verify model output, when viable or appropriate.	✓			Manually investigate unusual, automated processes that are flagged as anomalous.

4.3	Adopt a secure-by-design approach to updates and continuous learning.				
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Implementation done / reason not done**
4.3.1	Treat major updates as new versions and integrate software updates with model updates and renewal.	✓			To validate and benchmark new models and updates against a 'Golden dataset'
4.3.2	Treat new input data used for training as new data.	✓			New data used for finetuning will be validated as they were new data.

4.4	Establish a vulnerability disclosure process				
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Implementation done / reason not done**
4.4.1	Maintain open lines of communication.	✓			Establish a vulnerability disclosure program (bounty program, etc.) to encourage responsible reporting and handling of security vulnerabilities by the users.
4.4.2	Share findings with appropriate stakeholders.	✓			New findings will be shared with company CISO

3.1.2.5. END OF LIFE

5.1	Ensure proper data and model disposal				
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Implementation done / reason not done**
5.1.1	Ensure proper and secure disposal/destruction of data and models in accordance with data privacy standards and/or relevant rules and regulations.	✓			All data related to the chatbot, and vector database will be deleted through the CSP data disposal process, in line with company data policy.

3.2. STREAMLINED IMPLEMENTATION EXAMPLE

Case Study: Patch attacks on image recognition surveillance system

- Company B has recently implemented an advanced AI-driven facial recognition gantry system at all access points at their office.
- The system is part of enhanced security measures to identify individuals and to streamline employee flow by reducing dependence on manual checks.
- Facial recognition systems utilise deep learning algorithms to identify individuals, by analysing visual data captured through cameras.

Patch Attacks

In this example, the system owner has identified patch attack as a possible attack vector for this system

A patch attack is a type of attack that disrupts object classification in a camera's visual field by introducing a specific pattern or object. This disruption can lead to misinterpretation or evasion attacks.

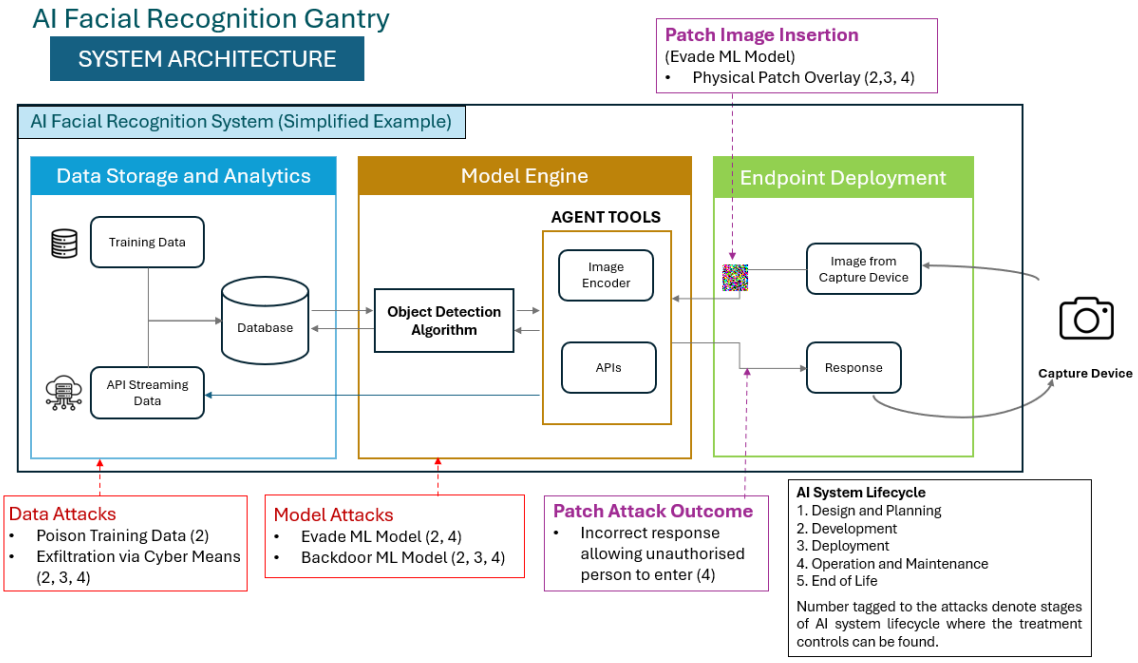


Figure 3. AI Facial Recognition Gantry System Architecture

3.2.1. RISK ASSESSMENT EXAMPLE – EXTRACT ON PATCH ATTACK

The following is an extract from a security risk assessment, specific to an image patch attack.

Risk Scenarios	Impact	Likelihood	Proposed Mitigations	Risk Level
Image Patch Evasion attack. Attacker can use adversarial patches to compromise physical security measures, leading to unauthorised access and potential security breaches.	**Integrity: High** Integrity of the AI facial recognition system will be impacted allowing unauthorised personnel to access the gantry	**Likelihood: Low** Threat actors need to know how the facial recognition AI model works in order to generate a malicious patch that is effective	• Adversarial training • Ensemble model • Multiple sensors • Input Filtering	**Initial Risk Level: High** **Residual Risk Level: Low**

3.2.2. RELEVANT TREATMENT CONTROLS FROM COMPANION GUIDE

To avoid repetition from section 5.1, we outline only the essential controls related to the Patch Attack scenario.

2.2	**Consider security benefits and trade-offs when selecting the appropriate model to use**				
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Implementation done / reason not done**
2.2.2	Consider Model hardening if appropriate.	✓			**Adversarial training** is implemented. **Ensemble Model**: Utilised ensemble approaches that combine multiple facial recognition algorithms. These measures can enhance robustness and resilience against image patch attacks, mitigating the impact of individual vulnerabilities
2.2.3	Consider implementing techniques to strengthen/harden the system apart from strengthening the model itself.	✓			**Multi-Sensor Fusion:** Multiple cameras and lasers used to detect the face.

4.1	**Monitor AI system inputs**				
	Treatment Measures/Controls	**Yes**	**No**	**NA**	**Implementation done / reason not done**
4.1.1	Validate/monitor inputs to the model and system for possible attacks and suspicious activity.	✓			**Additional input filtering layer to detect if abnormal patches are present.** **Having a staff to verify when one is detected.**

GLOSSARY

Term	Brief description
AI system	Artificial Intelligence. A machine-based system that for explicit or implicit objectives, infers, from the input it receives, how to generate outputs such as predictions, content, recommendations, or decisions that can influence physical or virtual environments. Different AI systems vary in their levels of autonomy and adaptiveness after deployment.
Adversarial Machine Learning	The process of extracting information about the behaviour and characteristics of an ML system and/or learning how to manipulate the inputs into an ML system in order to obtain a preferred outcome.
Anomaly Detection	The identification of observations, events or data points that deviate from what is usual, standard, or expected, making them inconsistent with the rest of data.
API	Application Programming Interface. A set of protocols that determine how two software applications will interact with each other.
Backdoor attack	A backdoor attack is when an attacker subtly alters AI models during training, causing unintended behaviour under certain triggers.
Chatbot	A software application that is designed to imitate human conversation through text or voice commands
Computer Vision	An interdisciplinary field of science and technology that focuses on how computers can gain understanding from images and videos.
Data Breach	Data Breach occurs when a threat actor gains unauthorised access to sensitive/confidential data.
Data Integrity	The property that data has not been altered in an unauthorised manner. Data integrity covers data in storage, during processing, and while in transit.

Data Leakage	Unintentional exposure of sensitive, protected, or confidential information outside its intended environment.
Data Loss Prevention	A system's ability to identify, monitor, and protect data in use (e.g., endpoint actions), data in motion (e.g., network actions), and data at rest (e.g., data storage) through deep packet content inspection, and contextual security analysis of transaction (e.g., attributes of originator, data object, medium, timing, recipient/destination, etc.) within a centralised management framework.
Data Poisoning	Control a model with training data modifications.
Data Science	An interdisciplinary field of technology that uses algorithms and processes to gather and analyse large amounts of data to uncover patterns and insights that inform business decisions.
Deep Learning	A function of AI that imitates the human brain by learning from how it structures and processes information to make decisions. Instead of relying on an algorithm that can only perform one specific task, this subset of machine learning can learn from unstructured data without supervision.
Defence-in-Depth	Defence in depth is a strategy that leverages multiple security measures to protect an organization's assets. The thinking is that if one line of defence is compromised, additional layers exist as a backup to ensure that threats are stopped along the way.
Evasion attack	Crafting input to AI in order to mislead it into performing its task incorrectly.
Extraction attack	Copy or steal an AI model by appropriately sampling the input space and observing outputs to build a surrogate model that behaves similarly.
Generative AI	A type of machine learning that focuses on creating new data, including text, video, code and images. A generative AI system is trained using large amounts of data, so that it can find patterns for generating new content.
Guardrails	Restrictions and rules placed on AI systems to make sure that they handle data appropriately and don't generate unethical content.

Hallucination	An incorrect response from an AI system, or false information in an output that is presented as factual information.
Image Recognition	Image recognition is the process of identifying an object, person, place, or text in an image or video.
LLM	Large Language Model. A type of AI model that processes and generates human-like text. LLMs are specifically trained on large data sets of natural language to generate human-like output.
ML	Machine Learning. A subset of AI that incorporates aspects of computer science, mathematics, and coding. Machine learning focuses on developing algorithms and models that can learn from data, and make predictions and decisions about new data.
Membership Inference attack	Data privacy attacks to determine if a data sample was part of the training set of a machine learning model.
NLP	Natural Language Processing. A subset of AI that enables computers to understand spoken and written human language. NLP enables features like text and speech recognition on devices.
Neural Network	A deep learning technique designed to resemble the human brain's structure. Neural networks require large data sets to perform calculations and create outputs, which enables features like speech and vision recognition.
Overfitting	Occurs in machine learning training when the algorithm can only work on specific examples within the training data. A typical functioning AI model should be able to generalise patterns in the data to tackle new tasks.
Prompt	A prompt is a natural language input that a user feeds to an AI system in order to get a result or output.
Reinforcement Learning	A type of machine learning in which an algorithm learns by interacting with its environment and then is either rewarded or penalised based on its actions.

SDLC	Software Development Life Cycle The process of integrating security considerations and practices into the various stages of software development. This integration is essential to ensure that software is secure from the design phase through deployment and maintenance.
Training data	Training data is the information or examples given to an AI system to enable it to learn, find patterns, and create new content.

ANNEX A

Technical Testing and System Validation

Efficient testing is an essential component for Security by Design and Privacy by Design, ensuring that AI systems meet the needs and expectations of end-users, deliver value, solve real-world problems, and are safe, reliable, accurate, and beneficial for intended users and purposes.

AI systems can be vulnerable to adversarial attacks where malicious actors manipulate inputs to cause the system to malfunction. Testing helps expose these vulnerabilities and implement safeguards to mitigate them. Repeated iterations can improve the design lifecycle and lead to a deeper understanding of how individual AI components are interacting with each other in an eco-system, which should be secured in its totality.

TYPES OF TESTING

There are three main categories of AI testing, each with varying levels of access to the internal workings of the AI system:

White-Box Testing: In white-box testing, you have complete access to the source code, model weights, and internal logic of the AI system. This allows for very detailed testing, focusing on specific algorithms and code sections. However, it requires significant expertise in the underlying technology and can be time-consuming

Grey-Box Testing: Grey-box testing provides partial access to the AI system. You might have knowledge of the algorithms used but not the specific implementation details. This allows for testing specific functionalities without getting bogged down in the intricate code.

Black-Box Testing: Black-box testing treats the AI system as a complete unit, with no knowledge of its internal workings. This is similar to how a user would interact with the system. Testers focus on inputs, outputs, and expected behaviours.

PROS AND CONS OF BLACK BOX TESTING FOR AI

Black-box testing offers several advantages, particularly for securing sensitive information:

Protects Intellectual Property: By not requiring access to source code or model weights, black box testing safeguards proprietary information and trade secrets.

Focus on User Experience: It prioritises real-world functionality from a user's perspective, ensuring the AI delivers the intended results.

Reduced Expertise Needed: Testers do not need in-depth knowledge of the underlying algorithms, making it more accessible for broader testing teams.

However, it is important to note that black box testing alone might not be sufficient for the most comprehensive form of AI testing, because:

Limited Visibility into Issues: Without understanding the internal workings, it can be difficult to pinpoint the root cause of errors or unexpected behaviours.

Challenges in Debugging: Debugging issues becomes more complex as you cannot isolate problems within the specific algorithms or code sections.

CHALLENGES OF AI TESTING

Despite considerable research to uncover the best methods for enhancing robustness, many countermeasures would fail when subjected to stronger adversarial attacks. The recommended approach would be to subject the AI system iteratively to robustness testing with respect to different defences, using a comprehensive testing tool or system, like running a penetration test.

Such a platform would then subject the test system via not just multiple attacks that will scale upwards progressively but would manage the testing cycles with knowledge to optimise the attack evaluation process, e.g., Black box attacks that do not need the help of insiders. In addition, the project teams can also test the robustness of their AI systems against the full set of known and importantly, unknown adversarial attacks.

Other challenges are:

Non-determinism: resulting from self-learning, i.e. AI-based systems may evolve over time and therefore security properties may degrade.

Test oracle problem: where assigning a test verdict is different and more difficult for AI-based systems, since not all expected results are known a priori.

Data-driven paradigm: AI algorithms, where in contrast to traditional systems, (training) data will predominately determine the output behaviour of the AI.

Developing diverse test datasets: Creating datasets that represent various languages, modalities (text, image, audio), and potential attack vectors.

Evaluating performance across modalities: Measuring the effectiveness of attacks and model robustness across different data types.

Limited testing tools: The need for specialised tools to handle the complexities of blended AI models.

LIST OF AI TESTING TOOLS

AI testing is extremely complex, and the tools listed here will not be always able to reduce its complexity and difficulty.

The list of tools for AI model testing will be split into three categories: Offensive AI Testing Tools, Defensive AI Testing Tools, and Governance AI Testing Tools, based on the primary purpose and functionality of the tools.

Offensive AI Testing Tools

Offensive AI Testing Tools are designed to identify vulnerabilities and weaknesses in AI systems by simulating adversarial attacks or malicious inputs. These tools help evaluate the robustness and security of AI models against various types of attacks, such as adversarial examples, data poisoning, and model extraction.

Defensive AI Testing Tools

Defensive AI Testing Tools, on the other hand, focus on enhancing the robustness and resilience of AI systems against potential threats and vulnerabilities. These tools aim to detect and mitigate the impact of adversarial attacks, natural noises, or other forms of corrupted inputs, ensuring that AI models maintain their intended behaviour and performance. Tools that have both offensive and defensive elements are listed under Offensive Testing.

Governance AI Testing Tools

Governance AI Testing Tools are broader in scope and are primarily concerned with assessing the trustworthiness, fairness, and transparency of AI systems. These tools provide frameworks, guidelines, and resources to evaluate and ensure that AI systems align with principles of responsible AI development, deployment, and governance.

Note: The tools mentioned in these tables are often open-source projects or research prototypes that are still under active development. As such, their functionality, performance, and capabilities may change over time, and they might not always work as intended or as described. It is essential to regularly check for updates, documentation, and community support for these tools, as their features and effectiveness may evolve rapidly. Additionally, some tools might have limited support or documentation, requiring users to have a certain level of expertise and familiarity with the underlying concepts and technologies. Therefore, it is crucial to thoroughly evaluate and validate these tools in a controlled environment before deploying them in production or critical systems. Using highly automated settings may result in violations of cybersecurity misuse legislation that forbids any form of scanning or vulnerability scanning unless permission has been granted. For open-source tools, their long-term maintenance, ease of use, other tools integration, reporting and community adoption may be a concern, especially compared to commercial or enterprise-backed AI security solutions.

OFFENSIVE AI TESTING TOOLS

Tool Name Description	License Type	Model Type	Pros	Cons
Gymnasium[3] Malware Environment for single-agent reinforcement learning environments, with popular reference environments and related utilities (formerly Gym)	Open-source	Various	Provides a toolkit for developing and comparing reinforcement learning algorithms. This makes it possible to write agents that learn to manipulate PE files (e.g., malware) to achieve some objective (e.g., bypass AV) based on a reward provided by taking specific manipulation actions.	Limited to the malware domain.
Deep-Pwning[4] Metasploit for Machine Learning.	Open-source	Various	Comprehensive framework for evaluating robustness of ML models against adversarial attacks. Offers flexibility and customisation options, allowing testers to fine-tune attack parameters and strategies to suit their specific testing requirements.	Requires expertise in adversarial machine learning.
Garak[5] LLM Vulnerability Scanner.	Open-source	LLM, Hugging Face models and public ones.	Specifically designed for testing LLMs for vulnerabilities, i.e. probes for hallucination, data leakage, prompt injection, misinformation, toxicity generation, jailbreaks, and many other weaknesses.	Limited to LLMs, relatively new tool.
Adversarial Robustness Toolbox (ART)[6] Library that helps developers and researchers improve the security of machine learning models.	Open-source	Various but not LLMs	Originated from IBM. Was part of a DARPA project called Guaranteeing AI Robustness Against Deception (GARD). Good for research, with modules for attacks, defences, metrics, estimators, and other	Donated by IBM to the Linux Foundation AI & Data Foundation in 2020 and has lost steam, as no version updates since 2020 and has little new activities.

[3] https://github.com/Farama-Foundation/Gymnasium
[4] https://github.com/cchio/deep-pwning
[5] https://github.com/leondz/garak/
[6] https://github.com/Trusted-AI/adversarial-robustness-toolbox

			functionalities to help secure machine learning pipelines against adversarial threats.	Does not directly address LLM security issues like prompt injection.
CleverHans[7] A Python library for creating and evaluating adversarial examples, benchmarking machine learning models against adversarial examples.	Open-source	Various and developing LLM attacks as well.	Good educational and research library, offering a wide range of attack and defence methods via a modular design. Offers a comprehensive set of tools for generating and analysing adversarial examples. These are carefully crafted inputs designed to deceive machine learning models, helping researchers and developers identify weaknesses in their systems. Various evaluation metrics that go beyond standard accuracy measurements. It includes metrics like robustness, resilience, and adversarial success rates, providing a more comprehensive understanding of a model's performance.	Requires a steep learning curve for beginners to understand all the concepts and effectively utilise. Being a static framework, may not inherently keep pace with the rapidly evolving landscape of adversarial attacks and defence strategies. Documentation and tutorials are focused on computer vision models. While CleverHans offers implementations for popular machine learning frameworks like TensorFlow and PyTorch, it may not support all existing frameworks or the latest updates.
Foolbox[8] A Python toolbox for creating adversarial examples that fools machine learning models.	Open-source	Various but not LLMs	Open-source Python library that offers a wide variety of adversarial attack methods, including gradient-based, score-based, and decision-based attacks, hence more feature-rich, compared to the ART toolkit.	Specialised focus on image classification models and does not cover other areas well.

[7] https://github.com/cleverhans-lab/cleverhans
[8] https://github.com/bethgelab/foolbox

			Also provides defences against these attacks. Provides in-depth tools and techniques for analysing adversarial attacks and security in the context of computer vision tasks.	
Advertorch[9] A PyTorch library for generating adversarial examples and enhancing the robustness of deep neural networks.	Open-source	Various but not LLMs	Offers broader set of attack and defence techniques compared to the ART toolkit, such as universal adversarial perturbations and ensemble-based defences. Allows users to seamlessly apply adversarial attacks and defences to PyTorch models.	Steep Learning Curve. Specifically designed for PyTorch models, which may limit its applicability to frameworks or models from different libraries.
Adversarial Attacks and Defences in Machine Learning (AAD) Framework[10] Python framework for defending machine learning models from adversarial examples.	Open-source	Various but not LLMs	Provides a comprehensive set of tools for evaluating and defending against adversarial attacks on machine learning models, which includes a wider range of attack and defence techniques compared to the ART toolkit, covering areas like evasion, poisoning, and model extraction attacks. Defence techniques include adversarial training, defensive distillation, input transformations, and model ensembles.	High complexity.

[9] https://github.com/BorealisAI/advertorch
[10] https://github.com/changx03/adversarial_attack_defence

DEFENSIVE AI TESTING TOOLS

Tool Name Description	License Type	Model Type	Pros	Cons
CNN Explainer[11] Visualisation tool for explaining CNN decisions.	Open-source	CNN	Helps understand and validate CNN model decisions. A good visualisation system to educate new users via visualisation.	Limited to CNNs only and does not cover any other AI vision model.
Nvidia NeMo[12] A framework for generative AI.	Open-source	LLM	Includes guardrails specifically designed for LLM security, e.g. monitoring, and controlling LLM behaviour during inference, ensuring that generated responses adhere to predefined constraints. It provides mechanisms for detecting and mitigating harmful or inappropriate content, enforcing ethical guidelines, and maintaining user privacy. Guardrails are customizable and adaptable to different use cases and regulatory requirements.	Complex and GPU intensive, thus expensive and affects latency.
AllenAI's AllenNLP[13] An Apache 2.0 NLP research library, built on PyTorch, for developing deep learning models on a wide variety of linguistic tasks.	Open-source	LLM	NLP library that includes guardrails for LLM security: tools for bias detection, fairness assessment, and data governance, helping users build and deploy LLMs responsibly. Designed to be flexible and adaptable to different use cases.	Steep learning curve, complex setup, heavily focused on research and experimentation - some of its features might be more geared towards academic research rather than production-level applications. No new features to be added, tool is only maintained.

[11] https://poloclub.github.io/cnn-explainer/
[12] https://github.com/NVIDIA/NeMo
[13] https://github.com/allenai/allennlp

AI GOVERNANCE TESTING TOOLS

Tool Name Description	License Type	Model Type	Pros	Cons
Assessment List for Trustworthy AI[14] Self-assessment tool for trustworthiness of AI systems.	Open-source	Various	Fairly comprehensive framework for evaluating trustworthiness.	Not an automated tool, requires manual assessment.
OECD AI System Classification[15] Classification and tools for developing trustworthy AI systems.	Open-source	Various	Provides guidelines and resources for trustworthy AI development.	Not a specific testing tool, more of a framework.
Charcuterie[16] Collection of tools for data science and machine learning.	Open-source	Various	Provides a variety of tools for data analysis and model development.	Not specifically focused on testing, more of an assistance tool.
LangKit[17] Open-source text metrics toolkit for monitoring language models.	Open-source	LLM	Helps monitor and evaluate LLM performance, safety, and security.	Limited to LLMs, may not cover broader AI system governance.
AI Verify (IMDA)[18] AI governance testing framework and software toolkit that validates the performance of AI systems through standardised tests.	Open-source	Various	A comprehensive tool designed for AI governance and responsible AI practices. It offers a range of features to support organisations in managing and evaluating their AI systems throughout their lifecycle. Provides guidance on bias detection and mitigation, fairness assessments, and stakeholder engagement.	Does not cover LLM.

[14] https://digital-strategy.ec.europa.eu/en/library/assessment-list-trustworthy-artificial-intelligence-altai-self-assessment
[15] https://www.oecd.org/digital/ieconomy/artificial-intelligence-machine-learning-and-big-data/trusted-ai-systems/
[16] https://github.com/moohax/Charcuterie
[17] https://github.com/whylabs/langkit
18 https://aiverifyfoundation.sg/what-is-ai-verify/

Project Moonshot[19] (IMDA) An LLM Evaluation Toolkit designed to integrate benchmarking, red teaming, and testing baselines. It helps developers, compliance teams, and AI system owners manage LLM deployment risks by providing a seamless way to evaluate their applications' performance, both pre- and post-deployment. This open-source tool is hosted on GitHub and is currently in beta.	Open-source	LLM	Moonshot provides intuitive results, so testing unveils the quality and safety of a model or application in an easily understood manner, even for a non-technical user	Does not cover LLM system security.
threat-composer (AWS labs) A simple threat modelling tool to help humans to reduce time-to-value when threat modelling	Open-source	Various	Identify security issues in the context of own AI system. Provides insights on how to improve.	

CSA does not endorse any commercial product or service. CSA does not attest to the suitability or effectiveness of these services and resources for any particular use case. Any reference to specific commercial products, processes, or services by service mark, trademark, manufacturer, or otherwise, does not constitute or imply their endorsement, recommendation, or favouring by CSA.

[19] https://aiverifyfoundation.sg/project-moonshot/

REFERENCES

Articles

1. LinkedIn: How can you design test AI Systems Safely? [20]
2. Elinext: How to test your medical AI for safety[21]
3. Mathworks: The Road to AI Certification: The importance of Verification and Validation in AI[22]
4. Techforgood Institute: AI Verify Foundation: Shaping the AI landscape of tomorrow[23]
5. FPF.Org: Explaining the Crosswalk Between Singapore's AI Verify Testing Framework and The U.S. NIST AI Risk Management Framework[24]
6. FPF.Org: AIVerify: Singapore's AI Governance Testing Initiative Explained[25]
7. Data Protection Report: Singapore proposes Governance Framework for Generative AI[26]

Standard / Regulatory Bodies

8. NIST: Executive Order on Safe, Secure, and Trustworthy Artificial Intelligence[27]
9. ETSI: Securing Artificial Intelligence Introduction[28]
10. CSA: GUIDELINES FOR AUDITING CRITICAL INFORMATION INFRASTRUCTURE JANUARY 2020 [29]
11. IMDA: Singapore launches AI Verify Foundation to shape the future of international AI standards through collaboration[30]

[20] https://www.linkedin.com/advice/1/how-can-you-design-test-ai-systems-safety
[21] https://www.elinext.com/industries/healthcare/trends/step-by-step-guide-how-to-test-your-medical-ai-for-safety
[22] https://blogs.mathworks.com/deep-learning/2023/07/11/the-road-to-ai-certification-the-importance-of-verification-and-validation-in-ai/
[23] https://techforgoodinstitute.org/blog/articles/ai-verify-foundation-shaping-the-ai-landscape-of-tomorrow/
[24] https://fpf.org/blog/explaining-the-crosswalk-between-singapores-ai-verify-testing-framework-and-the-u-s-nist-ai-risk-management-framework/
[25] https://fpf.org/blog/ai-verify-singapores-ai-governance-testing-initiative-explained/
[26] https://www.dataprotectionreport.com/2024/02/singapore-proposes-governance-framework-for-generative-ai/
[27] https://www.nist.gov/artificial-intelligence/executive-order-safe-secure-and-trustworthy-artificial-intelligence/test
[28] https://portal.etsi.org/Portals/0/TBpages/SAI/Docs/2021-12-ETSI_SAI_Introduction.pdf
[29] https://www.csa.gov.sg/docs/default-source/csa/documents/legislation_supplementary_references/guidelines_for_auditing_critical_information_infrastructure.pdf?sfvrsn=8fe3dab7_0
[30] https://www.imda.gov.sg/resources/press-releases-factsheets-and-speeches/press-releases/2023/singapore-launches-ai-verify-foundation-to-shape-the-future-of-international-ai-standards-through-collaboration

ANNEX B

AI Security Defences and their trade-offs

As AI becomes a cornerstone of innovation and national security, protecting its core components becomes paramount. Implementing a multi-layered, dynamically adaptive approach that combines technical safeguards (encryption, air gapping) with robust security protocols (access control, monitoring) and a culture of cyber awareness within organisations is crucial to safeguarding these new "crown jewels" of the digital age.

DEFENDING AI MODELS

Importantly, the models themselves are “fragile” and can be easily attacked using image or text adversarial robustness attacks, or the LLMs could be attacked using malicious prompts.

The table below gives a short summary on the techniques to defend AI systems (non LLM) from examples of adversarial attack.

Defence	Description
Adversarial Training	Train AI model using adversarial samples
Ensemble Models	Utilise blended models to perform a task, and compare their results
Defensive Distillation	Train AI model using class probabilities, instead of discrete class labels, to learn more information about data
Adversarial Detection • Compression • Blurring	Attempt to identify whether an input is an adversarial sample Counter Image Attacks
Explainability	Identify which part of the input had the highest impact in producing the resulting classification. To discover how and why the attack is happening and what makes it work?

Table C1: Countermeasures with description

ADVERSARIAL TRAINING

The most viable method is to introduce adversarial training into the training dataset and retrain the system, i.e., to simply generate and then incorporate adversarial examples into the training process. There are toolsets to do this. In addition, some of the latest image object recognition algorithms e.g. Yolo5, would incorporate adversarial training within this workflow when running training. This will improve model robustness but may not eliminate it.

Hence, the main goal of Adversarial Training is to make a model more robust against adversarial attacks by adding adversarial samples into the model's training dataset. Like adding augmented samples, such as mirrored or cropped images, into the training dataset to improve generalisation. An existing attack algorithm is used to generate these adversarial samples, and there are several variants that utilise different algorithms to generate the adversarial samples for training. Adversarial Training can also be thought of as a brute-force approach, which aims to widen the input distribution of the model so that the boundaries between classes become more accurate.

LIMITATIONS OF ADVERSARIAL TRAINING

Adversarial Training requires additional time to train the model using adversarial samples. Iterative attack algorithms such as Projected Gradient Descent (PGD) requires a much larger time cost, making it difficult to be used for training with massive datasets.

Adversarial Training is mainly effective against the adversarial samples the model was trained against. To note that models, even with Adversarial Training, are susceptible to black-box attacks that utilise a locally trained substitute model to generate adversarial samples. Another technique proposed in "Ensemble Adversarial Training: Attacks and Defences" by Tramèr et. Al.[31] adds random perturbations to an input before running the adversarial attacks on the perturbed input, successfully bypassing the Adversarial Training defence.

[31] https://arxiv.org/abs/1705.07204

ENSEMBLE MODELS

Another intuitive approach to enhance model robustness would be to use multiple models (best to be handling different aspects of the recognition problem) to either detect an attack or to prevent a bypass attack. For example, as depicted in the diagram below, if there were a second AI head detector, the person detector even though fooled by the physical logo on the attacker's shirt, the head detector would not be fooled. Additionally, if there are multiple recognition models, the summation results of different AI systems could still be functional, despite one model being successfully attacked.

LIMITATIONS OF ENSEMBLE MODEL

As multiple models are used on each input, the use of ensemble methods will require additional resources, more memory and computational power for each classification. Ensembles of models may also require more time for development and be more difficult to be used in scenarios where fast, real-time predictions may be required.

DEFENSIVE DISTILLATION

Distillation, also known as Teacher-Student Models, is a procedure which utilises knowledge obtained from a trained 'teacher' Deep Neural Network (DNN) to train a second 'student' DNN. The classes of the labelled training data are known as hard labels, and the output classifications of the 'teacher' DNN are known as soft labels which captures probability distributions indicating how confident the model is for each class. The 'student' DNN is trained using soft labels and the softer predictions make it harder to fool the student which has learnt a more nuanced representation of the dataset. This makes the DNN more robust to adversarial attacks.

LIMITATIONS OF DEFENSIVE DISTILLATION

However, defensively distilled models are still vulnerable to various black-box attacks, due to the strong transferability of adversarial samples generated by these attacks. Modified versions of existing attack algorithms, such as the modified Papernot's attack, have also successfully bypassed defensive distillation.

UTILISING EXPLAINABILITY

A different approach to Adversarial Detection involves the incorporation of Explainable AI (XAI) techniques, which 'explain' the reasons that led to the AI model's prediction. XAI is an emerging field in machine learning that aims to explain predictions made by AI models to improve accuracy, fairness and at the same time aid in the detection of possible anomalies or adversarial attacks. In order to understand the complex black boxes that are AI models, XAI is expected to provide explanations interpretable by humans with clear and simple visualisations.

The main strength of this method is its ability to gain insights into weaknesses present in the model, such as when the reasons leading to the resultant prediction are incorrect. A local interpreter is built to explain the factors that cause adversarial samples to be wrongly classified by the target model.

Furthermore, adversarial samples that exploit these weaknesses can then be generated for use in adversarial training, allowing the model to overcome them. In addition, as the interpretation technique is general to all classifiers, this method can be applied to improve any type of model that supports XAI techniques.

Finally, AI Explainability techniques can be applied to the suspected adversarial inputs, providing visualisations to human operators explaining why these inputs are potentially malicious. The operators can then find out if the detection was a false positive and work on improving the detection model. Otherwise, if the detection was accurate, problems with the defended model can potentially be identified, and the appropriate countermeasures can be applied.

A COMBINATION OF TECHNIQUES

Multiple countermeasures can be used to complement one another, creating a defence-in-depth approach as a higher level of using ensemble defences with differently configured AI models. to would ensure even stronger robustness against adversarial attacks.

DEFENDING YOUR AI SYSTEMS BEYOND THE MODELS

After defending the AI models, since it is still possible to subvert, poison and tamper with the AI system, enhanced infrastructural security measures would have to be added to counter the offensive TTPs that were identified during the risk assessment. The key areas to focus on include:

Continuous monitoring and threat intelligence: Staying informed about the latest threats and vulnerabilities through threat intelligence feeds and security monitoring tools.

Implementing security best practices: This includes basic hygiene measures like patching vulnerabilities, using strong passwords, and implementing multi-factor authentication. Increase system segregation and isolation using containers, VMs, air gaps, firewalls etc.

User awareness training: Educating employees about social engineering tactics and how to identify and avoid phishing attacks.

Security testing and vulnerability assessments: Regularly testing systems for vulnerabilities and implementing security controls to mitigate risks.

Investing in Security Automation: Utilise automation tools to streamline security processes and improve efficiency.

By staying proactive and adapting to the evolving threat landscape that heralds powerful AI-armed APT intruders, organisations can build stronger AI crown jewel defences and mitigate the impact of cyberattacks. Remember, cybersecurity is an ongoing process, not a one-time fix.

AI SECURITY DEFENCES AND THEIR TRADE-OFFS

It is prudent to start implementing countermeasures to protect AI models against attacks early, even as there remain unknowns.

- No one method or countermeasure can reliably defend against all attacks
- Limited awareness and know-how in understanding and operationalising adversarial countermeasures, exacerbated by the complexity of AI models that also makes it difficult to prove how and which defence method will work against some subset of attacks
- As with other changes to the AI model and system, modifications to the model to enhance defences can have impact on model/ system performance

Regardless, traditional security practices continue to be relevant, and provide a good foundation for securing cutting-edge technologies like AI, even as work in this space continues to evolve.

REFERENCES

Standards / Regulatory Bodies

1. **NIST**
    a. NIST: NIST Secure Software Development Framework for Generative AI and for Dual Use Foundation Models Virtual Workshop [32]

NIST: Executive Order 14110 on Safe, Secure, and Trustworthy Artificial Intelligence (October 2023)[33]

b. NIST: AI RMF Knowledge Base[34]
c. NIST: A USAISI Workshop: Collaboration to Enable Safe and Trustworthy AI[35]
d. NIST: USAISI Workshop Slides[36]
e. NIST: Artificial Intelligence[37]
f. NIST: Biden-Harris Administration Announces first ever consortium dedicated to AI Safety[38]
g. NIST: NIST Secure Software Development Framework for Generative AI and for Dual Use Foundation Models Virtual Workshop[39]

2. **ENISA**
    a. ENISA: Artificial Intelligence[40]
    b. ENISA: EU Elections at Risk with Rise of AI-Enabled Information Manipulation [41]
    c. ENISA: Multilayer Framework for Good Cybersecurity Practices for AI [42]
    d. ENISA: Cybersecurity and AI and Standardisation Report[43]
    e. ENISA: Artificial Intelligence Cybersecurity Challenges [44]
    f. ENISA: Is Secure and Trusted AI Possible? The EU Leads the Way [45]
    g. ENISA: Cybersecurity and privacy in AI - Medical imaging diagnosis [46]
    b. ENISA: Cybersecurity and privacy in AI - Forecasting demand on electricity grids

3. **NCSC**
    a. NCSC: Guidelines for secure AI system development [4748]

4. **NSA**
    a. Deploying AI Systems Securely: Best Practices for Deploying Secure and Resilient AI Systems[49]

5. **Singapore**
    a. CSA: Codes of Practice[50]
    b. PDPC: Primer for 2nd Edition of AI Gov Framework [51]
    c. Gov.SG ISAGO[52]
    d. PDPC: Advisory Guidelines On use of Personal Data In AI Recommendation and Decision Systems[53]

6. **Standards and Guides**
    a. ISO/IEC 42001:2023 Information Technology: Artificial Intelligence: Management System[54]
    b. ISO/IEC 23894:2023 Information Technology: Artificial Intelligence: Guidance on Risk Management[55]
    c. OWASP AI Security and Privacy Guide[56]

7. **Others**
    a. PartnershiponAI[57]
    b. AJL[58]
    c. International Telecommunication Union (ITU)[59]
    d. OECD Artificial Intelligence [60]

8. **GitHub Repositories**
   a. Privacy Library of Threats 4 Artificial Intelligence[61]
   b. Guardrails.AI[62]
   c. PyDP: Differential Privacy[63]
   d. IBM Differential Privacy Library[64]
   e. TenSEAL: Encrypting Tensors with Microsoft SEAL[65]
   f. SyMPC: Extends Pysft with SMPC Support[66]
   g. PyVertical: privacy-preserving, vertical federated learning using syft [67]

9. **Articles**
   a. CSO: NIST releases expanded 2.0 version of the Cybersecurity Framework[68]
   b. Technologylawdispatch.com: ENISA Releases Comprehensive Framework
   f. Kim & Chang: South Korea: Legislation on Artificial Intelligence to Make Significant Progress [73]
   g. MetaNews: South Korean Government Says No Copyright for AI Content[74]
   h. East Asia Forum: The future of AI policy in China [75]
   i. Reuters: China approves over 40 AI models for public use in past six months[76]

---

[32] https://www.nist.gov/news-events/events/nist-secure-software-development-framework-generative-ai-and-dual-use-foundation
[33] https://www.nist.gov/artificial-intelligence/executive-order-safe-secure-and-trustworthy-artificial-intelligence
[34] https://airc.nist.gov/AI_RMF_Knowledge_Base/AI_RMF
[35] https://www.nist.gov/news-events/events/usaisi-workshop-collaboration-enable-safe-and-trustworthy-ai
[36] https://www.nist.gov/system/files/documents/noindex/2023/11/20/USAISI-workshop-slides%20%28combined%20final%29.pdf
[37] https://www.nist.gov/artificial-intelligence/artificial-intelligence-safety-institute
[38] https://www.nist.gov/news-events/news/2024/02/biden-harris-administration-announces-first-ever-consortium-dedicated-ai
[39] https://www.nist.gov/news-events/events/nist-secure-software-development-framework-generative-ai-and-dual-use-foundation
[40] https://www.enisa.europa.eu/topics/iot-and-smart-infrastructures/artificial_intelligence
[41] https://www.enisa.europa.eu/news/eu-elections-at-risk-with-rise-of-ai-enabled-information-manipulation
[42] https://www.enisa.europa.eu/publications/multilayer-framework-for-good-cybersecurity-practices-for-ai
[43] https://www.enisa.europa.eu/publications/cybersecurity-of-ai-and-standardisation/@@download/fullReport
[44] https://www.enisa.europa.eu/publications/artificial-intelligence-cybersecurity-challenges
[45] https://www.enisa.europa.eu/news/is-secure-and-trusted-ai-possible-the-eu-leads-the-way
[46] https://www.enisa.europa.eu/publications/cybersecurity-and-privacy-in-ai-medical-imaging-diagnosis
[47] https://www.ncsc.gov.uk/collection/guidelines-secure-ai-system-development
[48] https://www.ncsc.gov.uk/files/Guidelines-for-secure-AI-system-development.pdf
[49] https://www.nsa.gov/Press-Room/Press-Releases-Statements/Press-Release-View/Article/3741371/nsa-publishes-guidance-for-strengthening-ai-system-security/
[50] https://www.csa.gov.sg/legislation/Codes-of-Practice
[51] https://www.pdpc.gov.sg/-/media/files/pdpc/pdf-files/resource-for-organisation/ai/primer-for-2nd-edition-of-ai-gov-framework.pdf
[52] http://go.gov.sg/isago
[53] https://www.pdpc.gov.sg/guidelines-and-consultation/2024/02/advisory-guidelines-on-use-of-personal-data-in-ai-recommendation-and-decision-systems
[54] https://www.iso.org/standard/81230.html
[55] https://www.iso.org/standard/77304.html
[56] https://owasp.org/www-project-ai-security-and-privacy-guide/
[57] https://partnershiponai.org/
[58] https://www.ajl.org
[59] https://www.itu.int/
[60] https://www.oecd.org/digital/artificial-intelligence/
[61] https://plot4.ai/
[62] https://github.com/guardrails-ai/guardrails
[63] https://github.com/OpenMined/PyDP
[64] https://github.com/IBM/differential-privacy-library
[65] https://github.com/OpenMined/TenSEAL
[66] https://github.com/OpenMined/SyMPC
[67] https://github.com/OpenMined/PyVertical
[68] https://www.csoonline.com/article/1310046/nist-releases-expanded-2-0-version-of-the-cybersecurity-framework.html
[73] https://www.kimchang.com/en/insights/detail.kc?sch_section=4&idx=26935
[74] https://metanews.com/south-korean-government-says-no-copyright-for-ai-content/
[75] https://eastasiaforum.org/2023/09/27/the-future-of-ai-policy-in-china/
[76] https://www.reuters.com/technology/china-approves-over-40-ai-models-public-use-past-six-months-2024-01-29/

for Ensuring Cybersecurity in the Lifecycle of AI Systems [69]

c. DataGuidance: ENISA releases four reports on AI and Cybersecurity [70]
d. KoreaTimes: Korea issues first AI ethics checklist [71]
e. Dig.watch: South Korea to boost trust in AI with watermarking initiative[72]
j. DataNami: Artificial Intelligence Leaders Partner with Cloud Security Alliance to Launch the AI Safety Initiative [77]
k. World Economic Forum: Why we need to care about responsible AI in the age of the algorithm [78]
l. What is Confidential Computing? | Data Security in Cloud Computing (Anjuna)[79]
m. What is Confidential Computing? (NVIDIA Blog)[80]

[69] https://www.technologylawdispatch.com/2023/06/data-cyber-security/enisa-releases-comprehensive-framework-for-ensuring-cybersecurity-in-the-lifecycle-of-ai-systems/
[70] https://www.dataguidance.com/news/eu-enisa-releases-four-reports-ai-and-cybersecurity
[71] https://m.koreatimes.co.kr/pages/article.asp?newsIdx=352971
[72] https://dig.watch/updates/south-korea-to-boost-trust-in-ai-with-watermarking-initiative
[77] https://www.datanami.com/this-just-in/artificial-intelligence-leaders-partner-with-cloud-security-alliance-to-launch-the-ai-safety-initiative/
[78] https://www.weforum.org/agenda/2023/03/why-businesses-should-commit-to-responsible-ai/
[79] https://www.anjuna.io/blog/confidential-computing-a-new-paradigm-for-complete-cloud-security
[80] https://docs.nvidia.com/nvtrust/index.html

Advisory and Cloud Providers

10. **Google**
    a. Google: Introducing Google's Secure AI Framework[81]
    b. Google: OCISO Securing AI Similar or Different? [82]

11. **Microsoft**
    a. Microsoft: Azure Platform[83]
    b. Microsoft: Introduction to Azure Security[84]
    c. Microsoft: Responsible AI[85]
    d. Microsoft: Responsible AI Principles and Approach[86]
    e. Microsoft: AI Fairness Checklist[87]
    f. Microsoft: AI Lab project: Responsible AI dashboard[88]
    g. Microsoft: Our commitments to advance safe, secure, and trustworthy AI[89]

12. **Amazon Web Services**
    a. Secure approach to generative AI[90]

[81] https://blog.google/technology/safety-security/introducing-googles-secure-ai-framework/
[82] https://services.google.com/fh/files/misc/ociso_securing_ai_different_similar.pdf
[83] https://www.microsoft.com/en-us/ai/ai-platform
[84] https://learn.microsoft.com/en-us/azure/security/fundamentals/overview
[85] https://www.microsoft.com/en-us/ai/responsible-ai
[86] https://www.microsoft.com/en-us/ai/principles-and-approach
[87] https://www.microsoft.com/en-us/research/project/ai-fairness-checklist/
[88] https://www.microsoft.com/en-us/ai/ai-lab-responsible-ai-dashboard
[89] https://blogs.microsoft.com/on-the-issues/2023/07/21/commitment-safe-secure-ai/
[90] https://aws.amazon.com/ai/generative-ai/security/

Consultancies

13. **Deloitte**
    a. Deloitte: Trustworthy AI[91]
    b. Deloitte: Omnia AI[92]
    c. Deloitte AI Institute: The State of Generative AI in the Enterprise: Now Decides Next[93]

14. **EY**
    a. EY: How to navigate generative AI use at work[94]
    b. EY: EY's commitment to developing and using AI ethically and responsibly[95]
    c. EY: Making Artificial Intelligence and Machine Learning trustworthy and ethical[96]

15. **KPMG**
    a. KPMG: AI security framework design[97]
    b. KPMG: Trust in Artificial Intelligence[98]

16. **PwC**
    a. PwC: Balancing Power and Protection: AI in Cybersecurity and Cybersecurity in AI [99]
    b. PWC: What is Responsible AI[100]

17. **Research Papers**
    a. TASRA: a Taxonomy and Analysis of Societal-Scale Risks from AI[101]
    b. China Academy of Information and Communications Technology: Whitepaper on Trustworthy Artificial Intelligence[102]
    c. Trustworthy AI: From Principles to Practices [103]

[91] https://www2.deloitte.com/us/en/pages/deloitte-analytics/solutions/ethics-of-ai-framework.html
[92] https://www2.deloitte.com/ca/en/pages/deloitte-analytics/articles/omnia-artificial-intelligence.html
[93] https://www2.deloitte.com/us/en/pages/deloitte-analytics/articles/advancing-human-ai-collaboration.html#
[94] https://www.ey.com/en_us/consulting/video-how-to-navigate-generative-ai-use-at-work
[95] https://www.ey.com/en_gl/ai/principles-for-ethical-and-responsible-ai
[96] https://assets.ey.com/content/dam/ey-sites/ey-com/en_gl/topics/consulting/ey-making-artificial-intelligence-and-machine-learning-trustworthy-and-ethical.pdf
[97] https://kpmg.com/us/en/capabilities-services/advisory-services/cyber-security-services/cyber-strategy-governance/security-framework.html
[98] https://kpmg.com/xx/en/home/insights/2023/09/trust-in-artificial-intelligence.html
[99] https://www.pwc.com/m1/en/publications/balancing-power-protection-ai-cybersecurity.html
[100] https://www.pwc.com/gx/en/issues/data-and-analytics/artificial-intelligence/what-is-responsible-ai.html
[101] https://arxiv.org/abs/2306.06924
[102] http://www.caict.ac.cn/english/research/whitepapers/202110/P020211014399666967457.pdf
[103] https://arxiv.org/abs/2110.01167