CAMPUS CYBER

HUB FRANCE IA

# Analysis of attacks on AI systems

January 2026

## Table of Contents

# 1 Introduction

## 1.1 Context

Artificial intelligence (AI), whether predictive[1] or generative[1] is transforming many sectors of activity. While these technologies offer unprecedented opportunities, they also expose organizations to new cybersecurity risks.

Like traditional systems, AI systems[2] (AIS) must be protected against the multitude of possible attacks. These AIS also present specific vulnerabilities, characteristics of their architecture and functioning, which rely on complex algorithms and large data sets. It is therefore essential to implement security measures adapted to these specificities.

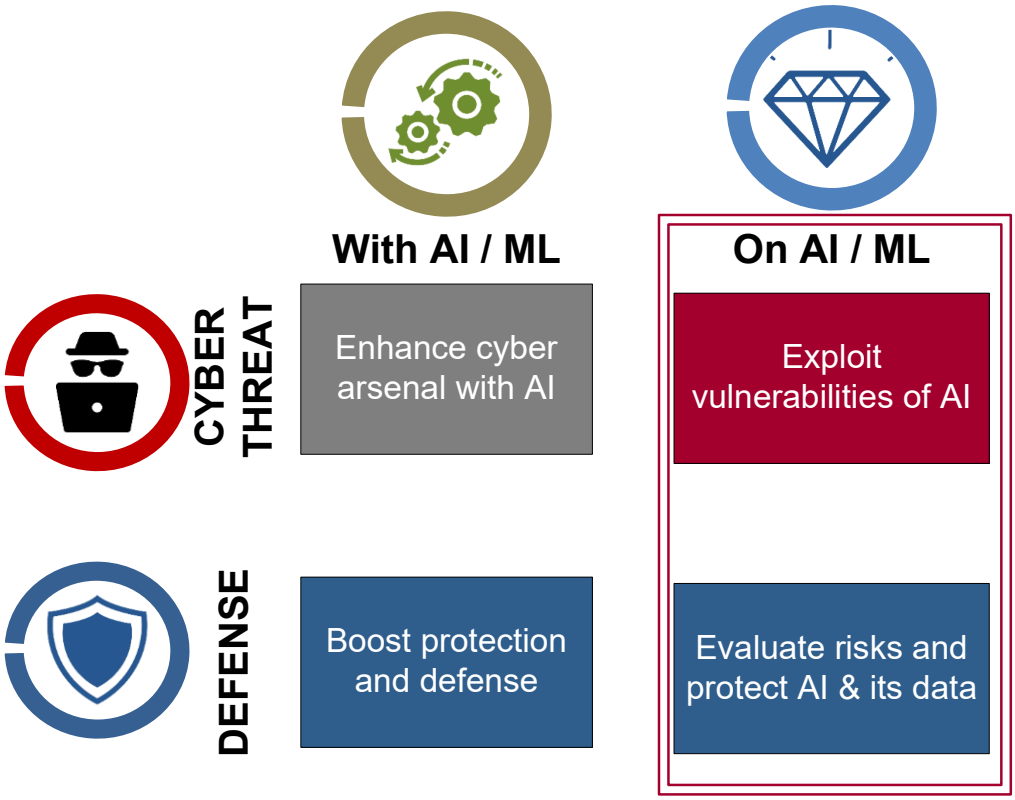



*Figure 1 – AI and cybersecurity*[3]

First, let us note that AI plays different roles in cybersecurity or cybercrime, as seen in Figure 1:

- With AI

[1] Term explained in section 8 Glossary

[2] Throughout the following, we will refer to AI System (AIS) as "machine-based system that is designed to operate with varying levels of autonomy and that may exhibit adaptiveness after deployment, and 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". This definition is taken from the AI Act [6].

[3] According to https://wiki.campuscyber.fr/IA_et_cybers%C3%A9curit%C3%A9

  - Attacker side (crime): attackers can generate new attack techniques that lead to crime. For example, data poisoning, or "deepfake" (the most famous example is the deepfakes used for CEO fraud).
  - Defender side (security): defenders can strengthen their protection techniques, for example through AI techniques for detecting anomalies or impersonation attempts.
- On AI
  - Attacker side (crime): attackers can develop new forms of attacks, such as data poisoning which degrades performance and therefore the quality of the AIS responses.
  - Defender side (security): defenders must implement appropriate and reactive countermeasures to defend against these new attacks, for example by encrypting data.

This paper focuses on these attacks **on AI** (right in Figure 1).

This document aims to provide an in-depth overview of major cyber attacks targeting both predictive and generative AI systems. In order to deal with these attacks, the intervention of both AI and cybersecurity experts is required; it is therefore essential that both types of experts understand the context and issues of these attacks. This document addresses, in an educational manner, the challenge of AI in cyber by specifying the context and issues of the attacks and by using language and references common to both fields of expertise.

The focus is on **intentional**, offensive threats that seek to compromise the confidentiality, integrity, or availability of these systems. However, it is important to note that AI systems can also be exposed to other risks, such as design errors, biases or data governance flaws. These vulnerabilities, while crucial, are more a matter of ethical and model robustness challenges than of cybersecurity in the strict sense and are beyond the scope of this document. Similarly, attacks on the legal and judicial aspects of systems incorporating AI are not considered in this document.

It is important to understand that attacks targeting AI systems are distinguished by their unique nature, exploiting vulnerabilities specific to these technologies, which we can illustrate with a few examples.

- Training data *poisoning*: by subtly inserting erroneous data into the training set, attackers can alter the model's behavior and cause prediction errors with potentially serious consequences.
- *Generation of content* biased or malicious: imagine a text generation model trained with manipulated data to systematically associate an ethnic group with hateful speech; the content generated by this model would risk spreading discrimination and hatred. The case of *prompt injection* is a poisoning of the prompt data by a malicious third party that can produce a false, offensive or

discriminatory response, or even one that contradicts what the system is "entitled" to say (in Figure 2, the LLM must not say how to build a bomb).

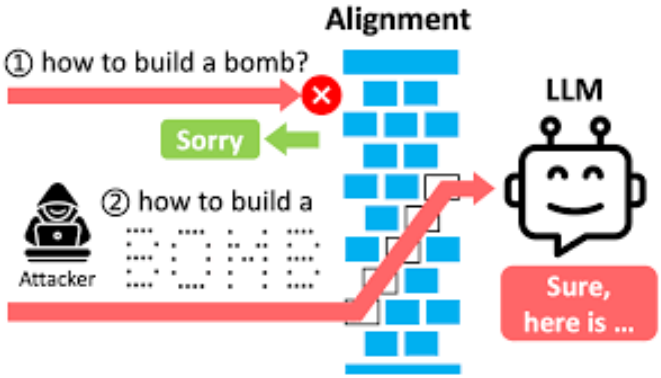



*Figure 2 - Schematic example of prompt injection*[4]

The complexity and opacity of AI algorithms make detecting and neutralizing these attacks particularly challenging. Interpreting the mechanisms of an attack and assessing its impact on the system is often a complex task.

For each type of attack, we offer an in-depth analysis that will be structured around

- Stages of the AI system lifecycle,
- Matching MITRE ATLAS tactics, possibly enhanced to reflect the latest developments in generative AI.

This dual approach allows for a better understanding of attack mechanisms, potential entry points and attackers' objectives.

The description of the attacks will be completed by proposals for prevention and mitigation measures.

## 1.2 References used

This document is based on reference work from NIST, MITRE ATLAS, OWASP and ANSSI recommendations, ensuring comprehensive and up-to-date coverage (as of the date of publication of this document) of threats (references described in section 2.3).

The objective is to provide operational teams with the knowledge and tools necessary to effectively anticipate, detect and counter attacks targeting AI systems, with the aim of ensuring their security and reliability.

The formalization of the lifecycle of an AI system presented here also uses references, such as the OECD. These formalizations are detailed in section 2.1.

[4] According tohttps://arxiv.org/pdf/2402.11753

Other references are also used for the qualitative evaluation of attacks: CyberDico from ANSSI [4], CVSS indicator [19], EBIOS RM method from ANSSI [5] (references described in section 2.4).

Section 7 lists the main references cited in the document.

The document is structured as follows:

- Section 2: description of attacks against AI systems, with the lifecycle and protection systems of AI systems, the main attack repositories, qualitative assessments of attacks, our taxonomy of attacks and the description of the main categories of attacks;
- Section 3: presentation of some recent or lesser-known AI techniques (RAG, agentic systems, federated learning, cryptography and adversarial attacks);
- Section 4: description of measures for protection, prevention and remediation;
- Section 5: presentation of pedagogical fact sheets with an analysis of the main attacks identified in our taxonomy;
- Section 6: conclusion;
- Section 7: presentation of the main reference documents used in this document;
- Section 8: presentation of a glossary of the main terms used here in AI and cyber;
- Appendices 1 and 2: lists of prevention and remediation measures used in the fact sheets.

# 2 Understanding Attacks on AI Systems

Describing attacks against AI systems requires a structured framework for categorizing threats, which is why this document introduces a *taxonomy of* attacks on AI (excluding generic attacks on computer systems).

The first level of this taxonomy is based on the phases of an AI project's lifecycle. This approach allows experts, engineers, and other AI practitioners to quickly identify the most relevant threats based on their current stage of development.

The following levels describe, for the corresponding phase, the types of attacks possible for the AIS, which obviously depend on the techniques used by the AIS being considered. The scope of analysis covers the main predictive AI and generative AI systems, without being totally exhaustive (see some examples not covered in section 3).

Choosing how to formalize an AI project's lifecycle is fundamental as it forms the foundation for the classification of attacks. The chosen approach is outlined in detail, in particular with regard to the selection of the most appropriate life-cycle formalization from among the models proposed by the OECD, ISO, ANSSI and ENISA.

## 2.1 The importance of the lifecycle

A well-defined lifecycle helps break down the development of an AI system into distinct phases: it is a classical tool for data scientists when developing an AI system. Each phase presents specific vulnerabilities, and the lifecycle therefore serves as an entry point for identifying potential attacks. The goal is to choose a formalism that is granular enough to capture the nuances of the different stages, while remaining generic enough to be applicable to a wide variety of AI systems.

### *2.1.1 Stages of the lifecycle*

The **lifecycle of an AI system**, from its design to its operation, includes a series of interdependent steps, which represent as many potential entry points as possible for malicious attacks.

Here are the main stages of the lifecycle of an AI system:

- **Planning and design**: from the design stage of the system, decisive choices are made in terms of architecture, data and algorithms, directly impacting its robustness against attacks.
- **Data collection and processing**: this step, essential to the system's learning, can be compromised by the introduction of erroneous, biased, or manipulated data. The AI lifecycle is of course closely linked to the data lifecycle.
- **Construction of the model / adaptation of an existing model**: it is during this phase that the system learns from the data. Poisoning attacks on this data can be carried out to alter its behavior.
- **Test/evaluation/verification**: before deployment, the system is tested and evaluated. It is crucial to ensure that these tests take into account the risks of attacks and that the security measures put in place are effective.
- **Provision/use/deployment**: once deployed, the system is exposed to new threats and vulnerabilities. Security must be integrated into the design of the deployment architecture.
- **Exploitation/maintenance**: throughout its life, the system must be maintained and updated regularly to re-start learning, if necessary, correct security flaws and counter new threats. Continuous performance evaluation is a valuable indicator to follow to identify weak signals of abnormal events.
- **Decommissioning/scrapping**: the end of life of an AI system also requires special attention in terms of security, particularly for secure data deletion and system deactivation.

Each stage of the lifecycle presents specific risks that are essential to consider for ensuring the safety and reliability of AI systems.

### 2.1.2 The main lifecycle formalisms

#### 2.1.2.1 The OECD lifecycle[5]

The OECD (Organization for Economic Co-operation and Development) lifecycle covers the stages mentioned above and clearly shows the possible feedback loops at each stage: in fact, the process of developing an AIS is iterative and there is always a need to go back if the results obtained are not satisfactory.

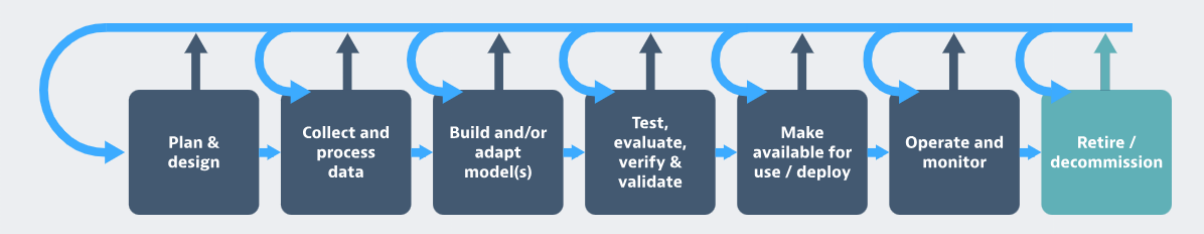
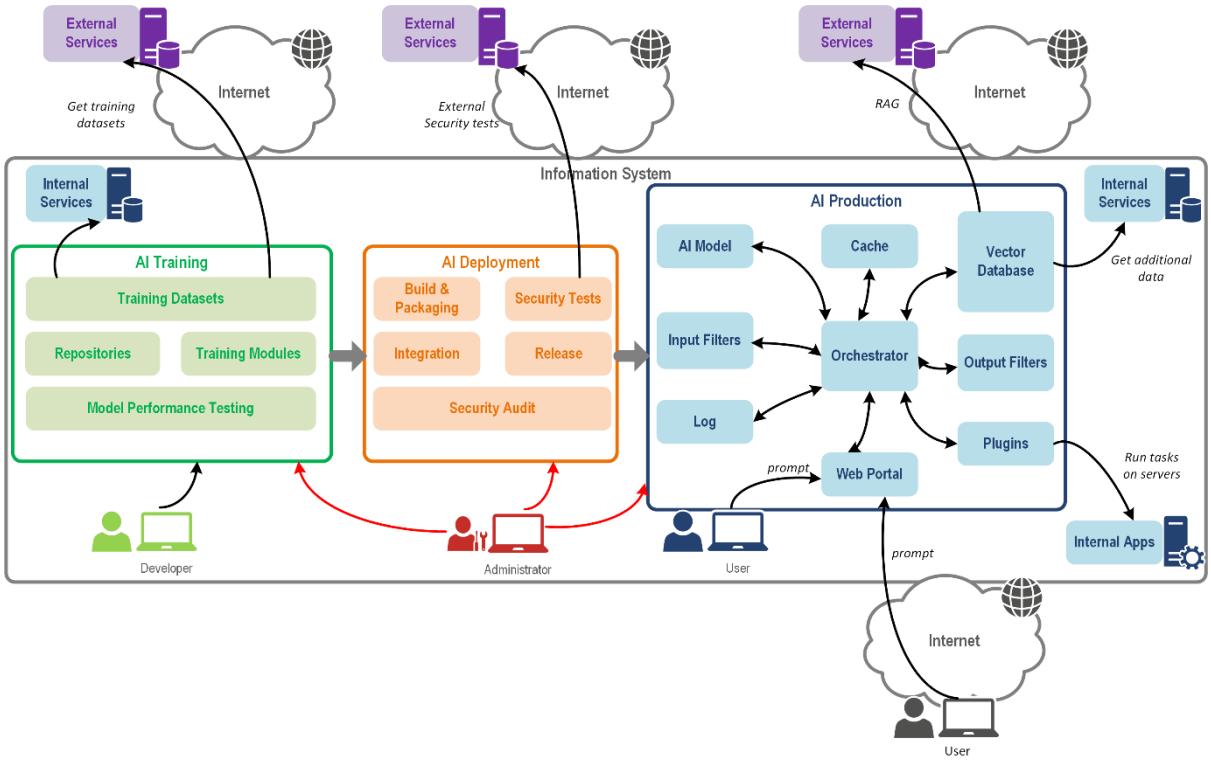



*Figure 3 – Formalization of the lifecycle of an AI project by the OECD*

#### 2.1.2.2 The ANSSI lifecycle

The ANSSI lifecycle [1] (National Agency for the Security of Information Systems) includes 3 phases (therefore fewer than the OECD), and seeks to highlight, at each stage, access to data sources, libraries, internal and external services, which are the targets of classic cyber-attacks: Let's not forget that any attack on an AIS goes through a classical entry path. The ANSSI lifecycle does not include the decommissioning phase.




[5] https://oecd.ai/en/ai-principles

*Figure 4 – Formalization of the lifecycle of an AI project by ANSSI*

### *2.1.2.3 The ISO lifecycle*

The ISO (International Organization for Standardization) standard provides a slightly different lifecycle structure [14] with a description of subtasks per phase. In the monitoring phase, the tasks of *continuous validation* and *re-evaluation* are not detailed in other formalisms.

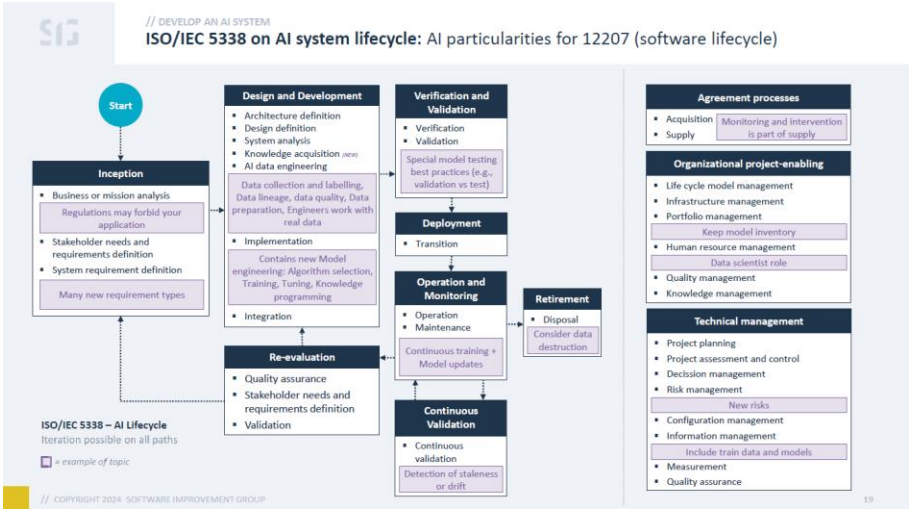



*Figure 5 – Formalization of the lifecycle of an AI project by ISO*

### *2.1.2.4 The lifecycle of ENISA*

The ENISA [16] (European Cybersecurity Agency) formalism details the planning and design phase well, but very little the maintenance phases and not at all decommissioning.

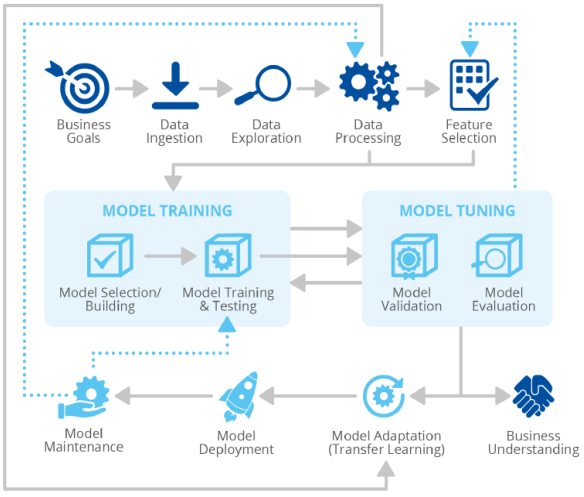



*Figure 6 – Formalization of the lifecycle of an AI project by ENISA*

### *2.1.3 Lifecycle Choices: A Comparative Analysis*

After examining the different models proposed by the OECD, ISO, ANSSI and ENISA, we compared their characteristics (Figure 7).

- **OECD**: the OECD lifecycle, with its seven distinct phases, provides sufficient granularity to cover the entire development process of an AI system, from planning to disposal.
- **ISO**: This international standard offers a more detailed lifecycle, particularly regarding verification and validation aspects. While useful, this additional granularity is not essential for our goal of attack classification. Furthermore, the ISO phases can be easily mapped onto those of the OECD.
- **ANSSI**: ANSSI proposes a more macroscopic lifecycle, focused on the training, deployment, and production phases. This model, while relevant, lacks the granularity for fine-grained attack classification. However, we have integrated ANSSI's vision by superimposing its phases on those of the OECD. For example, ANSSI's AI training phase encompasses the first four phases of the OECD cycle (planning, data collection, model building, and testing/evaluation), as shown in Figure 7 below.
- **ENISA**: ENISA proposes a lifecycle closer to that of the OECD, with clearly identifiable phases such as training and model deployment. However, the OECD model offers a more comprehensive structure that is better suited to our needs.


*Figure 7 – Comparison of the formalizations of the lifecycles of the OECD, ANSSI, ISO and ENISA*

We opted for the OECD formalism, which offers sufficient granularity to cover the entire development process of an AI system, and which aligns well with other formalisms, in particular the one adopted by ANSSI, which is the format used in the attack fact sheets (see section 5):

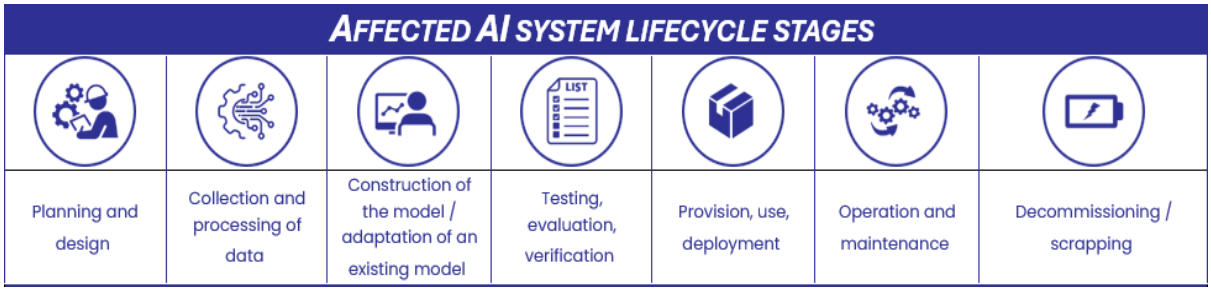



*Figure 8 – Representation of the OECD lifecycle in the attack description fact sheets*

## 2.2 Protect the AI system

Usually, cybersecurity processes focus on protecting the final model (the computer program used in production) and infrastructure (network access and machines). These processes must, of course, remain in place. To attack an AI system, you must first penetrate it, and cyber processes are there to protect this first step (see section 4.1.1 below).

However, an AIS presents a larger attack surface through various components:

- The data (training and those used in production to query the model),
- The final model (and associated parameters),
- The model's inputs/outputs, as well as interactions with humans or other computer systems,
- The processes for training, testing, deploying, and operating the model,
- As well as the necessary infrastructure.

To protect AI, we will therefore have to protect, during the different stages of the life, **all these elements**: data, models and infrastructures.

Securing data requires protecting training data during training (for example, data poisoning, e.g. by swapping class labels, will result in massive degradation of classification performance) and deployment (poisoning the prompt or *prompt injection*, for example, will deteriorate the quality of the response returned by the AIS).

In the particular case of a generative AI system using a RAG mechanism (see section 3.1), a knowledge base (e.g., user-specific or company-specific data) is used. During the training phase, the associated embeddings and the vector representation of this knowledge base are calculated. During exploitation, we use this representation to enrich the prompt. We can therefore see that, in the case of the RAG, we can attack the data of the knowledge base during training (for example by poisoning the vector representations) and during exploitation.

## 2.3 Overview of key attack frameworks

The information security community has developed several attack frameworks to help data scientists navigate the complex threat landscape facing AI systems.

These frameworks provide methodological guides, best practices, and tools to identify, assess, and mitigate security risks associated with AI.

In this document, we present a harmonized synthesis of four major reference frameworks, covering both general threats to AI (NIST AI 100-2e2023 [7], MITRE ATLAS [17]), as well as risks specific to generative and machine learning models (OWASP Top 10 LLM [10], OWASP Top 10 ML [11]) as well as ANSSI recommendations for generative AI [1].

By understanding the principles and recommendations of these frameworks, AI professionals will be able to develop and deploy more robust, resilient, and secure AI models. The goal is to equip both AI experts and AI project managers with the knowledge needed to integrate security from the beginning of their AI projects' lifecycle, from design to production, and thus contribute to a more reliable and trustworthy AI ecosystem.

In the following sections, we explore these frameworks and their practical applications for securing AI systems in detail. We begin with the NIST AI 100-2e2023 framework, then explore the MITRE ATLAS Knowledge Base, and then the specific risks to generative and machine learning models identified by the OWASP Top 10 LLM and Top 10 ML. Finally, we analyze ANSSI's recommendations for strengthening the security of generative AI.

### *2.3.1 NIST.AI.100-2e2023*

**What is NIST?**

NIST[6] (*National Institute of Standards and Technology*) is an agency of the United States Department of Commerce whose mission is to promote American innovation and industrial competitiveness by advancing measurement science, standards, and technology.

In the context of artificial intelligence, NIST plays a crucial role in developing guidelines, assessments, and data to support the development, use, and reliability of artificial intelligence, particularly in security matters.

**What is NIST.AI.100-2e2023?**

NIST.AI.100-2e2023 [7] is a report published by NIST that provides a comprehensive taxonomy and standardized terminology for *Adversarial Machine Learning* (AML). It aims to help AI experts, security engineers, and other stakeholders navigate the complex and ever-changing AML landscape.

What are the main points to remember from this framework?

- A four-dimensional taxonomy of attacks

[6] https://www.nist.gov/

1. **The learning method and lifecycle phase:** this dimension considers the type of learning (supervised, unsupervised, etc.) and the phase of the model lifecycle (training, deployment, etc.). This is fundamental because vulnerabilities differ depending on the method and phase.
2. The attacker's objectives
   - **Disruption of availability:** make the model unavailable or significantly slow it down, preventing its normal use,
   - **Violation of integrity:** change model predictions to get incorrect results,
   - **Compromise of confidentiality:** extracting sensitive information from the model or its training data,
   - **Abuse (for generative AI):** exploit the model for unintended malicious uses, such as generating inappropriate content.
3. **Attacker's abilities:** definition of the means used by the attacker: control of training data, ability to submit queries, etc.
4. **Attacker's knowledge:** attacker's level of knowledge about the model and its environment (white box, gray box, black box).

- **A description of common attacks:** the report details the most common attacks, and concrete examples of attacks are provided for each category.
- **Mitigation techniques:** the report explores the main techniques for defending against attacks and their limitations.

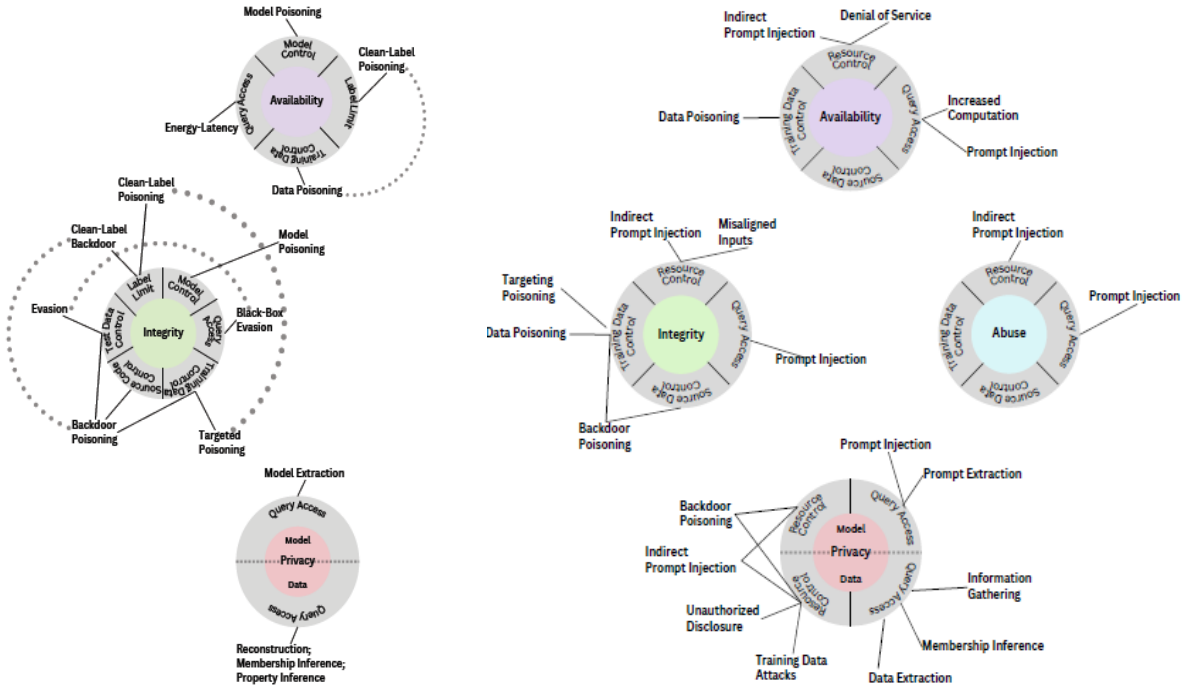



*Figure 9 – Attacks on predictive AI (left) and generative AI (right) according to the NIST.AI.100-2e2023 framework*

### *2.3.2 MITRE ATLAS*

**What is MITRE?** MITRE or MITRE Corporation[7] is an American non-profit organization. It operates federally funded research and development centers that support various U.S. government agencies in the fields of aviation, defense, healthcare, homeland security, and cybersecurity, among others.

**What is MITRE ATLAS?** MITRE ATLAS[8] (*Adversarial Threat Landscape for Artificial-Intelligence Systems*) is a repository that provides a detailed taxonomy of adversarial tactics and techniques targeting machine learning systems. It can be thought of as an encyclopedia of attacks against AI. The main takeaway from this framework is the organization of attacks into several levels:

- **The tactic:** the attacker's overall objective,
- **The technique:** the specific methods used to achieve the tactic,
- **The sub-technique (if applicable):** more precise variations of the technique.

Each tactic and technique are documented with detailed descriptions, examples, and references. Here's a summary of the main tactics.

- **Reconnaissance:** gathering information about the target AI system, its components (model, training data, etc.), and its environment. The goal is to identify potential vulnerabilities,
- **Resource development:** acquisition or creation of resources necessary for the attack, such as malicious data or specific tools,
- **Initial access:** obtaining an initial point of access to the AI system, whether through a software flaw, misconfiguration, or manipulation,
- **ML Model Access:** gaining access, often unauthorized, to the Machine Learning model itself, its parameters or its architecture,
- **Execution:** executing code or commands on the AI system, usually to modify its behavior or extract information,
- **Persistence:** maintaining access to the AI system after the initial attack, for subsequent actions,
- **Privilege escalation:** obtaining higher access rights over the AI system to perform more harmful actions,
- **Defense Evasion:** Bypassing the security mechanisms put in place to protect the AI system,
- **Credential Access:** obtaining credentials (usernames, passwords, API keys, etc.) to access the system,
- **Discovery:** identification of target AI system's components and resources, such as models, datasets, and APIs,
- **Collection:** retrieving data or information from the AI system, such as training data, model predictions, or identifiers,

[7] https://www.mitre.org/
[8] https://atlas.mitre.org/

- **ML Attack Staging:** setting up the elements necessary to execute an attack against the Machine Learning model,
- **Exfiltration:** transferring stolen data or sensitive information out of the AI system,
- **Impact:** compromising the end goal of the attack, such as denial of service, degradation of model performance, or manipulation of results.

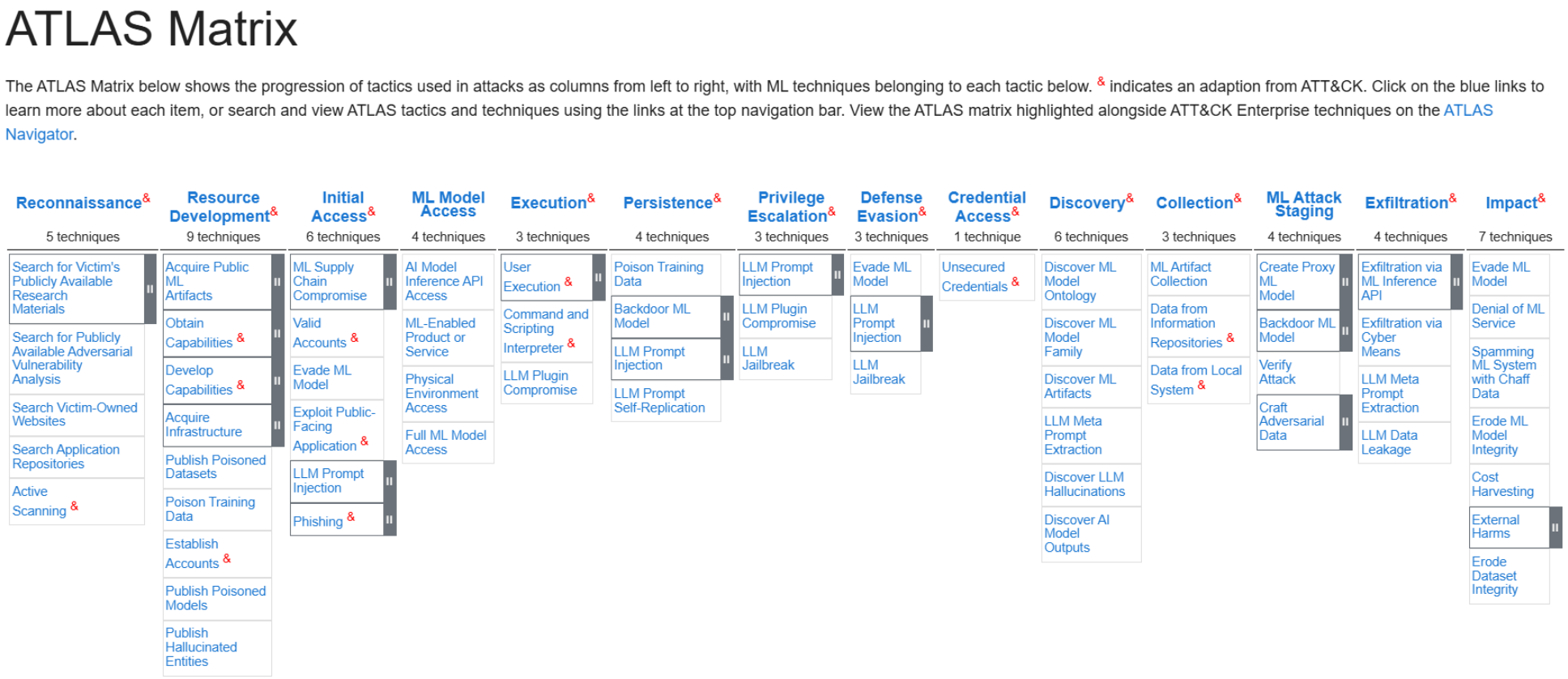



*Figure 10 – The MITRE ATLAS reference system [17]*

### *2.3.3 OWASP TOP 10 LLM & TOP 10 ML*

**What is OWASP?**

The *Open Worldwide Application Security Project* (OWASP)[9] is a non-profit foundation working to improve software security through its community-led open-source software projects, hundreds of local chapters worldwide, tens of thousands of members, and the organization of local and international conferences.

**What are the OWASP Top 10 ML and OWASP Top 10 LLM reference documents?**

The OWASP Top 10 ML [11] and the OWASP Top 10 LLM [10] are a list of the ten most critical security vulnerabilities for Machine Learning systems and LLMs (*Large Language Models*), prepared by security experts. These documents are a valuable resource for understanding potential threats and implementing effective protective measures.

What are the main points to remember from these benchmarks?

**The OWASP Top 10 ML vulnerabilities are**:

1. **Input Manipulation**: deliberately modifying input data to mislead the model. A generic term that includes adversarial attacks,
2. **Data poisoning**: manipulation of training data to compromise model behavior,
3. **Model Inversion:** reverse engineering the model to extract information from it,
4. **Membership inference:** manipulating the model's training data to cause it to behave in a way that exposes sensitive information,
5. **Model theft:** unauthorized access and theft of the trained model (access to these parameters),
6. **Supply chain**: modification or replacement of a machine learning library or model used by a system. This may also include data associated with machine learning models,
7. **Transfer learning**: An attacker trains a model on a task, then transfers their knowledge to the legitimate model so that it behaves in an undesirable manner,
8. **Model Skewing**: manipulation of the distribution of training data to make the model behave in undesirable ways,
9. **Output Integrity**: modifying or manipulating the output of a machine learning model in order to change its behavior or harm the system in which it is used,
10. **Model poisoning**: manipulation of model parameters to make it adopt undesirable behavior.

**The OWASP Top 10 LLM vulnerabilities are**:

1. **Prompt injection:** input manipulation to control LLM behavior,
2. **Sensitive Information Disclosure**: exposure of sensitive data, proprietary algorithms or confidential details through the LLM output,

[9] https://owasp.org/

3. **Supply chain**: LLM supply chains are susceptible to various vulnerabilities, which can affect the integrity of training data, models, and deployment platforms. These risks can result in biased results, security breaches, or system failures,
4. **Data and model poisoning**: manipulation of pre-training data, of *fine tuning* or *embeddings* to introduce vulnerabilities, backdoors or biases.

   Poisoning can also allow the implementation of a backdoor. These backdoors can leave the model's behavior intact until a specific trigger causes it to change,

5. **Improper Output Handling**: insufficient validation, sanitation or processing of outputs generated by models, before they are transmitted downstream to other components and systems.

   Since the content of the LLM generation can be controlled by input prompts, this behavior amounts to giving users access to additional functionality.

6. **Excessive Agency**: performing harmful actions in response to unexpected, ambiguous, or manipulated outputs from an LLM, without regard to what is causing the LLM to malfunction,
7. **System Prompt Leakage**: disclosure of sensitive information that may be contained in system prompts or instructions used to direct model behavior. Examples include information about bypassing system safeguards, improper separation of privileges, etc.
8. **Vector and Embedding Weaknesses**: exploitation of the generation, storage or retrieval of vectors and embeddings, particularly in systems using retrieval-augmented generation (RAG) with large language models (LLM). The goal here is to inject harmful content, manipulate model results, or access sensitive information,
9. **Misinformation**: production by models of false or misleading information that appears credible. This vulnerability can lead to security breaches, reputational damage, and legal liability,
10. **Unbounded Consumption**: excessive and uncontrolled demands on the model, which may lead to denial of services, economic losses, theft of the model, or degradation of the service.

Note that OWASP has just released a new document on attacks on agent systems[10] which we will not take into account in this document.

### *2.3.4 ANSSI recommendations*

In its document published in April 2024 [1], ANSSI issues 35 recommendations for the implementation of "secure-by-design" AI (generative AI). At each of the 3 main phases of the lifecycle of an AI system, the users and environments involved are different:

[10] https://genai.owasp.org/resource/agentic-ai-threats-and-mitigations/

- Training phase: data scientists use a development environment,
- Integration and deployment phase: data scientists and IT administrators use a CI/CD environment,
- Operational exploitation phase: the end customer (internal or external) uses a production environment.

ANSSI proposes 35 recommendations valid for generative AI (and very often also predictive AI) which complement the "usual" security requirements:

- **17 general recommendations**
  - **R1**: integrate security into all phases of an AI system's lifecycle,
  - **R2**: conduct a risk analysis on AI systems before the training phase,
  - **R3**: evaluate the trust level of libraries and external modules used in the AI system,
  - **R4**: assess the level of trust of external data sources used in the AI system,
  - **R5**: apply DevSecOps principles across all phases of the project,
  - **R6**: use secure AI model formats,
  - **R7**: take data confidentiality issues into account from the design stage of the AI system,
  - **R8**: take into account the need-to-know issue from the design stage of the AI system,
  - **R9**: prohibit the automated use of AI systems for critical actions on the IS,
  - **R10**: control and secure privileged access of developers and administrators to the AI system,
  - **R11**: host the AI system in trusted environments consistent with security needs,
  - **R12**: partition each phase of the AI system into a dedicated environment,
  - **R13**: implement a secure Internet gateway in the case of an AI system exposed on the Internet,
  - **R14**: favor *SecNumCloud* hosting in the case of a deployment of an AI system in a public Cloud,
  - **R15**: plan for a degraded mode of business services without an AI system,
  - **R16**: dedicate GPU components to the AI system,
  - **R17**: consider side-channel attacks on the AI system.
- **4 recommendations specific to the training phase**
  - **R18**: train an AI model only with data legitimately accessible by users,
  - **R19**: protect the integrity of the AI model training data,
  - **R20**: protect the integrity of the AI system files,
  - **R21**: prohibit retraining the AI model in production.
- **3 recommendations specific to the deployment phase**
  - **R22**: secure the production deployment chain of AI systems,
  - **R23**: plan security audits of AI systems before deployment in production,

  - **R24**: plan functional business tests of AI systems before deployment in production

- **5 recommendations specific to the production phase**
  - **R25**: protect the AI system by filtering user inputs and outputs,
  - **R26**: master and secure the interactions of the AI system with other business applications,
  - **R27**: limit automatic actions from an AI system processing uncontrolled inputs,
  - **R28**: partition the AI system into one or more dedicated technical environments,
  - **R29**: log all processing carried out within the AI system.

There are specific use cases addressed in the document, which lead to new recommendations:

- **3 recommendations for AI-assisted source code generation**
  - **R30:** systematically control the source code generated by AI,
  - **R31:** limit AI source code generation for critical application modules,
  - **R32:** raise awareness among developers about the risks associated with AI-generated source code.
- **1 recommendation in the case of consumer AI services exposed on the Internet**
  - **R33**: tighten security measures for consumer AI services exposed on the Internet.
- **2 recommendations when using third-party generative AI solutions**
  - **R34**: prohibit the use of generative AI tools on the Internet for professional use involving sensitive data,
  - **R35:** perform a regular review of the configuration of generative AI tools' rights on business applications.

## 2.4 Qualitative assessments of attacks

To qualitatively assess attacks targeting AI systems, we will draw on several recognized frameworks in cybersecurity and risk management. Each of these frameworks provides a specific approach to analyzing vulnerabilities and their impact.

- **CyberDico** [4] is an online dictionary offered by ANSSI to provide clear and precise definitions of terms, expressions and acronyms used in the field of cybersecurity. Using this dictionary makes it easier for everyone to understand cybersecurity vocabulary based on definitions compiled by the relevant national authority in this field. This dictionary has a general scope of cybersecurity and is not limited to specific themes such as: development, cloud computing or artificial intelligence. It should be appreciated as a source of

definitions on general cybersecurity concepts. It can be supplemented by additional elements provided by ANSSI such as reports, recommendations, security notices or even by the law in force.

- **ISO/IEC 27000:2018 Standard** provides an overview and vocabulary [15] of information security management systems (ISMS):
  - This standard provides a comprehensive framework for information security management,
  - It defines the main terms and concepts used in the ISO 27000 family of standards,
  - Its high level of abstraction may limit its practical application for evaluating attacks on systems.
- **CVSS Indicator** [19] (*Common Vulnerability Scoring System*)
  - The degree of abstraction of the ISO27000 definition is, in our opinion, too great to be usable or readable as it stands,
  - We therefore decided to use the definition of availability as perceived by the *Forum of Incident Response and Security Teams* (FIRST). This non-profit organization is behind the CVSS indicator,
  - This indicator is a standardized evaluation system commonly used in the cybersecurity market to qualify the characteristics and severity of a vulnerability (applications, systems or others).
- **EBIOS RM Method** (*Expression of Needs and Identification of Objectives Security Risk Manager*) from ANSSI [5]: this is the method favored by ANSSI for assessing and treating cyber risks.
  - It provides a methodology for assessing and managing cyber risks,
  - It allows the threats weighing on a system to be assessed and appropriate remediation measures to be defined,
  - Its structured framework is particularly suited to identifying and prioritizing risks associated with AI systems.

### *2.4.1 Evaluation criteria*

#### *2.4.1.1 AIC principles*

Within an organization, preventing and responding to an attack involves having in place a set of organizational and technical systems that are regularly tested and proven. The prism chosen here is that of ***cybersecurity***, that is to say, as ANSSI would define it in its CyberDico [4], that it is a question of searching for a "*state […] for an information system enabling it to withstand events from cyberspace likely to compromise the availability, integrity or confidentiality of the data stored, processed or transmitted and the related services that these systems offer or make accessible*".

It is therefore a question of ensuring that the "*security needs*" are covered: ***availability***, ***integrity*** and ***confidentiality*** ("***AIC***"). Each of these needs can be

covered through different techniques and processes deployed within a company. The implication for an artificial intelligence system is major, insofar as if one of these security needs were to be compromised by a malicious act, the expected results and operation would be impacted.

These combined elements can affect the ***reliability*** of the artificial intelligence system, regardless of the stage of the system's lifecycle:

- In its ability to infer in accordance with the purpose for which it was developed;
- In its ability to produce reliable results.

Here we describe the constituent elements of the attack description sheets.

#### *2.4.1.2 Attack context and technical facilities*

To understand the threats to which artificial intelligence systems are exposed, it is necessary to ***understand the context*** in which an attack can be implemented. This context is decisive insofar as it puts into perspective the attacker profiles and the means they must have to execute a more or less complex scenario. Indeed, a cyberattack is in essence a malicious act undertaken against an organization regardless of its size or activity. It is an event calling for constant vigilance due to the diversity of the perpetrators and the methods implemented. In its CyberDico [4], ANSSI defines the ***cyberattack*** as follows: "*A cyberattack involves damaging one or more computer systems in order to satisfy malicious interests."*

ANSSI defines a cyberattack by its target and its purposes. The definition can be supplemented by the fact that it is a voluntary act whose author, the *way of working* and motivations can vary (configuration faults or other factors can of course lead to an information leak that is not the result of a voluntary act). In fact, these elements fluctuate depending on whether the attacker is an amateur, a criminal group, an ideologist or a state-funded entity. Determining an attacker's profile makes it possible to assess the ***resources*** at his or her disposal for " *attacking one or more computer systems*" but also and depending on the method of operation and the nature of the attack, to estimate the impacts on an information system.

Furthermore, while it is necessary to know the malicious individual(s) behind an attack, it is also worth considering the conditions they must meet to achieve their objective. A user must have a series of ***knowledge*** inconsistent with those he would have provided, of an ***expertise*** or of ***specific access rights*** necessary to implement the attack. The more resources the attacker has at his disposal, the fewer obstacles or difficulties he will encounter in exploiting the attack scenario.

#### *2.4.1.3 Qualitative evaluation criteria*

Estimation of the different criteria

To provide useful reading keys for understanding attack scenarios and their implications for an information system or an artificial intelligence system, we

propose in the following sections qualitative qualifications of the criteria mentioned previously.

That is to say that, first it is necessary to qualify and at least propose an estimate of the impacts that an attack on the AIS would imply by considering: the measurement of the impact of the attack on the needs of availability, integrity and confidentiality but also on the subsequent reliability of the model. Then, secondly, a set of conditions that it appears necessary to satisfy, at the time of writing this document, to compromise with more or less difficulty an artificial intelligence system.

Adaptation required for the use case

It is important to take a step back from the assessments proposed for each attack sheet covered in this booklet. The assessments proposed are generic, and an attack will not have the same impact depending on the use case provided by the attacked artificial intelligence system, or on the cybersecurity maturity of the targeted organization.

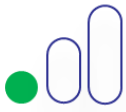
For all practical purposes, it should be noted here that the suggestions and reflections proposed below are intended to be broadly applicable to artificial intelligence systems. That is to say, our attention is not focused specifically on artificial intelligence systems for generative uses, LLMs or artificial intelligence systems for predictive uses. The objective is to have keys for reflection that are resilient to technological developments and intended to be broadly applicable to the cases studied and to subsequent developments in the threat.

### *2.4.2 Impact Indicator (Availability, Integrity, Confidentiality, Reliability)*

Presentation of the impact indicator

The premise of this indicator is to propose an average impact of the attack based on security needs and reliability, i.e. to take an average of the four criteria (availability, integrity, confidentiality and reliability) whose scores are scaled from 1 to 3 depending on the attack scenario. Impact level 1 corresponds to a low impact while impact level 3 corresponds to a high impact.

The scale of impact on AIS safety and reliability requirements is as follows:



**Low (1)**

**Medium (2)**

**High (3)**

The *Impact* of the scenario is established as **Low (1)**, resp. **Medium (2)**, resp. **High (3)** when the average of the criteria leads to the assumption that the attack

generates a low, resp. medium, resp. high impact on the security needs as well as on the reliability of the AIS.

Formula for calculating the indicator value

The formula justifying the level of *Impact* of a scenario on all the criteria[11] is to be designed as follows:

**Impact = (Availability + Integrity + Confidentiality + Reliability) / 4**

Any decimal number obtained from the formula must be rounded up or down following the usual rounding rules:

- If the Impact is greater than (>) or equal to (=) 1.5 or 2.5 then the rounding is upwards:
  - > or = to 1.5 = 2
  - > or = to 2.5 = 3
- If the Impact is less than (<) or equal to (=) 1.4 or 2.4 then the rounding is downwards:
  - < or = to 1.4 = 1
  - < or = to 2.4 = 2

Please note that, for contextualization purposes, this scale should be adapted according to the context of each sheet. The proposals made subsequently do not take into account the strategic choices that certain organizations may adopt in prioritizing one security need over another. For example: the following proposals do not take into account the prioritization that could be made of the need for confidentiality for organizations subject to a given legal framework.

Example: an attack involving strong impacts on reliability and availability would see its level of *Impact* defined as follows:

Impact = (Availability (3 - High) + Integrity (1 - Low) + Confidentiality (1 - Low) + Reliability (3 - High)) / 4

Impact = (3 + 1 + 1 + 3) / 4 = 8/4

Impact = 2

The *Impact* of the attack scenario is estimated to be

- Average due to the absence of any breach of the need for integrity and confidentiality.
- Raised on the need for availability of the system and its services as well as on the reliability of its inference capacity.

[11] This criterion will be taken into account in this calculation if it is not assessed as N/A (i.e. if it has an estimated level between 1 and 3 inclusive)

Finally, it is worth considering the hypothesis in which assessing the impact on a security need does not appear applicable or possible. This is, for example, the case of a model extraction which, a priori, has no impact on availability, integrity or reliability.

Such cases are classified as "N/A", not applicable and do not fall within the proposed calculation formulas. When a criterion is considered "N/A", it is grayed out on the sheet, because it is deemed that the assessment of an impact is not possible or is not relevant due to the nature of the attack.

#### *2.4.2.1 The availability criterion*

Under ISO27000:2018 [15], the **availability** in terms of information security management system is defined as: "*the property of being accessible and usable on demand by an authorized entity*". This means that an attack on availability can, for example, qualify the impossibility of accessing the services of a model, of generating results or of ensuring its administration or training.

The objective here is to qualify the impact on accessibility and the possibility of using an AIS that is the subject of an attack. The degree of abstraction of the ISO27000 definition is, in our opinion, too high to be usable or readable as it is. Therefore, we decided to use the definition of availability as perceived by the CVSS indicator [19]. Based on this observation, it seemed relevant to us to summarize the essentials through the following three levels of impact inspired by the CVSS index described previously:

- **Low (1):** exploitation of the scenario **does not appear to affect the availability** of the artificial intelligence system;
- **Medium (2):** exploitation of the scenario **appears to affect the availability of** the system or its services **for a short period**;
- **High (3):** operation may affect the availability of the system or its services **for an extended period**.

The risk appetite for the interruption of an AIS's services appears to be a subjective criterion and dependent on the context of the organization, therefore no specific proposal for the duration of interruption has been proposed. The criterion proposed in this booklet is intended to be qualitative.

#### *2.4.2.2 The integrity criterion*

Under ISO27000:2018 [15], **Integrity** in terms of information security management system is defined as: the "*property of accuracy and completeness*". ANSSI, in its CyberDico [4], also formulates it as: "*Guarantee that the system and the information processed are only modified by a voluntary and legitimate action*".

An attack on integrity therefore describes the compromise of data entering or leaving a system, resulting in distorted results or results diverted from the initial destination.

In other words, it is a matter of data preserving its characteristics throughout the processing phase. The integrity criterion measures the extent of data alteration or destruction, and therefore the difficulty of investigating to repair the model and/or its services.

The stakes are high since it is a question of ensuring the legitimacy of the results produced and, therefore, the reliability of the entire system and its algorithms. The implications for the models, particularly in the training phases, are that, for example, the input data can be altered by an external action (e.g. in the case of data poisoning) and affect the result. In the same way as for the availability criterion, we have decided to use the definition of integrity given by the CVSS indicator [19].

The subtlety of the integrity criterion presented in this document is that it includes in certain aspects the expectations of traceability. Indeed, given the complexity of tracing and explaining the actions taken by certain models (particularly for LLMs), we start from the assumption that a compromised data set would impact the ability to go back to determine the causes of the breach of integrity.

So, if a data set has its integrity impacted, then:

- The data is potentially distorted,
- The ability to trace the history of malicious actions,
- and/or data repair is made more complex,
- and therefore, the attack on integrity is greater.

The goal is to keep in mind that the data, the fuel used by the model to produce its results, determines the reliability of the AIS that processes it. It goes without saying that this reasoning also applies to the model itself. If a model's characteristics are modified by the exercise of administrative rights for malicious purposes, the traces, the reconstruction of the model, or the readability of the actions would likely be rendered illegible.

Finally, it should be noted that all of these traces are useful for auditing the system and tracing the path taken by the malicious user. We therefore propose the following three levels of impact in summary:

- **Low (1):** exploitation of the scenario appears to cause virtually **no impact** on the integrity of the data processed by the artificial intelligence system or its services. It is possible to **easily reconstruct** the data and/or repair the model. The **history of user actions is readable and/or accessible**.
- **Medium (2):** exploitation of the scenario could lead to the modification of data with **low impact on the operation and/or on the results produced** by the AIS or its services. **Repairing the model and its services may involve difficulties**. Investigations conducted to determine the actions taken by the user **may be obstructed**.

- **High (3):** exploiting the scenario allows the malicious user to modify data with a **high impact on the operation and/or results** produced by the AIS or its services. **Repairing** the model and its services, and/or **investigations** conducted to determine the actions taken by the user **are made very difficult or even impossible**.

The security need for integrity is also subject to interpretation and is conditioned by the specific needs of an organization. Therefore, the user is encouraged to place the proposed scale in their specific context. Indeed, this criterion is subjective and does not represent the needs of all sectors.

#### *2.4.2.3 The confidentiality criterion*

Under ISO27000:2018 [15], **confidentiality** in terms of information security management system is defined as: "*property according to which information is not disseminated or disclosed to unauthorized persons, entities or processes*". The definition of confidentiality is quite graphic in that it qualifies the need to ensure that only authorized people have access to information. Thus, the breach of the need for confidentiality implies a disclosure or sharing of sensitive information regulated by law (for example: the GDPR[12] for personal data, the applicable positive law on intellectual property for patents) or subject to a particular classification within an organization.

The implications for an AIS are severalfold and depend on the intended use and the information it is required to communicate to its users. Indeed, the degree of impact on confidentiality will be high for an organization whose strategic data is disclosed through the model, as much as the company whose personal data of its customers would leak through the compromise of the AIS. Conversely, an organization that does not feed its model and that operates it only from publicly accessible data will suffer a lesser impact on its need for confidentiality.

The consequences of these disclosures and data leaks are of several kinds since they can involve subsequent impacts, whether strategic, legal or image related. To continue in the same dynamic as the two previous criteria, we based ourselves on the definition proposed by the CVSS indicator [19]. The synthesis of these definitions is presented in three levels as follows:

- **Low (1):** exploitation of the scenario **does not appear to impact data confidentiality**,
- **Medium (2):** exploitation of the scenario **may lead to the disclosure of confidential information with low impact, strategic, legal and/or image**,
- **High (3):** exploitation of the scenario may result in the **disclosure of confidential information with strong strategic, legal and/or image impact**.

[12] General Data Protection Regulation (GDPR). More information is given on the ANSSI CyberDico [4]

Like the two previous criteria, the need for security, such as confidentiality, must be contextualized. A construction company will not be subject to the same regulatory constraints as a banking institution. However, both are subject to the General Data Protection Regulation (GDPR).

#### *2.4.2.4 The reliability criterion*

As for the criterion of *Reliability*, it is not based on any standardized definition by the international ISO organization or by ANSSI. It is a proposal aimed at establishing a purely qualitative criterion of what an attack on the reliability of the results could imply (we recall that in the absence of any attack, an AIS can provide false answers (the hallucinations of generative AI), although unlikely) and on the satisfaction of expectations. The objective is to provide the operational counterpart of the use of the model and its capacity to do what it was developed for. That is to say, to evaluate to what extent the inference capacity and the results of the model are affected. We therefore propose, in three levels, the impacts that an attack on the reliability of an artificial intelligence system could have:

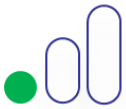
- **Low (1):** exploitation of the attack scenario does not divert the system from its purpose and **its results are not influenced**,
- **Medium (2):** exploitation of the attack scenario partially affects the inference capabilities of the system and its services by diverting them from their purpose. **The results are partially erroneous or unexpected**,
- **High (3):** exploiting this attack scenario affects the inference capabilities of the system and its services in such a way that they are diverted from their purpose. **The results contain erroneous, unexpected, and/or illegal** content. Such a scenario implies a distrust in the reliability of the system, its services, and the entirety of its results.

### *2.4.3 Technical Ease Indicator (Time Spent, Resources, Expertise, Knowledge, Access)*

Presentation of the impact indicator

Through this indicator, we decided to use qualitative criteria to rate the means necessary to implement the scenario. The added value provided by this approach is, in our opinion, that it provides details on the typology of perpetrators of malicious acts, on the knowledge of the context, the determination, and/or the means they must have to achieve their objective. The approach described below is purely pragmatic and is based only on proposed scales which cannot replace an in-depth study of the state of the threat and the context of the targeted organization.

The proposal of the *Technical ease* indicator is based on an average of five criteria rated from 1 to 3. The higher the criterion rating, the easier it will be to implement the attack scenario studied. The scale of the indicator *Technical ease* implementation of the attack scenario studied materializes as follows:



The Technical Facility is **Low (1)**, resp. **Medium(2)**, resp. **High (3)** if we consider that the attack scenario is difficult to implement, or moderately simple to execute, or simple or with reduced constraints to implement. The average of the criteria below leads to the assumption that a malicious actor must have: significant (or minimal, or very limited) knowledge of the system, a time frame and significant (or limited, or very limited) means to exploit the scenario presented.

Formula to calculate the indicator value

The formula justifying the level of *Technical ease* of a scenario is substantially similar to that used for *Impact* in that it constitutes an average of the criteria studied as follows:

**Technical ease = (Time spent + Resources + Expertise + Knowledge + Access) / 5**

Any decimal number obtained from the formula must be rounded up or down using the following rules along with the rounding rules defined previously.

For contextualization purposes, this scale should be adapted to the specific context of each sheet. The proposed formula leads to a priori estimate of the level of complexity of an attack. Such an analysis is purely subjective and therefore requires recontextualizing the proposed scales to the reality of the reader's organization. The proposals presented in this booklet cannot replace an in-depth study of the threat situation and an analysis of an organization's risk appetite.

Example: for the case of modifying the retraining data of a model, such as a publicly accessible chatbot, in order to introduce a deviation in its behavior. This case may see its *Technical ease* determined as follows:

Technical ease = (Time spent + Resources + Expertise + Knowledge +Access) / 5

Technical ease = (3 + 3 + 3 + 3 + 3) / 5

Technical ease = 3

The *Technical ease* of this scenario is estimated as high due to a short implementation time, the absence of the need to be an expert on the subject or to know the system and its services, this with simply public access to the chatbot and without any particular organization.

#### *2.4.3.1 The criterion of Time spent*

The criterion of the **Time spent** aims to qualify the time required for the malicious user to implement the scenario. The purpose of such a criterion is to propose a range of time required for an attacker to achieve his objective. This proposed scale

is a particularly fragile criterion in that it is likely to evolve as the use of AIS s becomes more widespread. Indeed, an attack that required a day of implementation or preparation three years ago may no longer require as much time today.

In the same way, this criterion is subjective, the needs of each organization being variable from one sector of activity to another, this scale will certainly need to be adapted by the reader. An organization can have coordination and technical measures of robustness that can justify seeing the implementation time upwards, in the same way depending on the exposure of the model the time spent can be seen downwards. We propose three levels of implementation time ranges:

- **Long (1):** exploitation of the attack scenario studied seems to require **a long preparation and its execution can take several weeks to several months**,
- **Moderate (2):** exploitation of the attack scenario studied seems to require **preparation time and its execution can take from several days to a week**,
- **Court (3):** exploitation of the attack scenario studied **does not seem to require any preparation** and its implementation **only takes from a few hours to a day**.

#### *2.4.3.2 The Resources Criterion*

The criterion of **Resources** necessary is inspired by the notion of "*source of risk*" (attacker profiles) of the EBIOS Risk Manager method [5]. This is a proposal for measuring the level of motivation and organization that a malicious user or group must have to implement the attack scenario.

The more human and material resources a group has, the more likely it is to be motivated to compromise a system. In cybersecurity, the source of risk can be of several kinds:

- From an amateur who has no means other than his workstation,
- Through the criminal group acting for financial gain,
- Or even the most prepared organizations structured and financed by States for the purposes of political destabilization.

The diversity of profiles is significant and is left to the discretion of organizations to appropriate this scale of Resources necessary to implement an attack. Indeed, it will probably be more relevant for a VSE/SME to be wary of internal malicious acts, organized criminal groups than of an organized group financed by a State to which it would not be exposed a priori.

It should be kept in mind, however, that: "*he who can do more can do less*", organized groups can carry out attacks of a certain technical and implementation simplicity. Thus, the scale of resources required is intended to be flexible and general and must be adapted to the context of the organization. Here is the breakdown below:

- **High (1):** implementation of the scenario studied **requires considerable material, human and financial capacities.** This scenario is particularly likely to be exploited by state groups or intelligence agencies characterized by their ability to carry out particularly sophisticated offensive operations over a long period of time,
- **Average (2):** implementation of the scenario studied **requires human, financial and material resources.** This scenario is particularly likely to be exploited by organized groups (terrorists, criminals or ideologists) capable of conducting more or less sophisticated operations,
- **Weak (3):** implementation of the scenario studied **does not require any particular financial or material resources.** This scenario is particularly likely to be exploited by amateurs or smaller activist groups.

#### *2.4.3.3 The criterion of Expertise*

The criterion of **Expertise** aims to provide a contextualization of attacks on artificial intelligence systems at a time when their compromise is not yet on a systematic and widespread scale. The aim here is to consider, while remaining humble, that the public and de facto the attackers are not yet all familiar with the functioning of AI and their services. Therefore, we propose a scale of knowledge and technical understanding of the environment inherent to the characteristics of artificial intelligence systems.

This criterion can be considered by balancing knowledge in cybersecurity and data science. We invite the reader to take this scale and assess their organization's situation with regard to this topic. Indeed, a model whose technical understanding requires only ten hours of training does not require the same level of attention as an LLM whose parameters are administered by experts in the discipline. We therefore propose the following scale:

- **High (1):** implementation of the attack scenario studied **requires very advanced or specific technical skills and/or the development of targeted tools**,
- **Average (2):** implementation of the attack scenario studied **requires the implementation of simple techniques and/or publicly available tools**,
- **Low (3)**: implementation of the attack scenario studied **does not appear to require any specific technical skills or particular tools**.

#### *2.4.3.4 The criterion of knowledge about the system*

Unlike the previous criteria, the criterion of **Knowledge** focuses on the context of the artificial intelligence system itself. That is, the organizational and technical context in which it is situated. In other words, it is a matter of evaluating to what extent specific knowledge of the system and its services in its environment is necessary to be able to implement the attack scenario under consideration.

The purpose here is to put into perspective the greater complexity of implementing the scenario on a complex model in an equally rich environment. Where a widely used model might no longer hold any secrets for the market and at the same time, for malicious users.

Once again, this scale must be put into context, since it is entirely possible to use a widely democratized model by following specific security recommendations to strengthen its parameters. Knowledge of the system is therefore no longer sufficient and its environment plays just as much a role. To materialize this analysis, we propose the following scale:

- **High (1):** the attack scenario studied is more difficult to exploit since the attacker **must have complete knowledge of the integration** of the model in the artificial intelligence system and its environment,
- **Average (2):** the attack scenario studied is exploitable subject to certain constraints insofar as the attacker **must have some knowledge of the information system** in which the artificial intelligence system is located. It is **necessary to have either knowledge of the context** in which it is located, **or other elements with which it would be interfaced**, or knowledge of the technical characteristics of the model,
- **Low (3):** the attack scenario studied is simpler in its implementation since the attacker **does not need to have specific knowledge of the object model** of the attack or its environment.

#### *2.4.3.5 The Access criterion*

Finally, the criterion of **Access** is a pragmatic proposal to qualify the need to have accounts with varying levels of privileges in order to use, produce outputs, administer or modify the model's parameters for malicious purposes.

The scenario will therefore become more easily achievable if simple user access is required to access the models and these functionalities. The ease of implementation will also be more evident if this same user account has access to administrative functions normally limited to certain profiles.

Similarly, if a publicly accessible account can produce actions that have consequences on the functioning of the model or its services, this can make the attack scenario even easier to implement (for example, in the case of prompt injection).

Conversely, a model whose access is strictly segmented by user profile with a dedicated rights nomenclature, and a limited number of administrators will increase the complexity of implementing the attack scenario.

The proposed scale must be adapted to the context in which the model in question is used, depending on whether it is freely accessible to the public or requires the

creation of an account as the repercussions and security requirements will not be the same.

The scale of the necessary access criteria is as follows:

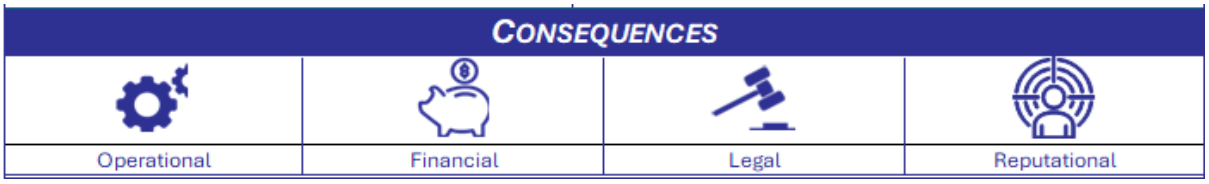
- **High Privilege Internal User (1):** implementation of the attack scenario studied **requires elevated rights**, such as administrative rights,
- **Internal User (2):** implementation of the attack scenario **requires being a internal user** and authenticated by the organization,
- **General public (3):** implementation of the attack scenario **does not require any specific access rights** (for example: if the artificial intelligence system is accessible to the public).

### *2.4.4 The consequences of an attack on the organization*

Before continuing, it should be noted that the previous elements were intended to qualify the more or less direct impacts of an attack on an artificial intelligence system. These proposals therefore focus on a specific security topic at the organizational level, in this case: the security of AI systems and the impacts of attacks on their components and services. The events in question will therefore most often be classified as "operational impacts."

But not every attack on an information system or one of its components has the sole consequence of disrupting operations. On the contrary, an attack can have collateral impacts on a more strategic scale. This is particularly true if corporate secrets are exposed, the confidentiality or integrity of personal customer or employee data is compromised, or if the event has consequences on the financial results of a business.

To ensure that these strategic aspects that may result from a compromise of an AI system are not overlooked, we propose a section that succinctly identifies the consequences of an attack on an AI model for an organization. Like the EBIOS RM method, which proposes impact categories, we briefly propose four categories of strategic consequences for an organization.




## 2.5 Taxonomy of attacks

To facilitate the understanding and management of security risks related to artificial intelligence (AI) systems, we have developed an attack taxonomy. This taxonomy aims to provide a structured and comprehensive framework for identifying, classifying, and analyzing the various threats to these systems. The taxonomy is based on the frameworks we described above:

- NIST.AI.100-2e2023 [7],
- MITRE ATLAS [17],
- OWASP Top 10 LLMs [10],
- OWASP Top 10 ML [11].

As we have seen previously, the different repositories provide different information, which we felt would be useful to group together in a single taxonomy. We have also noted that the very rapid evolution of AI technologies is constantly bringing new potential attacks to light. This is why we will undoubtedly have to update this taxonomy as new AI systems (for example, agentics) arrive.

The taxonomy is organized into four hierarchical levels, providing a granular and practical approach to understanding attacks:

1. **Lifecycle phases:** this first level uses the lifecycle of an AI project as the main axis of classification. We have chosen the ANSSI model [1] (above) and then the OECD[5] model, which breaks down the development of an AI system into seven distinct phases previously detailed (see section 2.1.3). This choice makes it possible to associate each attack with a specific phase of the lifecycle, thus facilitating the identification of relevant risks at each stage of a project. For a more holistic view, we have also integrated the three phases of the ANSSI lifecycle (Training, Deployment and Production), by superimposing them on the OECD model, as we described previously,
2. **Family of attacks:** the second level groups gather attacks that share common characteristics, such as similar attack mechanisms, common objectives, or comparable impacts. Examples of attack families include data poisoning, evasion, pattern extraction, etc. This grouping allows for a better understanding of the different threat categories and the development of more general defense strategies,
3. **Specific attacks:** the third level describes each attack in detail. Each attack is documented with a detailed description of how it works, its potential consequences, detection techniques, and mitigation measures. This level of detail provides AI and cybersecurity experts with the information needed to understand and counter specific threats.

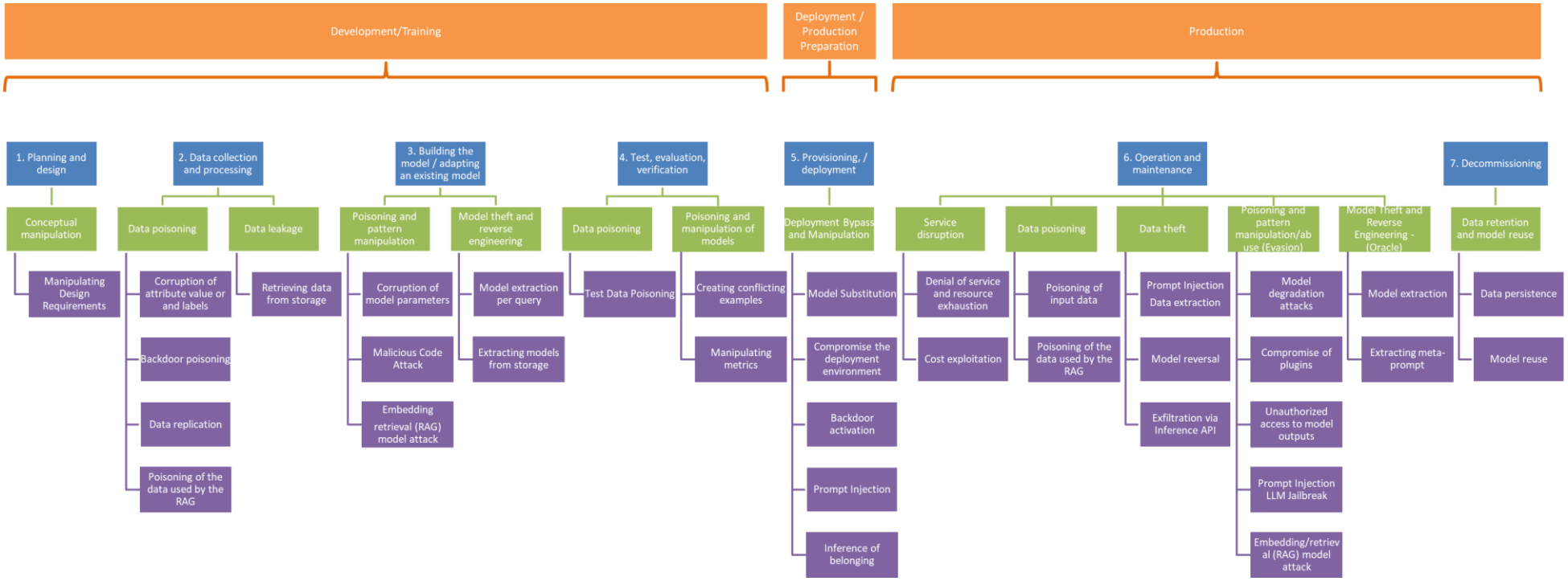



*Figure 11 – Taxonomy of attacks on AI*

## 2.6 Main categories of attacks

Here we present a simplified typology of attacks, focusing on the categories of poisoning, evasion, and oracle.

### *2.6.1 Poisoning Attacks*

These attacks target the model training phase, altering the training data or the model to compromise its integrity:

- **Data poisoning**: introduction of malicious data into the training set.

Analogy: corrupting a textbook so students learn the wrong answers.

Example: injecting fraudulent transactions into the reference data of a fraud detection model.

- **Model poisoning** (more specific to distributed and collaborative models): direct modification of model parameters during training.

Analogy: modifying the source code of a program to make it behave differently.

Example: a malicious participant in a federated training (see section 3.3) sends corrupted model updates (it transmits wrong parameters).

- **Supply chain attacks**: compromise of model components before use.

Analogy: receiving spyware hidden in a legitimate program.

Example: using a compromised software library or pre-trained model containing a back door.

### *2.6.2 Evasion Attacks (Evasion)*

These attacks target the model in production, modifying the input data to avoid being identified as a threat and to obtain erroneous predictions unnoticed:

- **Classic Evasion**: perturbation of input data to induce incorrect classification.

Analogy: slightly modify an image so that it is poorly recognized.

Example: modifying an image of a stop sign to fool a self-driving car.

- **Prompt injection** (LLM specific): manipulation of the LLM text interface to bypass restrictions and obtain unwanted responses.

Analogy: asking trick questions to a voice assistant.

Example: asking a chatbot to generate malicious content.

### *2.6.3 Oracle Attacks*

These attacks exploit access to the model to extract information or influence its behavior:

- **Inference attacks**: infer information about the training data or the model from its predictions.

Analogy: guessing exam questions by analyzing the answers.

Examples: Membership inference (determining whether a data item was present in the training set) or pattern extraction (reproducing a competing pattern).

- **Data extraction attacks** (more critical for LLMs): obtaining sensitive information from the model, often via carefully constructed prompts.

Example: extracting credit card numbers stored by a chatbot.

- **Excessive consumption of resources** (more critical for LLMs): overloading the model with requests to degrade service or exhaust resources.

Analogy: overloading a web server with requests to make it inaccessible.

#### *2.6.4 In conclusion*

This simplified classification highlights the main categories of attacks against AI systems. As an AI expert or project manager, understanding these threats is crucial to developing robust and secure models. The various attacks will be detailed in the following sections.

# 3 Other techniques to follow

We describe here some techniques that can bring new types of defenses (such as encryption in 3.4 Cryptography) or attacks, some of which are included in our taxonomy (3.1 RAG and 3.5 Adversarial Attacks) and others not yet (3.2 Agentic, 3.3 Federated Learning).

## 3.1 RAG

**Limitations that we want to face**

Generative artificial intelligence (AI) excels at creating text responses based on large language models, where the AI is trained on a large amount of data. The good news is that the generated text is often easy to read and provides detailed responses.

The bad news is that the information used to generate the response is limited to the information used to train the AI, often an LLM. The LLM data may be weeks, months, or years out of date, with no easy way to update it.

Additionally, in an enterprise AI chatbot, they may not consider information specific to the company's products or services.

This can lead to incorrect responses that erode some customers' and employees' trust in that technology.

### Birth of the RAG - Retrieval Augmented Generation

Corpus: the first step is to gather the targeted information and the additional data resources that we want to make available to the LLM included in our AI system. They form our documentary corpus or knowledge base.

This data is then processed in order to become usable by our RAG, through the following steps:

### How RAG works

Corpus: the first step is to gather the targeted information, the additional data resources that we want to make available to the LLM included in our AI system. They form our documentary corpus or knowledge base.

This data is then processed in order to become usable by our RAG:

- Chunking (*chunks*): the documents in the corpus are divided into short passages.

  Some of these passages will be provided as input to the LLM to assist in generating an appropriate response (this is the *context* of the prompt*).* They cannot be too large since the inputs provided to LLMs cannot exceed a certain amount, determined by the context of an LLM.

  A LLM's context window can be thought of as the equivalent of their working memory. It determines how long a conversation they can carry on without forgetting the details of the previous exchange. It also determines the maximum size of documents they can process at one time.

- Digital representation (*embeddings*): the semantic content of each passage is converted into vector form.

  This digital representation allows the meaning of words to be preserved, since, for example, words with a similar meaning will be transformed into vectors with common characteristics, having a low vector distance.

- Vector base (*vector store*): these semantic vectors are stored in a database specially designed for vector calculations, which will be queried in addition to the user prompt.

  When a user queries the AI, the RAG comes into play to provide the AI service with additional information that will allow the underlying LLM to respond based on information from the document corpus:

- Digital representation (*embeddings*): the user's question is converted into semantic vectors, using the same method as previously used to create the vector base of the corpus.
- Similarity search: the search module uses similarity measures to compare the question vectors to the document vectors in the database. The vectors

corresponding to the passages "closest" to the question are selected for answer generation.

Once selected, these vectors are converted back into natural text, i.e., the corresponding passages of documents from the initial corpus.

- LLM: the LLM uses the question, and the extracts retrieved by the previous search to generate a relevant answer.

The diagram below illustrates all these steps:

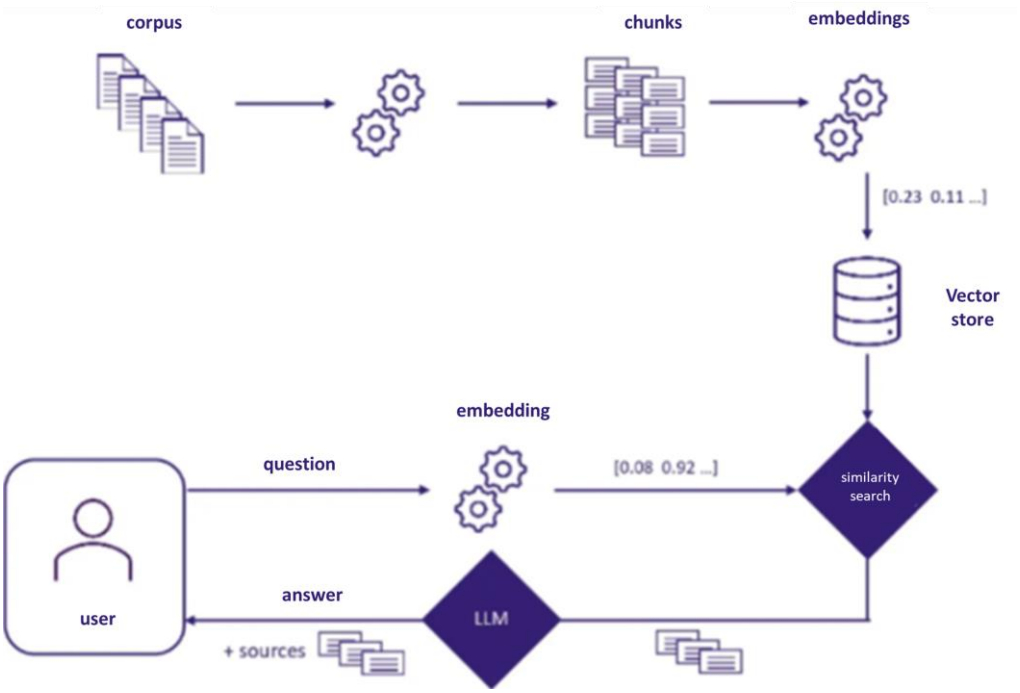



*Figure 12 – Operation of the RAG*

**Benefits of using RAG**

- While the LLM training process is long and expensive, it is the opposite for RAG updates. New data can be loaded and translated into vectors continuously and incrementally.
- RAG also has the advantage of using a vector database, which allows the AI service to provide the specific source of the data cited in its response, something LLMs cannot do. Therefore, if there is an inaccuracy in the AI output, the document containing this erroneous information can be quickly identified and corrected, and then the corrected information can be entered into the vector database.

**Specific Attacks on RAG**

- RAG systems often access large databases, raising concerns about data security and privacy. Protecting sensitive information while maintaining system functionality is crucial, requiring a delicate balance.

- Similarly, every manipulation of this data is a potential entry point for attackers: the digital representation, the search in the vector database, the transmission of the selected data to the LLM, and finally the interpretation of the selected data.
- The LLM model included in the AI service is vulnerable to classic attacks on AI systems.

## 3.2 Agentic systems

### Benefits of Agentic systems

Agentic systems represent a significant development in the field of artificial intelligence; unlike traditional language models, which generate responses based on a specific query, agentic systems make decisions autonomously and actively interact with their environment.

### Definition

An autonomous **agent** is capable, as shown in Figure 13, to:

- Interact with your environment.
- Make independent decisions.

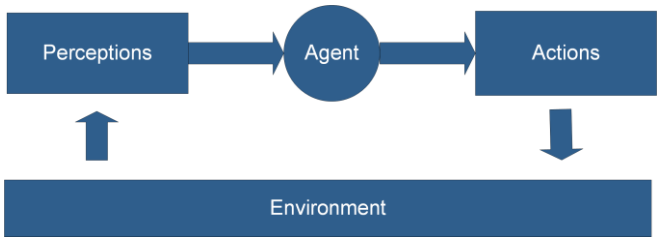



*Figure 13 - Schematic diagram of an agent*

An **agentic system** is an artificial intelligence architecture composed of one or more agents capable of interacting / collaborating with other entities (humans, agents) to achieve complex objectives. These agents are designed to operate with a certain degree of independence, allowing them to dynamically adapt to changes in their environment and continuously optimize their decision-making.

### Key Features

- **Autonomy and decision-making**: Agents in an agentic system can act independently, relying on their perception of the environment. Unlike reactive AIs, they do not require constant supervision and can initiate actions based on the situations encountered.
- **Interconnection and collaboration**: agents are able to communicate and exchange information with other agents or systems and their environment. This ability to learn and evolve allows them to adapt their behavior to dynamic and unpredictable contexts. These emergent behaviors can be unforeseen and more complex than the individual behaviors of agents.

- **Distributed architecture**: rather than relying on a single powerful agent, agent-based systems can adopt a distributed approach, distributing skills and responsibilities among multiple entities. The issue of orchestrating the different agents is very important and difficult to resolve.

**Functioning**

An agentic system can consist of several essential components:

1. **Objective and Planning**: the agent receives an overall objective, which breaks down into subtasks to achieve the expected result. It can adjust its plans according to the events it encounters.
2. **Memory and learning**: an agent can retain information about its past interactions, either temporarily (contextual memory) or over the long term (persistent storage). This memory allows it to adapt its behavior and optimize its actions over time.
3. **Connection with other systems**: Agents can connect to APIs, query databases, and interact with other systems. These interactions allow them to access real-time information and make more informed decisions.
4. **Feedback and improvement loop**: an agent analyzes the impact of its actions and adjusts its behavior to optimize its future decisions. This feedback learning mechanism improves the agent's performance, but can also open security holes if an attacker manipulates the learning data.

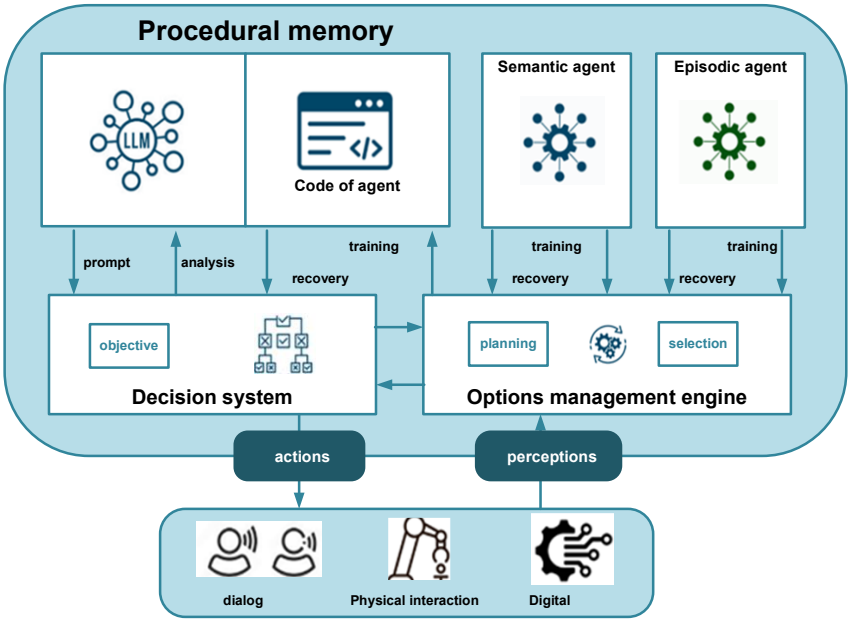



*Figure 14 – Example of an agentic system*

**Link with RAG:** some AI agents integrate RAG to access precise and up-to-date information in real time from a documentary corpus, thus improving their ability to provide relevant and up-to-date responses.

**Applications:** Agentic systems find applications in various fields, including:

- Supply chain management and logistics optimization: dynamic planning and anticipation of stock shortages.
- Personalized health assistance: medical agents for patient monitoring and diagnostic analysis.
- Software development and project management: automation of repetitive tasks and intelligent coordination of teams.
- Financial analysis and decision-making: identification of market trends and automatic execution of trading orders.
- Scientific research and innovation: autonomous exploration of databases and generation of new hypotheses.
- Note that we also find many agentic systems in cooperative robotics and in video games.

**Multi-agentic systems**

Multi-agentic systems (MAS) are composed of several AI agents working together in the same environment with different objectives. Each agent is specialized in a realm and will collaborate with other agents to fulfill a common final task. For instance, in a warehouse, one agent focuses on the workforce dispatch for the preparation of orders, another one monitors the warehouse storage, while the last one analyzes the flow. Together, they aim at optimizing the process of order management.

**What are the attacks specific to agentic systems?**

Agentic systems expose a new attack surface because they combine autonomous decision-making and interactions with other systems. They must be robust against individual agent failures. If an agent fails or behaves unexpectedly, it can disrupt the entire system. Typically, the behavior of agents and their interactions is modeled. If an attack targets the management of the behavior and complexity of the agentic system, this can compromise the robustness of the model. The recently published OWASP papers specifically address attacks on agentic systems [8] and multi-agentic systems[13].

- **Privilege compromise in a MAS**: an agent that accesses external services can be manipulated to gain higher privileges and access critical resources. An attacker can indeed exploit configurations errors and privilege inheritance mechanisms between agents (e.g. implicit delegations), to elevate an agent's privileges or to take over excessive permissions (e.g. sabotage the multi-agent system).
- **Overwhelming Human-in-the-Loop**: an attacker exploits the human supervision system of multi-agent systems (MAS) or an autonomous agent by generating a high volume of queries or alerts to cause decision fatigue, pushing

[13] OWASP. Multi-Agentic system Threat Modeling Guide v1.0. April 23, 2025. https://genai.owasp.org/resource/multi-agentic-system-threat-modeling-guide-v1-0/

supervisors to automatically accept queries and paving the way for dangerous or unwanted actions.

- **Rogue agent infiltration in a MAS**: If an attacker modifies an agent's parameters or manipulates its reward and learning system, they can divert the agent from its initial mission, sometimes with the aim of introducing conflicting objectives. This type of attack relies on gradual changes, difficult to detect immediately, which gradually alter the agent's behavior. Introduced in the multi-agent system, the rogue agent can spread false information to other agents and can thus manipulate them, leading to MAS to unsanctioned or unaligned actions.

## 3.3 Federated learning

**The goal of federated learning** is to allow multiple clients (e.g. individuals, institutes or companies) to train a model collaboratively but without ever sharing their data: on the contrary, clients will only share the model.

This process is coordinated by a central server (e.g., a service provider) and requires several learning rounds to complete the model training. Thus, at each round, the server transmits the current model to the clients, who will then update its parameters by training it independently using their own data. Only the locally updated model parameters are returned to the server, which will then aggregate them by performing a weighted average (by the size of the local datasets) and thus update the federated model.

**Benefits of federated learning:** compared to traditional centralized learning, which involves collecting as much training data as possible and then processing it in a data center, federated learning both reduces bandwidth requirements and improves data confidentiality. Federated learning is therefore of real interest for applications with sensitive customer data or data that is too large to be centralized.

**Attacks specific to federated learning:** however, by opening its learning phase to many actors, this process will facilitate the implementation of attacks related to the integrity of the model and/or expose itself to new attacks targeting the confidentiality of customer data. Next, we will describe the attacks that occur during the federated model's learning phase, but it is important to keep in mind that the federated model, once learned, will be exposed to the same risks of attacks as a centrally built model during its deployment and production phases.

**Integrity attacks:** unlike centralized learning where data can be inspected, the orchestrator of federated learning has no way of verifying that the parameters transmitted by a client correspond to legitimate learning. It is therefore very easy for a malicious client to poison its data or the federated model in order to degrade its performance or behavior. Data poisoning will be carried out either as in a centralized way via a backdoor or by modifying the attributes/tags of the local database.

**Privacy Attacks:** by design, federated learning protects locally stored customer data by aggregating model updates rather than raw data. This is a solution for data privacy but not for model privacy. And even though the model parameters contain much less information about customer data than the raw data, it is still possible to infer information about customer data.

## 3.4 Security of AI systems through cryptography

Artificial intelligence systems have a very broad attack surface. They present the same risks as any computer application, but also specific risks related to AI, such as content injection attacks. An AI system handles large amounts of data, which are found in various forms: in models, user queries, and sometimes in knowledge databases, often in vector form. The data is poorly structured, or not at all, which increases its potential for information leaks.

External attackers seeking to recover data can be of different natures:

- Third parties outside the system who steal stored data (models, database data) or during their transit;
- Infrastructure operators (e.g., a hosting provider) who conduct active attacks on data in transit or in memory;
- Malicious users who exploit vulnerabilities in authorization systems to access data to which they do not have access.

Encryption is the most effective way to protect data against external attacks. Its implementation depends on the system's usage context and the nature of the attackers against whom protection is needed.

When the system is operated *on-premises*, in a controlled environment, the ***encryption of data at rest*** is sufficient. Correctly implemented, with keys hosted in systems external to the AI system, such as KMS/HSM (see glossary), it protects against disk or backup theft. Disk encryption has the advantage of being extremely simple to activate and not affecting system performance. We can therefore recommend testing the performance of the AI system as soon as encryption is activated.

In order to protect a system running in the cloud against active attacks on memory, machine and network, the use of ***confidential computing machines*** is the industry solution currently being adopted. This technology uses CPUs and GPUs that host a secret in their silicon to encrypt and decrypt memory, and limits performance penalties to around 5% for confidential virtual machines. Disk encryption with secrets hosted in the TPM (*Trusted Platform Module*, or vTPM (*Virtual Trusted Platform Module* (if applicable) strengthens this protection, ensuring that an attacker only sees encrypted data in memory and on disk. In addition, user interactions are performed over TLS connections (*Transport Layer Security*) ending

in the machine's encrypted memory, ensuring end-to-end encryption that protects network interactions.

The integrity of the system is ensured by the ***verifiability***, which involves collecting cryptographic fingerprints of the machine's hardware, operating system, software, and models. These fingerprints can be verified externally at any time to assure the user that their system has not been tampered with.

Security can be further strengthened, in ***client-side encryption*** models and data before sending them to the cloud. In use, they will be decrypted, but within the encrypted memory of the machine. Client-side encryption guarantees greater control over the keys and cryptographic algorithms chosen, opening the possibility of more sophisticated encryption such as Covercrypt[14], which is post-quantum and allows access control in encrypted data.

Of the ***purely cryptographic systems***, which do not involve specialized confidential hardware, are currently being developed. As they are exclusively software, their attack surface is reduced and their deployment more universal. In the very short term, fully encrypted vector databases will emerge. In the medium term, fully homomorphic encryption[15]will allow calculations to be carried out directly on the figures.

However, it is recommended to systematically carry out ***performance tests*** as soon as countermeasures with encryption are put in place.

The synthesis of all these previous elements is therefore the following:

| **Context** | **Solution** | **Impact performance** | **Example of technology** |
|---|---|---|---|
| • on-premises<br>• controlled environment | Data encryption at rest | No impact | |
| | Keys hosted in external systems | | KMS / HSM |
| **Cloud**, server side | Confidential computing machines | 5% penalty for confidential virtual machines | TPM or vTPM |
| | Connections TLS | | |
| | Verifiability | | |
| **Cloud** with client-side security | Previous cloud solutions | Previous cloud impacts | Previous cloud technologies |
| | Encryption during transfer to the cloud | Minimal impact | |
| Context independence | Purely cryptographic systems | | • Technologies under |

[14] https://eprint.iacr.org/2023/836
[15] https://fr.wikipedia.org/wiki/Chiffrement_homomorphe

| | | | |
|---|---|---|---|
| | | | development: Encrypted vector bases<br>• Homomorphic encryption |

### *3.4.1 Cryptographic techniques*

Authenticated encryption: encryption ensures the confidentiality of data, but also its integrity (authenticity). A modification of encrypted data by an attacker, or the failure to provide additional authentication data, will cause an error during encryption. The most widely used standardized authenticated encryption is AES GCM[16] (GCM - *Galois Counter Mode* which provides the authentication tag). The size of an AES GCM ciphertext is equal to the size of the plaintext + 28 bytes (12 for the nonce[17], 16 for the tag). AES XTS[18], generally used to encrypt disks, is not authenticated; it provides the only guarantee that if a cipher has been modified, the decrypted data will be unreadable.

Disk encryption: Disk encryption is performed by the operating system, which encrypts or decrypts data on the fly by writing or rereading disks. It has the great advantages of being transparent to applications and users, and of being extremely efficient. On the other hand, it only protects against disk "tearing": once the machine is started, all data is accessible by an authenticated user on the system or on the application using it. Disk encryption systems are LUKS[19]on Linux, BitLocker on Windows or FileVault on macOS. Since BitLocker's code is not open source, alternatives exist such as VeraCrypt[20], whose code is free, or CRYHOD[21]of Prim'x qualified by ANSSI. Disk encryption generally uses AES XTS (see above), the AES key itself being encapsulated in another key. This other key, called the KEK (*Key Encryption Key*) must be, at a minimum, stored in the machine's TPM, or better, in an external KMS.

VM Confidential: use of virtual machines whose memory and disks are encrypted. Memory encryption is performed using a non-extractable secret hidden in the CPU (and possibly GPU) of the machine; the disk is encrypted using a secret hosted in a vTPM or better a KMS (see above). These confidential VMs allow you to operate in complete confidentiality on another's machine, typically that of a host, with high performance: around 5 % penalty compared to a standard VM. Ready-to-use

[16] https://csrc.nist.gov/pubs/sp/800/38/d/final
[17] https://fr.wikipedia.org/wiki/Nonce_(cryptographie)
[18] https://csrc.nist.gov/pubs/sp/800/38/e/final
[19]https://github.com/libyal/libluksde/blob/main/documentation/Linux%20Unified%20Key%20Setup%20(LUKS)%20Disk%20Encryption%20format.asciidoc
[20] https://www.veracrypt.fr/code/VeraCrypt/
[21] https://www.primx.eu/en/encryption-software/cryhod-en/

hardened Linux distributions, such as Cosmian[22] VM, are available from major hosting providers.

Verifiability: confidential VMs provide confidentiality through encryption, but do not guarantee system integrity; a hardware component could have been modified by the host, the operating system rebooted with a module leaking data, a binary or template replaced by a compromised version. Verifiability adds a service to retrieve cryptographic fingerprints of an entire audited system, then to be able to verify them at any time on a running system. Hardware verification is provided by default on confidential CPUs and GPUs, full system verification is provided by agents such as those available in Cosmian VMs.

Encryption with access control: this type of encryption allows the implementation of *Data Centric Security*. Data is encrypted with attributes and only users who can present keys with access policies on those attributes can decrypt the data. This type of encryption helps protect against a common class of attack, that of compromising application permissions, such as privilege escalations. An example of this type of encryption is Covercrypt, recently standardized by ETSI[23].

Post-quantum encryption: this type of encryption provides protection against new attacks available on quantum computers (Shor and Grover algorithms for example). The goal here is to protect against a future attack, for long-lived data, which could be collected, encrypted today, then decrypted tomorrow, when quantum computers are widely available. On the symmetric encryption side, the solution is quite simple: double the key size, to 256 bits for AES, for example, which slows down the encryption, but does not increase the size of the ciphertexts. On the public key encryption side, the situation is more complex. The NIST (American National Institute of Standards and Technology) has chosen an algorithm, Crystals Kyber, and standardized it under the name ML-KEM[24]. American regulations require that all public-key encryption be switched to post-quantum before 2035. In Europe, there are no dates to date, and it is recommended not to use this algorithm directly, but to hybridize it with a classic algorithm using an elliptic curve. This is what Covercrypt does, standardized by ETSI. Post-quantum encryption, even hybridized, is efficient; it is carried out in a few hundred microseconds on average.

### *3.4.2 Risks addressed by cryptography*

| Lifecycle phase | Family of attacks | Specific attacks | Solution |
|---|---|---|---|
| Data Collection and Processing | Data poisoning | Backdoor poisoning | • Authenticated encryption<br>• Verifiability |

[22] https://docs.cosmian.com/cosmian_vm/overview/
[23] https://www.etsi.org/technologies/quantum-safe-cryptography
[24] https://csrc.nist.gov/pubs/fips/203/final

| | | | |
|---|---|---|---|
| | | Data replication | Encryption |
| | Data theft | Extracting data from storage | Encryption |
| Model construction | Poisoning & manipulation | Corruption of parameters | • Authenticated encryption<br>• Verifiability |
| | | Malicious code attack | Verifiability |
| | Model theft | Extraction from storage | Encryption |
| Provision / deployment | Diversion & manipulation | Model substitution | • Authenticated encryption<br>• Verifiability |
| | | Environmental compromise | • Authenticated encryption<br>• Verifiability |
| | | Backdoor activation | • Memory & network encryption<br>• Verifiability |
| Operation & maintenance | Poisoning & manipulation | Degradation attacks | • Authenticated encryption<br>• Verifiability |
| | | Compromise of *plugins* (or grafts) | Verifiability |
| | | Unauthorized access | • Encryption with access control<br>• Verifiability |
| | Model theft | Model extraction | Encryption |
| Meta-prompt extraction | | | Storage & network encryption |
| Decommissioning | Data retention | Data persistence | Post-quantum encryption |
| | | Reusing the model | Encryption with access control |

## 3.5 Adversarial attacks

An **adversarial attack** is an operation in which an "attacker" modifies the input of an AI system to make it produce a different output than the attacked AI system would have produced if it had received the original, unmodified input. This is known in cybersecurity as an *evasion attack*.

To carry out an attack, the attacker must therefore be able to modify the input of the AI model and ensure that this modified input is submitted to the model. The mechanism is as follows:

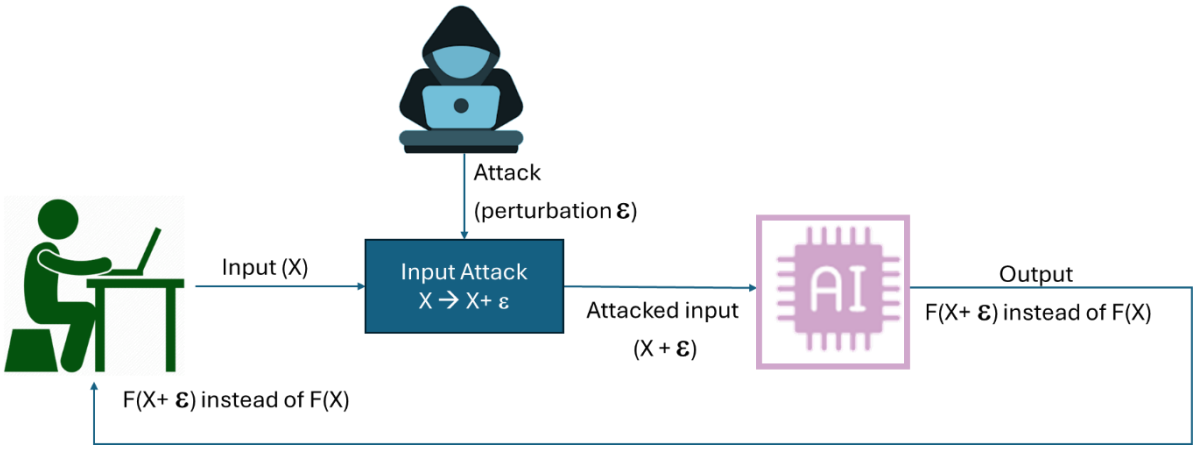



*Figure 15 – Adversarial attack*

In this attack, the attacker is not trying to modify or degrade the AI model of the attacked system. The attacker is only interested in making it produce an output that is inconsistent with the output it would have provided based on the original input before the modification. If the modification of the input does not generate a change in the output, then the attack will have failed.

Two types of attacks can be considered:

- The attack with target (*targeted attack*): in this type of attack, the attacker wants the output produced by the AI model being attacked to be equal to a specific target.
- Targetless attack: in this type of attack, the attacker only seeks to produce an erroneous result, without this erroneous result corresponding to a particular target.

Beyond the ability to access the input, modify it, and then submit the modified input to the AI system, the challenge for the attacker is to size the modification to the input so that it is:

- Weak enough that the modified input is not easily detected and therefore rejected by suitable protection mechanisms of the AI system.
- Strong enough that this change has an impact on the system output.

One of the first operational examples of an adversary attack was the falsification of a road sign to disrupt a driver assistance system. Typically:

- The raw input to the AI system analyzing the image (typically a ConvNet-type neural network), excluding attack, is the no entry sign (for example) on the left in the figure:

*Figure 16 – Attack on the entrance (on the left the entrance, a disturbance with stickers in the middle, the modified entrance on the right and finally the recognized sign)*

- The attack consists of putting stickers on the panel (which then play the role of the ε) so that the camera will submit the image of the modified panel (in the middle-right in the figure) to the AI system.
- If the attack is successful, these "additions" to the sign will change the output of the AI system, which will not recognize the "no entry" sign but any other sign (or any other object or even not detect an object at all). A target attack would be an attack sized so that the output of the AI system is equal to, for example, the "one way" sign (far right in Figure 16).

Overall, the attack is represented in the following Figure 17:

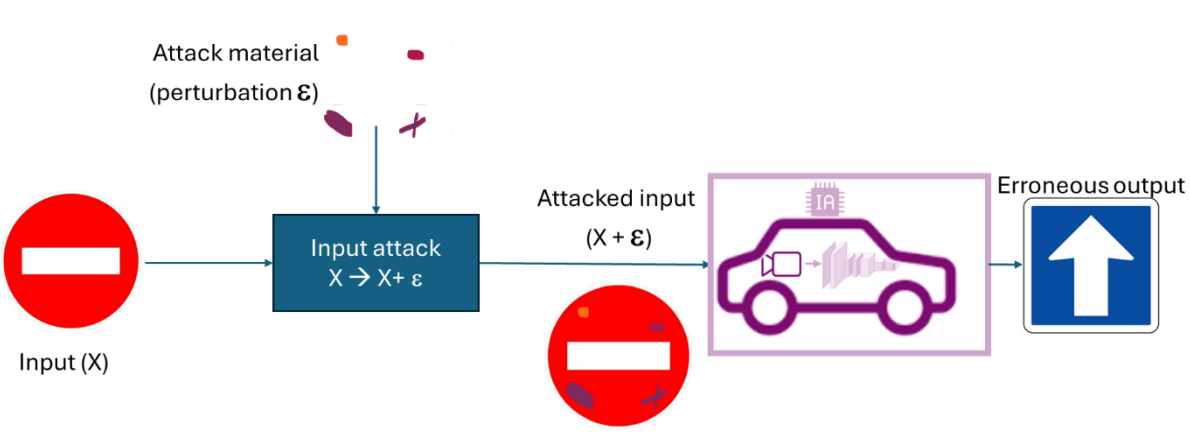



*Figure 17 – Adversary attack on a traffic sign*

In the general case, the effective sizing of the modification to be made to succeed in the desired attack will be facilitated by the fact that the attacker can have access to information about the AI system. Two cases can indeed arise:

- The structure and parameters of the model resulting from the system training are known to the attacker (for example in the case of an open-source model), this is called a "white box" attack. The attacker then has complete freedom to scale their modifications and carry out relevant or even targeted attacks.
- The structure and parameters of the model resulting from the system training are not known to the attacker; this is called a "black box" attack. If the attacker wants to define a modification that meets the constraints set out above, he will have to create a model that approximates the model he wants to attack. This "substitution" model will then allow him to calculate the modifications.

The attacker will then be confronted with the main question posed by this type of attack, that of transferability[25]of the attack: can an attack set on a substitution model work on a different model, and if so, with what probability of success? Although studies show that in some cases the attack can succeed with a certain probability, success is not guaranteed a priori regardless of the application.

For its part, the operator of the AI system will have to provide a system that is sufficiently robust so that a modification of the input below a detection threshold[26] that he has implemented does not modify the output of the said system. The defense strategy will depend on the knowledge that an attacker may have of the model and its parameters.

Here we have discussed adversarial attacks that aim to modify the inputs of the AI system. Attacks that aim to modify the outputs once calculated fall under cybersecurity in the sense of protecting the computer exchange channels between the system and the user.

For more details on adversarial attacks, please see[27].

# 4 Protect yourself

## 4.1 Prevention

Preventing attacks on AI includes a large number of methods that we will briefly present. The pedagogical fact sheets describing attacks on AI (see section 5) describe on the back of the sheet the specific prevention measures that should be implemented before putting the AI system into production to avoid the type of attacks described in the sheet (see section 5.1.2.2). All the prevention measures that will be described in these sheets will not be detailed here.

[25] https://arxiv.org/pdf/1605.07277

[26] In the case of an LLM, a defense mechanism could be, for example, to have the user confirm by reformulating their prompt that the LLM does indeed have the correct initial prompt, and this via a channel other than the one possibly attacked...

[27] https://arxiv.org/pdf/1412.6572 And https://arxiv.org/pdf/2302.09457

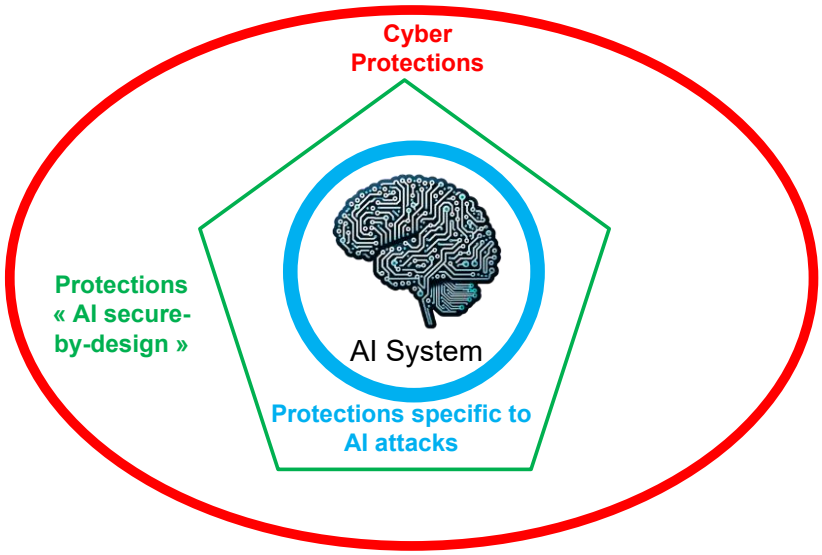



*Figure 18 – Protection of an AI system*

### *4.1.1 Types of preventive measures*

#### *4.1.1.1 Classic cybersecurity prevention measures*

Any cyber-attack is generally a series of actions (the *kill chain*) which will be linked together until the attacker achieves his objectives. MITRE [18] describes the different tactics used. Similarly, an attack on AI is a series of malicious actions and will generally begin with a classic cyberattack (for example, the attacker must gain access to the system's data), so to prevent an attack on AI, we must start by implementing all the classic cybersecurity techniques: *an attack on AI is a cyberattack + an AI-specific attack.* The AI attack prevention system therefore consists of three interlocking lines of defense, as shown in Figure 18 above. Classic cybersecurity prevention measures are, for example, presented by ANSSI in its IT hygiene guide which identifies 42 [3]. We will not detail them here.

#### *4.1.1.2 AI-specific prevention measures*

In the previous section 2.2, we already listed the specific elements of an AI system that require additional attention.

To go into more detail, in April 2024, ANSSI published a guide on *Security Recommendations for a Generative AI System* [1] which lists 35 recommendations to follow to build a (generative) AI system *secure-by-design*. Section 2.3.4 lists the 35 ANSSI recommendations for ensuring this protection from the design stage. This guide followed a document [20] published by all major global security agencies, *Guidelines for Secure AI System Development*, providing guidelines to help vendors build AI systems that perform as intended, are available when needed, and operate without revealing sensitive data to unauthorized parties.

Finally, ANSSI recently published with numerous partners a *Joint High-level Analysis of Cyber Risks Related to AI*[2]. The document proposes a list of recommendations that refine the 35 previous recommendations. Note that as an extension of this

document, we could deepen the risk analysis to prioritize countermeasures according to the level of risk. We have not carried out this work here.

#### *4.1.1.3 Specific prevention measures for certain types of attacks on AI*

Finally, targeted prevention measures can be implemented for certain types of AI attacks. Here are some examples:

- To protect data from poisoning during training: Mechanisms should be implemented to identify unexpected or malicious data that could impact model training. Where possible, data should be encrypted at rest and in transit (see section 4.2 on cryptography). It will also be possible to train the AI system to protect itself from poisoning by training it on poisoned data in addition to training data.
- To protect against model poisoning and manipulation: When using open-source models, a comprehensive security assessment will be performed on all dependencies and third-party components, such as libraries, *frameworks* or downloaded generative AI models, to analyze their reputation, known vulnerabilities, and their security posture. It is best to download them from reputable repositories, trusted platforms with well-established security practices, and only stable and well-maintained versions.
- AI security solutions radar: some vendors already offer solutions aimed at protecting against certain attacks. Wavestone has published an *AI Security Solutions Radar 2024 [13]* which identifies in September 2024 88 publishers offering solutions for:
  - Anti deepfake;
  - Data Protection and AI Privacy;
  - Detection and response of Machine Learning algorithms;
  - Secure Chatbot and LLM Filtering;
  - Secure Collaboration in Machine Learning;
  - Assessment of the robustness and vulnerabilities of the model;
  - AI Risk Management;
  - Synthetic data / Anonymization;
  - Ethics, explainability and fairness of treatment;
  - Compliance with AI regulations.

We have undertaken work to explore the software solutions market and will publish a dedicated report on the subject.


*Figure 19 – Wavestone Radar of AI Security Solutions*

### *4.1.2 Prevention measures by phase of the lifecycle*

To establish preventive measures adapted to the context of use of AI models, it is first necessary to implement a Risk Management Framework as described by NIST[28] to identify, assess, and manage risks associated with AI models. This includes categorizing information, selecting security controls, and ongoing monitoring.

In particular, ANSSI recommends an approach based on cyber risks [2] to develop trust in artificial intelligence. Risk assessment must be carried out throughout the lifecycle of an AI model, from its conception to its disposal, and taking into account the different IT environments (development, testing and validation, operation, etc.) on which it relies during each phase of its lifecycle. The means of protection must always be adapted to the business context and the identified risks.

The elements to be taken into account for these risk analyses are:

- The **computer systems** underlying that provide storage, computing and processing capabilities.
- The **AI model** in itself (parameters, storage format, etc.).
- The **data** which are used to train the AI model, but also those which are recovered during the exploitation phase of this model through the RAG mechanism applied to certain company data or directly on the Internet.
- The models' **inputs/outputs** and interactions with humans or with other AI models or/and computer systems. In the latter case, this also includes process automation technology.

In order to maintain consistency between attacks and means of defense, and as we have a classification of attacks according to the 7 phases of the lifecycle of an

[28] https://csrc.nist.gov/pubs/sp/800/37/r2/final

OECD AI model (see our taxonomy in section 2.5), the prevention measures to be applied must also be based on this structure:

A. Planning and design,
B. Data collection and processing,
C. Construction of the model / adaptation of an existing model,
D. Testing, evaluation, verification,
E. Provision, use, deployment,
F. Operation and maintenance,
G. Decommissioning / scrapping.

In order to identify the preventive measures to be implemented, we relied on the following documents:

- ANSSI, Security recommendations for a generative AI system, [1]
- ANSSI, Developing trust in AI through a cyber risk approach, [2]
- ANSSI, IT Hygiene Guide, [3]
- MITRE ATLAS, [17]

For clarity, all recommendations from these documents have been listed in Appendix 1 (even measures that fall within the context of "I- Cybersecurity protection on the infrastructure" and which are therefore not specific to AI).

The reflections carried out on each of them have been clearly outlined and explained, as follows:

- The ***duplicates*** have been identified,
- The distribution according to the different lines of defense illustrated in figure 18 has been made:
  - I Cybersecurity protections on infrastructure,
  - II AI "Secure by design" protection,
  - III Specific protections against AI attacks.
- The distribution according to the ***7 phases of the OECD:*** this has not been carried out for the classic cybersecurity protection measures, since it is not relevant in this generic framework which does not specifically concern AISs, nor therefore their lifecycle,
- A ***harmonized presentation of* measures' *categories*** is proposed (taking into account those already existing in the original documents).

To summarize:

| **Document source** | **Extraction of raw preventive measures** | **Treatments carried out on prevention measures (color codes as in Figure 18)** |
|---|---|---|
| ANSSI [1] | All 35 listed measures | • Some have been classified as classic cybersecurity measures. |

| | | |
|---|---|---|
| | | • Some have been categorized as “secure by design” AI-specific prevention measures.<br>• Duplicates identified with [2], [3] and [17]<br>• Distribution according to the 7 OECD phases |
| ANSSI [2] | All 43 listed measures | • Some have been classified as classic cybersecurity measures.<br>• Some have been categorized as “secure by design” AI-specific prevention measures.<br>• Duplicates identified with [1], [3] and [17]<br>• Distribution according to the 7 OECD phases |
| ANSSI [3] | All 42 listed measures | These measures have been classified as classic cybersecurity measures. |
| MITRE ATLAS [17] | All 25 listed measures | • Some have been classified as classic cybersecurity measures.<br>• Some have been categorized as “secure by design” AI-specific prevention measures.<br>• Some have been categorized as measures specific to certain attacks on AI.<br>• Duplicates identified with [1] and [2]<br>• Distribution according to the 7 OECD phases |

The result of this work is a consolidated and synthetic list of prevention measures presented through the tables proposed in section 9 (Annex 1 - Prevention methods). These tables make it possible to quickly identify the prevention measures to be deployed in each phase of the lifecycle of an OECD AI model and according to the protection context.

Furthermore, throughout the lifecycle of this document, new prevention measures specific to certain attacks on AI will be added, as the analysis on the attack sheets progresses. Among the sources already identified, we can mention:

- Preventive measures listed on the presented attack sheets,
- Scientific articles,
- Experience of the members of the working group that produced this document,
- Security solution publishers.

## 4.2 Remediation

### *4.2.1 Incident Management Architecture for AI Systems*

Faced with growing threats to Artificial Intelligence Systems (AIS), effective and structured incident management is essential to ensure resilience, security and regulatory compliance. We therefore propose an incident management architecture applied to AI systems, integrating best practices from reference frameworks such as ISO/IEC 27035, ANSSI and NIST recommendations, and CNIL

guidelines [21 – 24]. It is structured around three main components: Governance and Crisis Management, Detection and Investigation, and Remediation and Reconstruction, accompanied by a continuous improvement loop to ensure the resilience and optimization of AI incident response processes.

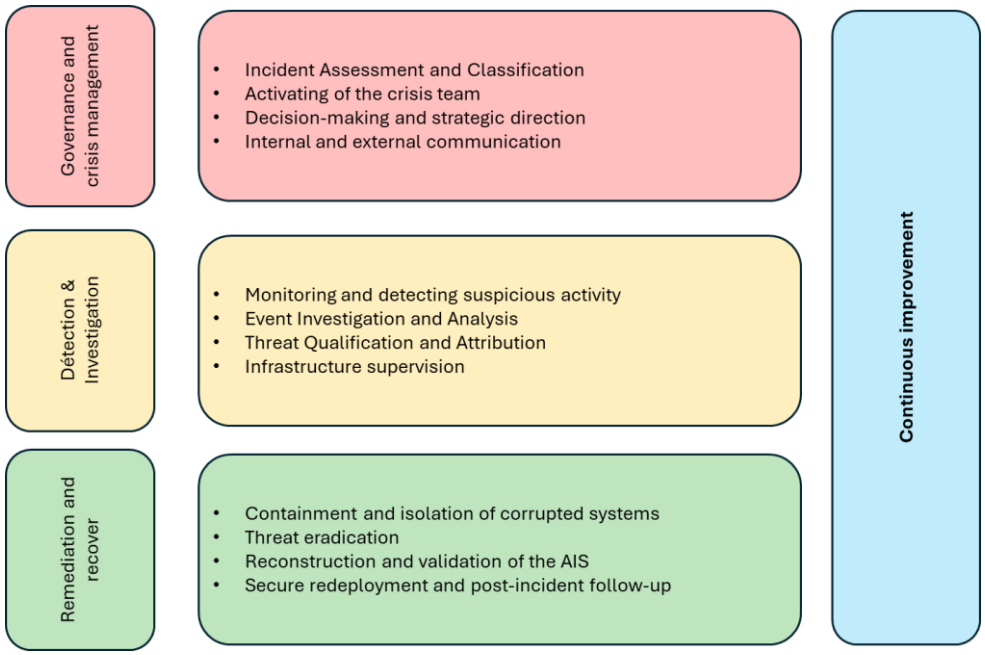



*Figure 20 – Incident management for AI systems*

### 🔴 Governance & Crisis Management

Crisis governance and management ensure the orchestration and strategic alignment of responses to AI incidents, ensuring optimized responsiveness and regulatory compliance. This phase is based on:

- The assessment and classification of incidents according to their impact on the Traceability, Availability, Integrity and Confidentiality of AI systems.
- The activation of a crisis unit mobilizes the SOC, DevSecOps, AI and Legal teams, while ensuring coordination with regulators (ANSSI, CNIL, partners).
- Real-time risk analysis guides decision-making, allowing the response to be directed towards immediate containment, in-depth investigation or priority remediation.
- Managing internal and external communications ensures transparency and compliance with notification obligations.

This phase aligns with the NIST CSF Cyber Resilience Principles (*Govern & Identify*) and ANSSI recommendations, ensuring effective and strategic supervision of AI incidents.

### 🟡 Detection & Investigation

The detection and investigation phase allows for proactive monitoring and qualification of threats; it is based on:

- Proactive monitoring and in-depth analysis of AI threats to quickly identify compromises and qualify attacks. The SOC (*Security Operations Center*) exploits indicators of compromise (IoC) and relies on *Threat Intelligence* solutions and correlation of security events via SIEM for advanced and reactive detection.
- Deep forensics analyze AI model training and inference flows to detect adversarial attacks, data poisoning, and algorithmic drift.
- Attributing attacks and qualifying threats helps guide containment and remediation measures adapted to the criticality of the incident.
- Continuous monitoring of AI infrastructures through auditing of MLOps pipelines, monitoring of API flows and behavioral analysis of deployed models is essential to anticipate risks of compromise and strengthen the AI cybersecurity posture.

This phase follows the supervision principles of the CNIL and the NIST CSF (*Detect*), guaranteeing an optimized detection and investigation capacity against emerging threats targeting AIS.

🟢 **Remediation & Reconstruction: Containment, validation and secure redeployment**

Remediation and reconstruction follow ANSSI's E3R (Containment, Eviction, Eradication, Reconstruction) model, guaranteeing secure recovery of AISs.

- Containment and isolation of compromised systems help stopping the spread of the attack by restricting access to affected infrastructure.
- Threat eradication removes malicious access and neutralizes intrusion vectors to prevent threat persistence.
- AIS reconstruction and validation involve correcting vulnerabilities, cleaning up AI datasets, and verifying the integrity of models and infrastructure.
- Secure redeployment and post-incident monitoring ensure a return to production without residual risk, validated by a compliance and cybersecurity audit.

This approach ensures a return to service that meets NIST CSF requirements (*Respond & Recover*), minimizing the risks of recurrence and ensuring enhanced resilience of AI systems against future threats.

🔵 **Continuous Improvement Loop**

Continuous improvement is essential to take advantage of each incident and sustainably strengthen the cybersecurity posture of AIS. This phase is based on structured Feedback on Experience (RETEX), allowing incidents to be documented, exploited vulnerabilities to be identified, and detection and remediation strategies to be refined. The evolution of AI cybersecurity policies is based on updating detection models and optimizing supervision mechanisms to anticipate new threats. In parallel, ongoing training of teams through AI Red Team exercises (see

Glossary), adversarial simulations and intrusion tests support developing proactive response capabilities to cyberattacks targeting AI systems. This approach is aligned with the principles of the NIST CSF (*Improve*) and the recommendations of ANSSI, guaranteeing a progressive and adaptive strengthening of AI cybersecurity.

#### *4.2.2 Remediation checklist aligned with the lifecycle of an AIS*

To effectively respond to incidents affecting an AIS, we rely on a comprehensive methodology, aligned with international standards (ISO/IEC 27035, NIST CSF, ANSSI, CNIL) and crisis management principles applied to IA environments. This approach covers the entire lifecycle of an AIS and is integrated into a well-defined incident response architecture.

To facilitate its implementation, an operational checklist has been developed. It includes strategic and technical actions enabling:

- To anticipate risks and structure AI security governance.
- To identify and qualify the threats weighing on AI models and their infrastructures.
- To effectively remediate attacks and restore impacted AI systems.
- To continuously improve the AI security posture through structured feedback.

This approach is pragmatic, adaptable, and tailored to the challenges of modern AI systems. Thus, the remediation checklist would allow CISOs, CTOs, and CIOs to effectively structure their responses to AI incidents, ensuring methodical implementation in line with cybersecurity best practices. The checklist is provided in the Appendix (Section 10) and provides the remediation methods used in the attack fact sheets.

# 5 Fact sheets: main attacks analyzed

## 5.1 Fact sheets format

The purpose of this section is to provide a practical understanding of known AIS compromise scenarios. To achieve this in the most readable and effective way possible, we propose the use of fact sheets in the following format.

#### *5.1.1 On the front side of the sheet*

For the front of a sheet, the following representation is proposed in Figure 21.

The front of the fact sheet provided in this section is read from top to bottom and from left to right. This arrangement is intended to initially describe the attack typology studied and to gradually go into detail about the scenario.

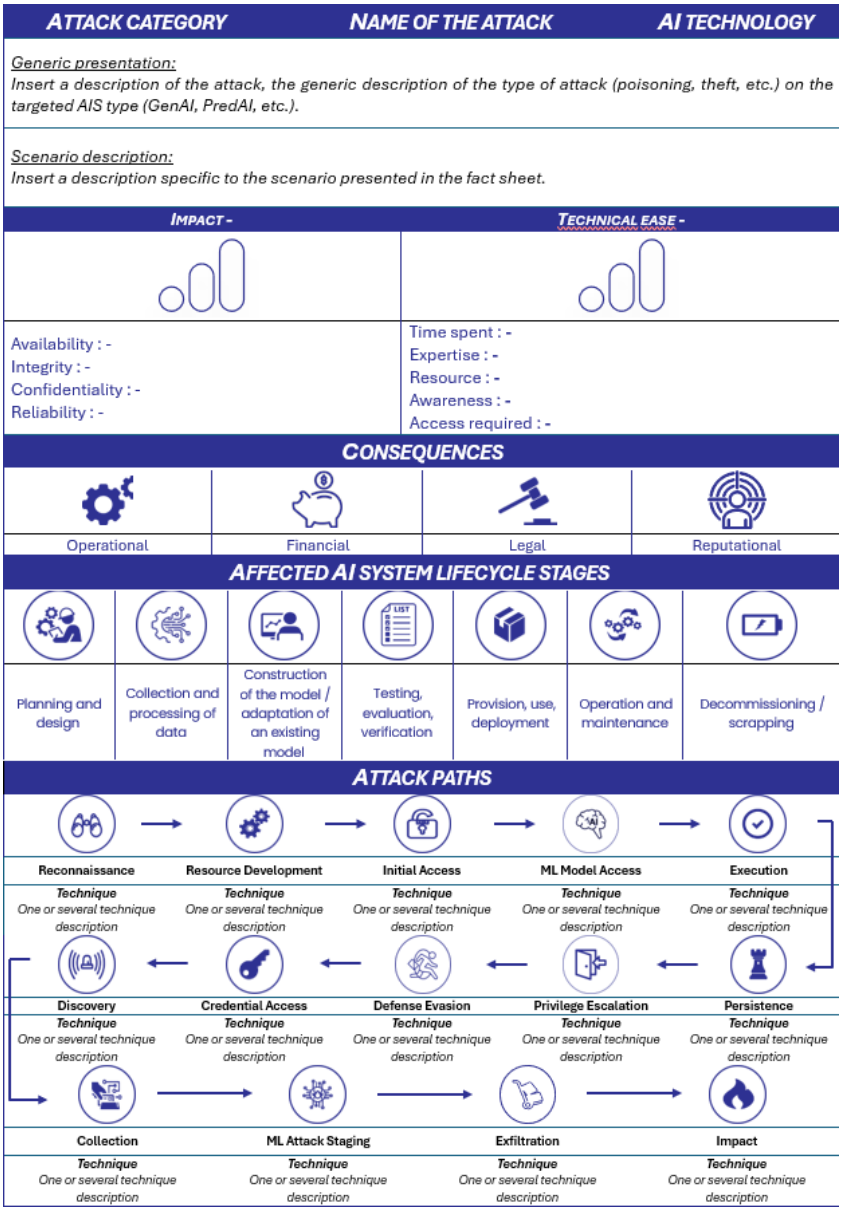



*Figure 21 – First page of the descriptive sheet of an AIS attack*

### 5.1.1.1 Description of the attack scenario

The document begins with the following descriptive segments:

| ATTACK CATEGORY | NAME OF THE ATTACK | AI TECHNOLOGY |
|---|---|---|
| Generic presentation:<br>*Insert a description of the attack, the generic description of the type of attack (poisoning, theft, etc.) on the targeted AIS type (GenAI, PredAI, etc.).* | | |
| Scenario description:<br>*Insert a description specific to the scenario presented in the fact sheet.* | | |

*Figure 22 – A blank scenario description format on the front of the sheet*

The following legend helps you understand the interest of each of the fields:

| | |
|---|---|
| *Attack category* | The *"attack category"* qualifies the attack category[29] which the scenario studied is part of. These will be examples of attack categories presented previously. |
| *Name of the attack* | The « *name of the attack*" describes the attack scenario studied in the pedagogical sheet. This is one of the scenarios listed in this deliverable among the major attack categories. |
| *AI technology* | The "*AI technology*" describes the artificial intelligence technology targeted by the attack scenario studied in the file. |
| *Generic presentation* | The "*generic presentation*" is a succinct and generic description of the attack category. |
| *Scenario Description* | The « *scenario description*" is a succinct description of the implementation of the scenario and its challenges for the targeted artificial intelligence system. |

#### *5.1.1.2 Attack scenario qualification*

The sheet continues with segments relating to the qualifications of the attack scenario. The purpose of this section is to propose a series of indicators to differentiate the severity of one attack scenario from another.

| IMPACT - | TECHNICAL EASE - |
|---|---|
|  |  |
| Availability : -<br>Integrity : -<br>Confidentiality : -<br>Reliability : - | Time spent : -<br>Expertise : -<br>Resource : -<br>Awareness : -<br>Access required : - |

*Figure 23 – A blank format for evaluating criteria and indicators*

The qualification method for each criterion and indicator is presented and detailed earlier in this document[30]. Criteria will be grayed out if an impact assessment was deemed not applicable or relevant to the nature of the attack.

#### *5.1.1.3 The consequences of attack scenarios*

In order not to exclude the strategic consequences of an attack on an organization, the section "*Consequences"* proposes to identify complementary impacts: operational, financial, legal or reputational.

[29] As a reminder, the main categories of attack are listed in the form of a proposed taxonomy in section 2.5.

[30] A section is dedicated to this topic in 2.4 Qualitative evaluations of attacks.

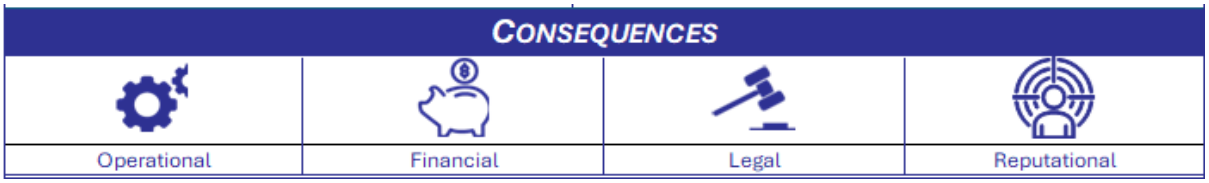



*Figure 24 – A blank format for identifying the strategic consequences of an attack*

The consequences are identified and justified upstream in this document[31].

#### 5.1.1.4 The stages of the lifecycle of the affected AI system

To contextualize an attack in the lifecycle of an AI system, it is proposed to identify the stages of the cycle most likely to be subject to these scenarios. To do this, the decision was made to adopt the OECD lifecycle approach (as mentioned in 2.1.2.1). This approach was chosen because of its effectiveness in summarizing the key stages of the lifecycle of an AIS while remaining agnostic of the technologies used. On a pedagogical sheet, this will therefore involve:

- To leave in blue the step(s) that could constitute a relevant context for the implementation of the attack scenario; or conversely,
- To gray out the lifecycle stage(s) if the attack has technical specificities or a mode of operation such that the scenario has little or no probability of occurring.

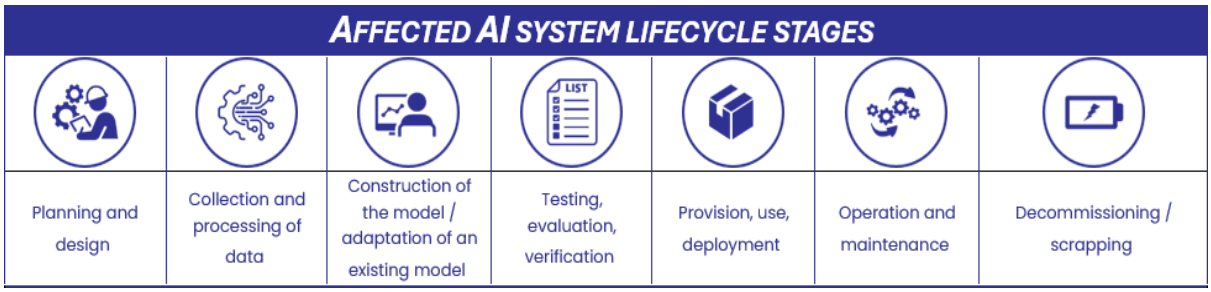



*Figure 25 – A blank format for identifying affected AIS lifecycle stages*

The stages of the AIS lifecycle are selected based on the assessment of their relevance at the time of writing the pedagogical sheet on the attack scenario.

#### 5.1.1.5 The attack pattern

The contextualization of the attack scenario continues with an exercise that consists of identifying the steps likely to be followed by an attacker. The objective is to sequence the path taken by the malicious user in implementing the scenario.

For this purpose, it was considered useful to use the MITRE Atlas [17] reference framework, which allows the tactics and techniques used to be highlighted based on the analysis of the scenario. The proposed visual representation is as follows:

[31] A section is dedicated to this subject in 2.4.4 The consequences of an attack on the organization

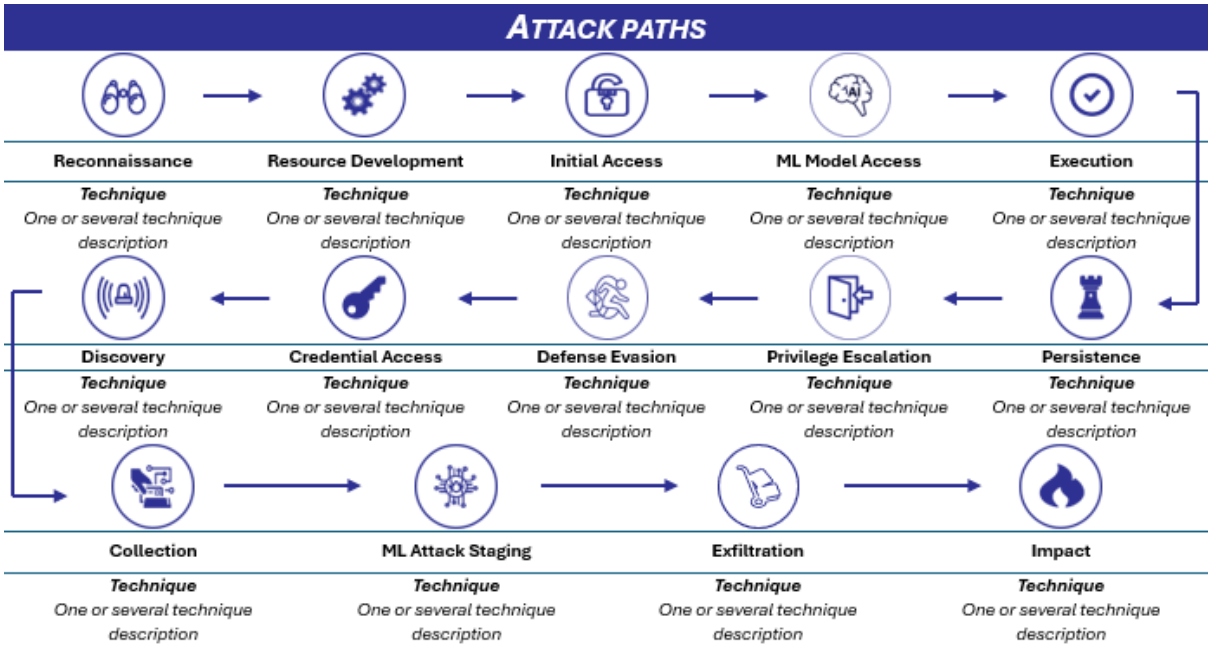



*Figure 26 – A graphical representation of the MITRE Atlas knowledge base in blank format*

Before going into the explanation of the different elements, it is appropriate to develop the reading order of the proposed graph. The proposed reading order is that chosen by MITRE to list the different tactics, this list was mentioned previously[32]. Which means that the reading must be done in the direction of the arrows proposed on the graph in *Figure 26*. The order is therefore materialized as follows: Reconnaissance, Resource Development, Initial access, AI Model Access, Execution, Persistence, Privilege Escalations, Defense Evasion, Credential Access, Discovery, Collection, AI Attack Staging, Exfiltration and Impact.

It should be noted, however, that depending on the scenario presented in a pedagogical sheet, the order of tactics may vary.

Example: For model extraction, the "*Exfiltration*" can take place before the "*Setting up the ML attack* »

The following legend helps you understand the interest of each of the proposed fields:

[32] Reference is made here to the enumeration of tactics made in 2.3.2.

| | |
|---|---|
| *Tactic* | A tactic is the attacker's objective, it appears in bold under the pictogram that graphically represents it. The MITRE Atlas lists 14 of them. |
| *Technique* | A technique represents the method by which he will seek to accomplish his objective. The techniques chosen to explain the scenario are found below the title of the tactic. The MITRE Atlas matrix [17] lists about 62 of them, each one has a code[33]The techniques selected are those that appeared to be relevant at the time of the analysis. Where appropriate, these will be accompanied by descriptions. |

### *5.1.2 On the back of the sheet*

For the back of a pedagogical sheet, the following representation is proposed:

**REMEDIATION MEASURE**

| Action | Teams to mobilize | Lifecycle stage | Complexity | Efficiency |
|---|---|---|---|---|
| **Measure 1** | AI team & production team |  | + | +++ |
| **Measure 2** | AI team & production team |  | +++ | + |

**PREVENTIVE MEASURE**

| | | | | |
|---|---|---|---|---|
| **Measure 1** | AI team & production team |  | + | +++ |
| **Measure 2** | AI team |  | ++ | ++ |
| **Measure 3** | AI team |  | +++ | +++ |

**TO GO FURTHER**

- [...]

**KNOWN EXAMPLES**

- [...]

*Figure 27 – Back of the descriptive sheet of an attack on an AIS*

The back of a pedagogical sheet is read from top to bottom and from left to right. This arrangement aims to successively describe the suggested remediation and prevention measures, the documentary sources used and some known examples of the implementation of the scenario.

[33]Example: the prompt injection has the code AML.T0051https://atlas.mitre.org/techniques/AML.T0051

#### *5.1.2.1 Prevention*

The purpose of the back of the pedagogical sheet is to firstly propose a remedial method, and secondly in the section "*Prevention*" to attempt to design an approach to anticipate, block or prevent a new attack of this type.

| PREVENTIVE MEASURE | | | | |
|---|---|---|---|---|
| **Measure 1** | AI team & production team |  | + | +++ |
| **Measure 2** | AI team |  | ++ | ++ |
| **Measure 3** | AI team |  | +++ | +++ |

*Figure 28 – A blank format of the section dedicated to attack prevention*

The following legend defines the different fields proposed for listing these prevention measures.

| | |
|---|---|
| *Action* | The section "*Action*" lists the measures adopted, at the time of writing the pedagogical sheet, to raise awareness, anticipate, or provide means to prevent or block an attack similar to the scenario studied. A measure is assigned to a team, located at the stage of the AIS lifecycle and evaluated in terms of its complexity and effectiveness. |
| *Teams to mobilize* | The teams to be mobilized are the ones responsible for implementing the preventive measure. This will be the team considered to be most capable of intervening to anticipate the scenario presented. |
| *Lifecycle stage* | The lifecycle stage is the section used to locate the most relevant prevention measure in the AIS lifecycle to prevent or block the scenario. |
| *Complexity* | Complexity is a succinct proposal for assessing the obstacles encountered in implementing the measure. It is done at three levels:<br>*+*The measure appears reasonably simple to implement. It requires few human, technical or time resources to be implemented;<br>*++*The measure involves mobilizing additional human and/or technical resources to be implemented;<br>*+++*The measure appears complex to implement and requires advanced human and technical resources, and time to be implemented. |

| | |
|---|---|
| *Efficiency* | Effectiveness is a succinct proposal for evaluating the effects of the measure on the risks and impacts of the attack.<br>+The measure does not allow for the anticipation or blocking of the risks and impacts of the attack on the system: it must be accompanied by other technical and organizational measures;<br>++The measure makes it possible to anticipate or partially or medium-term block the risks and impacts of the attack on the system.<br>+++The measure allows in the short term to significantly anticipate or block the risks and impacts of the attack on the system. |

#### *5.1.2.2 Remediation*

The scenario study approach aims to assess an attack and situate it within the lifecycle of an AIS. To be complete, the next step is to list, evaluate, and assign remediation measures deemed relevant. A remediation measure is understood to be: a more or less long-term action to limit the risks and impacts of an attack studied in one of the pedagogical sheets.

**REMEDIATION MEASURE**

| Action | Teams to mobilize | Lifecycle stage | Complexity | Efficiency |
|---|---|---|---|---|
| **Measure 1** | AI team & production team |  | + | +++ |
| **Measure 2** | AI team & production team |  | +++ | + |

*Figure 29 – A blank format of the section dedicated to attack remediation*

The following legend defines the different fields proposed for listing these remediation measures.

| | |
|---|---|
| *Action* | The section "*Action*" lists the measures adopted at the time of writing the fact sheet, to reduce or eliminate the risks and impacts caused by an attack. A measure is assigned to a team located at the stage of the AIS lifecycle and evaluated in terms of its complexity and its effectiveness in reducing or eliminating the risks and impacts. |
| *Teams to mobilize* | The teams to be mobilized are the ones responsible for the remediation measure. This will be the team considered to be most capable of intervening to remedy the scenario in question. |
| *Lifecycle stage* | The lifecycle stage is the section used to locate the remediation measure in the AIS lifecycle that is most relevant to reducing the risks and impacts of the attack. |

| | |
|---|---|
| *Complexity* | Complexity is a succinct proposal for assessing the obstacles encountered in implementing the measure. It is done at three levels:<br>+The measure appears reasonably simple to implement. It requires few human, technical or time resources to be implemented;<br>++The measure involves mobilizing additional human and/or technical resources to be implemented;<br>+++The measure appears complex to implement and requires advanced human and technical resources, as well as time to be implemented. |
| *Efficiency* | Effectiveness is a succinct proposal for evaluating the effects of the measure on the risks and impacts of the attack.<br>+The measure does not resolve the risks and impacts of the attack on the system and needs to be accompanied by other technical and organizational measures;<br>++The measure makes it possible to partially or medium-term resolve the risks and impacts of the attack on the system.<br>+++The measure allows in the short term to significantly reduce or eliminate the risks and impacts of the attack on the system. |

#### *5.1.2.3 Supplements*

The fact sheets conclude with the documentary sources used and known cases identified. These elements enabled the writing of the attack scenarios listed below in this booklet.

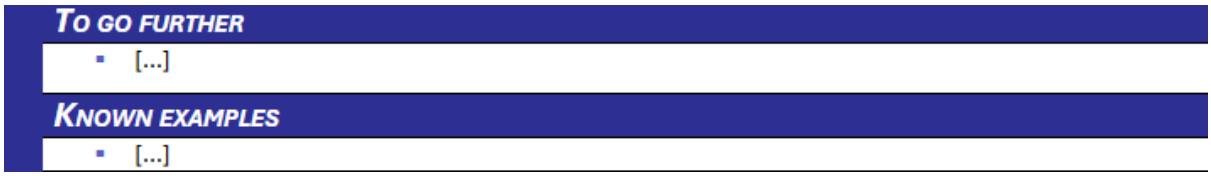


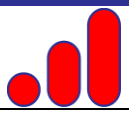

*Figure 30 – A blank format of the "Further Reading" and "Known Examples" sections*

| | |
|---|---|
| *To go further* | This section has been created to supplement and source the elements covered in the content studied. Sources may come from academic, scientific or institutional resources. |
| *Known examples* | This section aims to list known cases of implementation of the attack scenario examined in a fact sheet.<br>For instance: for chatbot poisoning, the Tay poisoning case would be a known example (see the sheet below). |

### *5.1.3 Demonstration using the example of the chatbot Tay*

The following sheet aims to illustrate the elements previously presented in this document as well as in sections 5.1.1. *On the front of the sheet* and 5.1.2 *On the other*

*back of the sheet.* This is a case of poisoning (category of attack) of the input data of a chatbot (name of attack), concerning generative AI (type of AI).

| POISONING | POISONING CHATBOT INPUT DATA | GENERATIVE |
|---|---|---|
| *Generic presentation:* *Modify a model's retraining data (e.g., history of conversations with users, etc.) to introduce a deviation in its behavior that can be exploited.* | | |
| *Scenario description:* *In the case of a chatbot using data from user interactions to continuously learn, malicious or risk-unaware users could provide it with data sets as input which, once used by the model to retrain, would cause unwanted responses from the model.* | | |
| IMPACT – Medium (2) | | TECHNICAL EASE – HIGH (3) |
|  | |  |
| Availability: N/A<br>Integrity: **High (3)**<br>Confidentiality: N/A<br>Reliability: **Average (2)** | | Time spent: **<1 day (3)**<br>Expertise: **Weak (3)**<br>Resource: **Average (2)**<br>Awareness: **Weak (3)**<br>Access required: **Internal user (2)** |
| CONSEQUENCES | | |
|  |  |   |
| Operational | Financial | Legal | Reputational |
| AFFECTED AI SYSTEM LIFECYCLE STAGES | | | |
|  |  |      | | | | | |
| Planning and design | Data collection and processing | Construction of the model / adaptation of an existing model | Testing, evaluation, verification | Provision, use, deployment | Operation and maintenance | Decommissioning / scrapping |
| ATTACK PATHS | | | | | | |
|      | | | | | | |
| Reconnaissance | Resource Development | **Initial Access** | **ML Model Access** | | | Execution |
| | | Access to the conversational platform AML.T0047 | Compromise of training data AML.T0010.002 | | | |
|      | | | | | | |
| Discovery | Credential Access | Defense Evasion | Privilege Escalation | **Persistence** | | |
| | | | | Training Data Poisoning AML.T0020 | | |
|     | | | | | | |
| Collection | ML Attack Staging | Exfiltration | **Impact** | | | |
| | | | Undermining the integrity of the model AML.T0031 | | | |

### *REMEDIATION*

| Action | Teams to mobilize | Lifecycle stage | Complexity | Efficiency |
|---|---|---|---|---|
| Return to stable versions of the model. | AI & Production Team |  | + | +++ |
| Rebuild the model with reliable data. | AI & Production Team |  | +++ | ++ |

### *PREVENTION*

| | | | | |
|---|---|---|---|---|
| Backup stable versions. | Production Release Team |  | + | +++ |
| Check the model retraining data. | AI Team |  | ++ | ++ |
| Re-evaluate the model after retraining. | AI Team |  | +++ | +++ |
| Implement a procedure called "*red button*". | Production Release Team |  | + | + |

### *TO GO FURTHER*

Attack on Microsoft's Tay chatbot:

- Wolf, M. J., Miller, K., & Grodzinsky, F. S. (2017). Why we should have seen that coming: comments on Microsoft's tay" experiment," and wider implications. Acm Sigcas Computers and Society, 47(3), 54-64.
- Lee, P. (2016, March 25). Learning from Tay's introduction - The Official Microsoft Blog. https://blogs.microsoft.com/blog/2016/03/25/learning-tays-introduction/
- AI Incident Database. https://incidentdatabase.ai/cite/6/#r1374

Poisoning attacks:

- OWASP Top 10 for LLM Applications VERSION 1.0.1. (2023). https://owasp.org/www-project-top-10-for-large-language-model-applications/assets/PDF/OWASP-Top-10-for-LLMs-2023-v1_0_1.pdf, section "LLM03: Training Data Poisoning"
- Vassilev, A., Oprea, A., Fordyce, A., & Anderson, H. (2024). Adversarial Machine Learning: https://doi.org/10.6028/nist.ai.100-2e2023, section 3.2.2 : "Poisoning Attacks"

### *KNOWN EXAMPLES*

- In 2016, Microsoft's Tay chatbot was manipulated by malicious users on Twitter, who bombarded it with racist and offensive messages. Reused in Tay's training, which continuously learned based on its interaction history, these messages caused the chatbot to start posting racist and offensive messages.
- Within 24 hours, Tay was deactivated to prevent further damage.

*Figure 31 – Tay's case description sheet*

## 5.2 Attack sheets by phase

Here we present 10 attack sheets in different phases of the lifecycle as indicated in the taxonomy below:

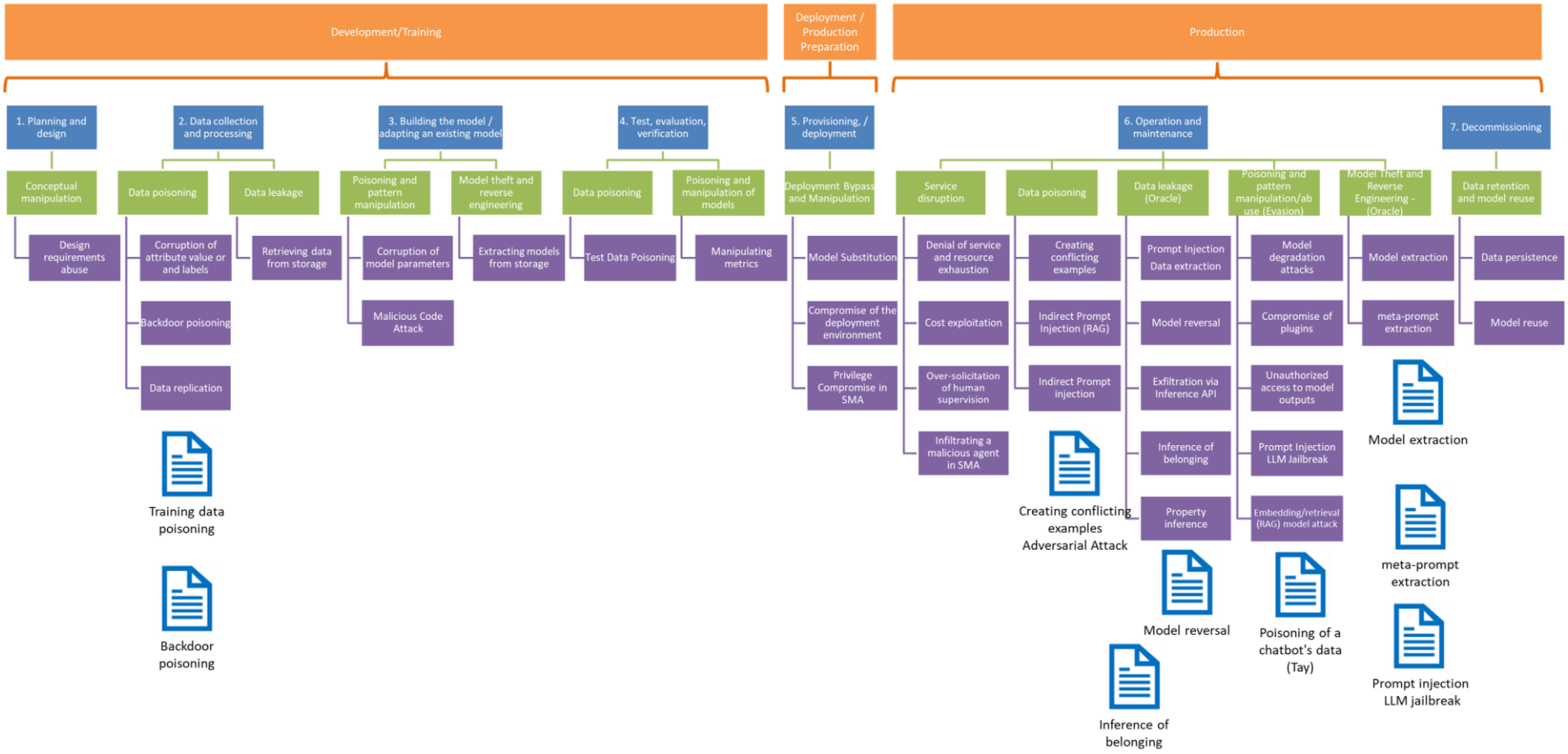



*Figure 32 – Files presented*

### *5.2.1 Planning and design*

#### *5.2.1.1 Conceptual manipulation*

##### 5.2.1.1.1 Handling design requirements

[File to come]

### 5.2.2 Data collection and processing

#### 5.2.2.1 Data poisoning

##### 5.2.2.1.1 Corruption of attribute values or labels

| POISONING | TRAINING DATA POISONING | PREDICTIVE & GENERATIVE |
|---|---|---|

*Generic presentation:*
*Poisoning aimed at modifying training data to mislead the model during training.*

*Scenario description:*
*The data itself or the labels on that data may be poisoned (i.e., modified). Depending on the proportion of training data that is poisoned and the quality of the poisoning, in its final use the model may provide an incorrect answer regardless of the data provided, or only for particular inputs.*

| IMPACT – HIGH (3) | TECHNICAL EASE – MEDIUM (2) |
|---|---|
| Availability: N/A<br>Integrity: **High (3)**<br>Confidentiality: N/A<br>Reliability: **High (3)** | Time spent: **Moderate (2)**<br>Expertise: **Average (2)**<br>Resource: **Low (3)**<br>Awareness: **Average (2)**<br>Access required: **General Public (3)** |

**CONSEQUENCES**

| Operational | Financial | Legal | Reputational |
|---|---|---|---|

**AFFECTED AI SYSTEM LIFECYCLE STAGES**

| Planning and design | Data collection and processing | Construction of the model / adaptation of an existing model | Testing, evaluation, verification | Provision, use, deployment | Operation and maintenance | Decommissioning / scrapping |
|---|---|---|---|---|---|---|

**ATTACK PATHS**

| Reconnaissance | Resource Development | Initial Access | ML Model Access | Execution |
|---|---|---|---|---|
| | Training Data Poisoning AML.T0020<br>Publication of poisoned datasets AML.T0019 | ML Supply Chain Compromise: Data *AML.T0010.002* | | |

| Discovery | Credential Access | Defense Evasion | Privilege Escalation | Persistence |
|---|---|---|---|---|
| Discover ML artifacts (training data) AML.T0007 | | | | Training Data Poisoning AML.T0020 |

| Collection | ML Attack Staging | Exfiltration | Impact |
|---|---|---|---|
| | | | Eroding integrity data and model AML.T0059 & AML.T0031 |

### *REMEDIATION*

| Action | Teams to mobilize | Lifecycle stage | Complexity | Efficiency |
|---|---|---|---|---|
| Return to stable versions of the model. | AI & Production Team |  | + | +++ |

### *PREVENTION*

| | | | | |
|---|---|---|---|---|
| Verify the origin and integrity of the training data. | Cybersecurity Team |  | ++ | +++ |
| Cleaning training data to remove possible poisoning. | AI Team |  | +++ | ++ |
| Searching for anomalies in training data using statistical methods. | AI Team |  | ++ | ++ |
| Monitor model performance metrics.<br>- Have a fixed set of reliable data on which to regularly test the model's performance. | AI Team |  | + | + |
| Reinforced model training. | AI Team |  | +++ | ++ |
| If the type of model chosen allows it, train the model directly on encrypted data. | AI Team |  | +++ | +++ |
| Verify the origin and integrity of the training data. | Cybersecurity Team |  | ++ | +++ |

### *TO GO FURTHER*

Poisoning attacks

- Apostol Vassilev, Alina Oprea, Alie Fordyce, Hyrum Anderson. Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations. https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.pdf, sections 2.3.1: “Availability Poisoning”, 2.3.2: “Targeted Poisoning” and 3.2.2: “Poisoning Attacks”
- OWASP Top 10 for LLMhttps://owasp.org/www-project-top-10-for-large-language-model-applications/, version 2025, section “LLM04: Data and Model Poisoning”
- Lucian Constantin. How data poisoning attacks corrupt machine learning models. https://www.csoonline.com/article/570555/how-data-poisoning-attacks-corrupt-machine-learning-models.html

### *KNOWN EXAMPLES*

- This example illustrates the case where the data itself is modified, causing the model to predict false results: Virus Total Poisoning. https://atlas.mitre.org/studies/AML.CS0002
- This example shows how to modify public data that can be used to train models: Web-Scale Data Poisoning: Split-View Attack. https://arxiv.org/pdf/2302.10149
- This example illustrates the case where the data is modified in a controlled way so that models trained with this data provide unpredictable predictions. https://www.siliconrepublic.com/machines/ai-art-nightshade-poison-images-glaze

### 5.2.2.1.2 Backdoor poisoning

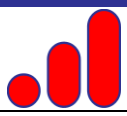
| POISONING | BACKDOOR POISONING | PREDICTIVE |
|---|---|---|

*Generic presentation:*

*A backdoor attack consists in injecting a malicious behavior into a model during the training phase, generally through data manipulation and then activating it during the inference phase using a trigger.*

*Scenario description:*

*The attacker inserts a small number of corrupted examples into the training set. These examples are incorrectly labeled but share a specific reason (the trigger), sometimes imperceptible to a human. The model then learns to associate this pattern with a target label. In inference, the model normally works on clean data, but if the trigger is present in an input, the model will produce the output desired by the attacker.*

| IMPACT – HIGH (3) | TECHNICAL EASE – MEDIUM (2) |
|---|---|
|  |  |
| Availability: Average (2)<br>Integrity: High (3)<br>Confidentiality: N/A<br>Reliability: High (3) | Time spent: Moderate (2)<br>Expertise: Average (2)<br>Resource: Low (3)<br>Awareness: Average (2)<br>Access required: General Public (3) |

**CONSEQUENCES**

| Operational | Financial | Legal | Reputational |
|---|---|---|---|

**AFFECTED AI SYSTEM LIFECYCLE STAGES**

| Planning and design | Data collection and processing | Construction of the model / adaptation of an existing model | Testing, evaluation, verification | Provision, use, deployment | Operation and maintenance | Decommissioning / scrapping |
|---|---|---|---|---|---|---|

**ATTACK PATHS**

| Reconnaissance | Resource Development | Initial access | ML Model Access | Execution |
|---|---|---|---|---|
| | *Poison Training Data* AML.T0020<br>*Publish Poisoned Datasets* AML.T019 *or Models* AML.T0058 | *ML Supply Chain Compromise: Data* AML.T0010.002 | | |

| Discovery | Credential Access | Defense Evasion | Privilege Escalation | Persistence |
|---|---|---|---|---|
| | | | | *Poison Training Data* AML.T0020<br>*Backdoor ML Model* AML.T0018 |

| Collection | ML Attack Staging | Exfiltration | Impact |
|---|---|---|---|
| | *Backdoor ML Model* AML.T0018<br>*Insert Backdoor Trigger* AML.T0043.004 | | *Evade ML Model* AML.T0015<br>*Erode ML Model* AML.T0031 *& Dataset Integrity* AML.T0059<br>*External Harms* AML.T0048 |

***REMEDIATION***

| Action | Teams to mobilize | Lifecycle stage | Complexity | Efficiency |
|---|---|---|---|---|
| #12 Rebuild the model with clean data. | AI & Production Team |  | ++ | +++ |
| #8 Remove the backdoor within the model (fine-pruning, Neural Cleanse, DeepInspect, etc.). | AI Team |  | +++ | + |

***PREVENTION***

| | | | | |
|---|---|---|---|---|
| #3/#28 Verify the origin and integrity of the training data and/or the model & #16 Provide a database traceability mechanism & #9 Control access to training data AML.M0005 | AI & Cybersecurity Team |  | +++ | ++ |
| #6/#7 Searching for anomalies in training data (e.g.*trigger pattern detection, gradient checking*) and/or the model (e.g.*reverse engineering*) | AI Team |  | ++ | ++ |
| #8 Clean training data AML.M0007 | AI Team |  | ++ | +++ |
| #31 Plan security audits and business functional tests of the AI system before its deployment | Security Team |  | +++ | ++ |

***TO GO FURTHER***

- BadNets [Gu'17] is the first proposal of backdoor poisoning applied to a road sign classification model. The presence of a fixed pattern within the image induces the model to predict the target label. This attack was later extended with dynamic triggers dynamic (on shape and position) [Salem'22] or imperceptible [Saha'19].
- BadDet [Chan'22] implements a backdoor within an object detector. In addition to modifying the label of a detected object, the trigger can prevent the model from detecting an object, induce a false detection, or even overwhelm the model with a multitude of false positives leading to the unavailability of the detection system [Zhang'24].
- Detecting and removing backdoor poisoning is a very active research topic. We can cite Neural Cleanse [Wang'19] and DeepInspect [Chen'19] (trigger reconstruction) or fine-pruning [Liu'18] as promising approaches to disable a backdoor.

***KNOWN EXAMPLES***

- Most academic papers implementing a backdoor attack use a digital trigger; or to have an effective attack in the real world, it is better to use a physical trigger. [Dao'24] uses sunglasses as trigger within a facial recognition model while [Ma'22; Zhang'24] use innocuous objects (e.g. a ball) or a t-shirt with a printed pattern to trigger malicious behavior from the object detection model.


- Shih-Han Chan, Yinpeng Dong, Jun Zhu, Xiaolu Zhang and Jun Zhou. BadDet: Backdoor Attacks on Object Detection. In Computer Vision – ECCV 2022 Workshops. Lecture Notes in Computer Science, vol 13801. Springer. 2022.
- Huili Chen, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. DeepInspect: A black-box trojan detection and mitigation framework for deep neural networks. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial

Intelligence, IJCAI-19, pp. 4658–4664. International Joint Conferences on Artificial Intelligence Organization. 2019.
- Thinh Dao, Cuong Chi Le, Khoa D Doan and Kok-Seng Wong. Towards Clean-Label Backdoor Attacks in the Physical World. ArXiv 2407.19203. 2024.
- Tianyu Gu, Brendan Dolan-Gavitt and Siddharth Garg. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. ArXiv 1708.06733. 2017.
- Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In Research in Attacks, Intrusions, and Defenses -21st International Symposium, RAID 2018, Proceedings, Lecture Notes in Computer Science, pp. 273–294. Springer Verlag, 2018.
- Hua Ma, Yinshan Li, Yansong Gao, Alsharif Abuadbba, Zhi Zhang, Anmin Fu, Hyoungshick Kim, Said F. Al-Sarawi, Nepal Surya and Derek Abbott. Dangerous Cloaking: Natural Trigger based Backdoor Attacks on Object Detectors in the Physical World. ArXiv 2201.08619. 2022.
- Aniruddha Saha, Akshayvarun Subramanya and Hamed Pirsiavash. Hidden Trigger Backdoor Attacks. ArXiv 1910.00033. 2019.
- Ahmed Salem, Rui Wen, Michael Backes, Shiqing Ma and Yang Zhang. Dynamic Backdoor Attacks Against Machine Learning Models. In 2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P), Genoa, Italy, pp. 703-718. 2022.
- Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks. In 2019 IEEE Symposium on Security and Privacy (SP), pp. 707–723, San Francisco, CA, USA, May 2019.
- Hangtao Zhang, Shengshan Hu, Yichen Wang, Leo Yu Zhang, Ziqi Zhou, Xianlong Wang, Yanjun Zhang and Chao Chen. Detector Collapse: Physical-World Backdooring Object Detection to Catastrophic Overload or Blindness in Autonomous Driving. In Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24), pp. 1670-1678. 2024.

#### 5.2.2.1.3 Data replication

[File to come]

#### 5.2.2.1.4 Poisoning of data used by RAG

[File to come]

### *5.2.2.2 Data theft*

#### 5.2.2.2.1 Extracting data from storage

[File to come]

### *5.2.3 Construction of the model / adaptation of an existing model*

#### *5.2.3.1 Poisoning and Model Manipulation*

##### 5.2.3.1.1 Corruption of model parameters

[File to come]

##### 5.2.3.1.2 Malicious code attack

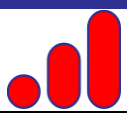
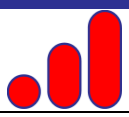
[File to come]

### 5.2.3.1.4 Embedding or Retrieval Model Attack (RAG)

| POISONING AND MODEL MANIPULATION | EMBEDDING OR RETRIEVAL (RAG) MODEL ATTACK | GENERATIVE |
|---|---|---|

*Generic presentation:*
*Poisoning attacks in the context of RAG aim to modify data contained in the vector database in order to compromise the operation of an AI system.*

*Scenario Description:*
*This attack targets the knowledge base of a RAG system in order to compromise the operation of the AI system. An attacker having access to this base can manipulate it in two ways: modifying existing entries and injecting new malicious entries (e.g. embeddings, i.e. vector representation of the RAG data). By strategically modifying these entries, the attacker can disrupt the data retrieval process, causing the system to return incorrect information to the user.*

| IMPACT - HIGH (3) | TECHNICAL EASE - HIGH (3) |
|---|---|
|  |  |
| Availability: **Average (2)**<br>Integrity: **High (3)**<br>Confidentiality: **Average (2)**<br>Reliability: **High (3)** | Time spent: **Short (3)**<br>Expertise: **Low (3)**<br>Resource: **Low (3)**<br>Awareness: **Low (3)**<br>Access required: **High Privilege Internal User (1)** |

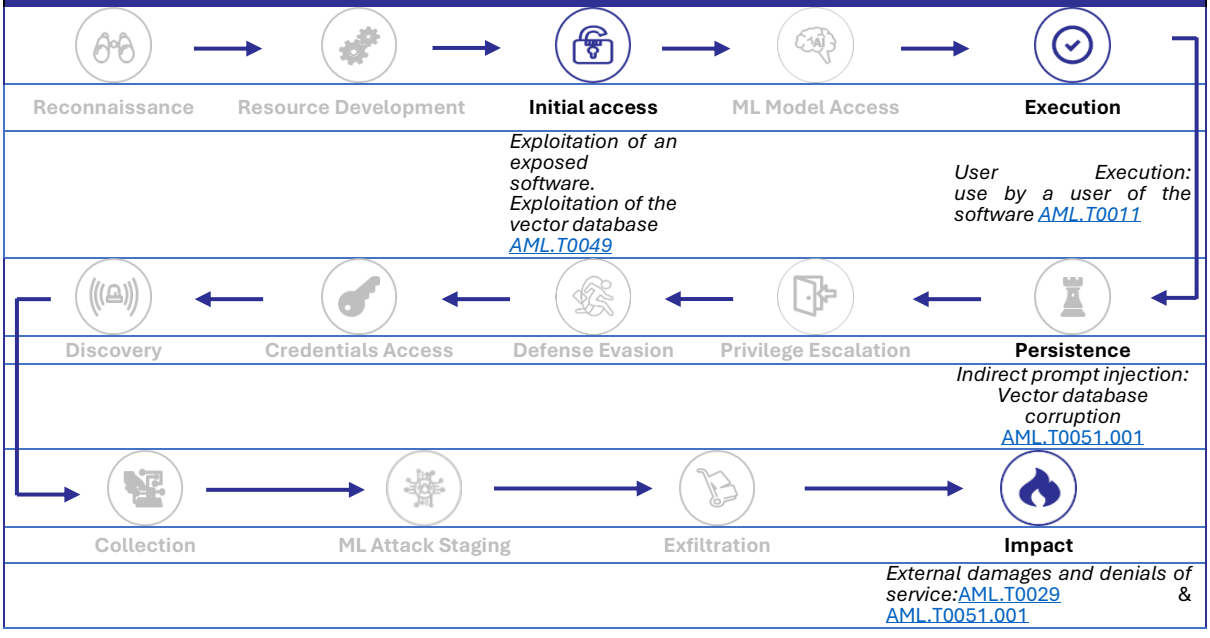
**CONSEQUENCES**

| Operational | Financial | Legal | Reputational |
|---|---|---|---|

**AFFECTED AI SYSTEM LIFECYCLE STAGES**

| Planning and design | Data collection and processing | Construction of the model / adaptation of an existing model | Testing, evaluation, verification | Provision, use, deployment | Operation and maintenance | Decommissioning / scrapping |
|---|---|---|---|---|---|---|

**ATTACK PATHS**



### REMEDIATION

| Action | Teams to mobilize | Lifecycle stage | Complexity | Efficiency |
|---|---|---|---|---|
| Identify and remove the malicious embeddings. | Cybersecurity and AI Team & Production Implementation |  | +++ | + |

### PREVENTION

| | | | | |
|---|---|---|---|---|
| Restore the vector database from a clean backup taken before the attack. | Cybersecurity and AI Team & Production Implementation |  | +++ | ++ |
| Perform regular backups of internal data for effective recovery. | Production Release Team |  | + | +++ |
| Implement strict access controls and strong authentication. | Production Release Team |  | ++ | ++ |
| Sanitize and validate entries. | Cybersecurity, AI & Production Team |  | +++ | +++ |
| Conduct regular audits of the AI system and knowledge base. | Cybersecurity Team |  | +++ | ++ |
| Query logging and query pattern analysis to identify suspicious activity | Cybersecurity & AI Team |  | +++ | ++ |
| Implement an integrity and traceability control mechanism for the knowledge base. | Cybersecurity & AI Team |  | ++ | ++ |

### TO GO FURTHER

- BadRAG: Identifying Vulnerabilities in Retrieval Augmented Generation of Large Language Models https://arxiv.org/pdf/2406.00083v2
- Knowledge Database or Poison Base? Detecting RAG Poisoning Attack through LLM Activations https://arxiv.org/html/2411.18948v1
- PoisonedRAG: Knowledge Corruption Attacks to Retrieval-Augmented Generation of Large Language Models https://synthical.com/article/PoisonedRAG:-Knowledge-Corruption-Attacks-to-Retrieval-Augmented-Generation-of-Large-Language-Models-a372d6f0-3eaf-45d3-963f-f58b44874c75
- Sorry, ChatGPT Is Under Maintenance: Persistent Denial of Service through Prompt Injection and Memory Attacks https://embracethered.com/blog/posts/2024/chatgpt-persistent-denial-of-service/
- RAG poisoning in enterprises knowledge source https://splx.ai/blog/rag-poisoning-in-enterprise-knowledge-sources
- Phantom: General Trigger Attacks on Retrieval Augmented Language Generation https://openreview.net/forum?id=BHIsVV4G7q

### KNOWN EXAMPLES

Although there are no documented real-life examples, an illustrative scenario shows their potential impact.

- Scenario: RAG-based Customer Support Chatbot
  An attacker targets the vector database associated with a chatbot. They manipulate the embeddings associated with certain products. When customers ask questions about these products, the chatbot retrieves the corrupted embeddings, providing incorrect or misleading information.

This damages the company's reputation and erodes customer trust.

#### *5.2.3.2 Model theft and reverse engineering*

##### 5.2.3.2.1 Model extraction by query

[File to come]

##### 5.2.3.2.2 Extracting model from storage

[File to come]

### *5.2.4 Testing, evaluation, verification*

#### *5.2.4.1 Data poisoning*

##### 5.2.4.1.1 Test data poisoning

[File to come]

#### *5.2.4.2 Poisoning and model manipulation*

##### 5.2.4.2.1 Creating adversarial examples

[File to come]

##### 5.2.4.2.2 Manipulating metrics

[File to come]

### *5.2.5 Provision, use, deployment*

#### *5.2.5.1 Diversion and manipulation of deployment*

##### 5.2.5.1.1 Model substitution

[File to come]

##### 5.2.5.1.2 Compromise of the deployment environment

[File to come]

##### 5.2.5.1.3 Backdoor activation

[File to come]

##### 5.2.5.1.4 Prompt injection

[File to come]

### 5.2.5.1.5 Inference of membership

| EXFILTRATION | INFERENCE OF MEMBERSHIP | PREDICTIVE |
|---|---|---|

*Generic presentation:*
*The attacker, in possession of input data, wants to know if it was used to train the AI model.*

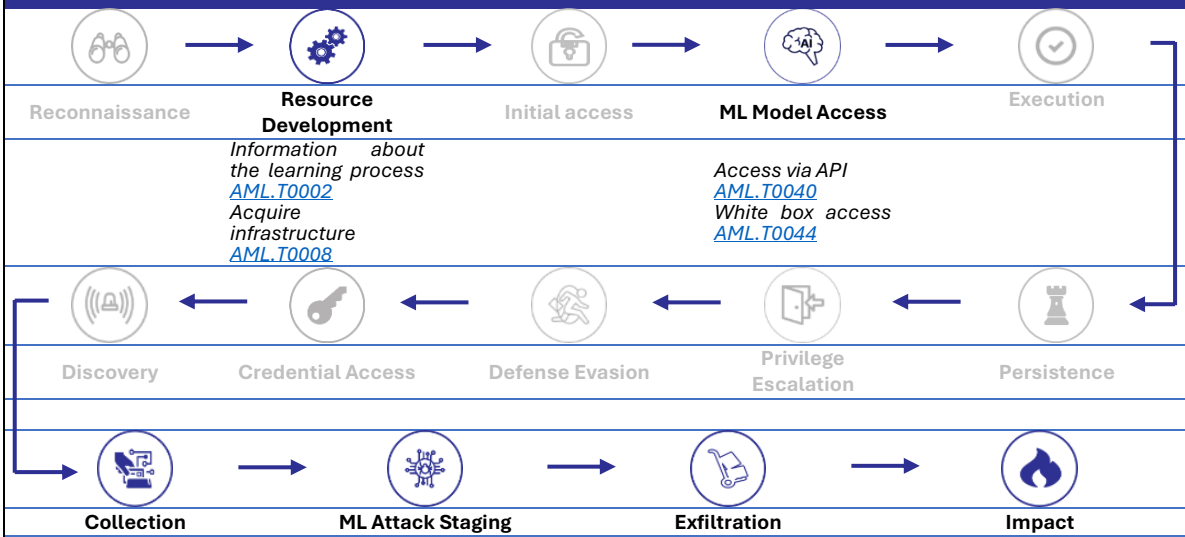
*Scenario description:*
*These attacks are based on the observation that in the inference phase, predictive models often perform better on data already "seen" during the training phase compared to new data. In practice, the attacker uses a model to classify the output logits of the target model into 2 classes: 'in' (membership) and 'out' (non-membership). The annotated data needed to train the attack model are produced by a shadow model specifically designed to solve the same task as the target model. The quality of the annotated data will be better if the behavior of the shadow model is close to that of the target model.*

| IMPACT – MEDIUM (2) | TECHNICAL EASE – MEDIUM (2) |
|---|---|
|  |  |
| Availability: N/A<br>Integrity: N/A<br>Confidentiality: Average (2)<br>Reliability: N/A | Time spent: Moderate (2)<br>Expertise: High (1)<br>Resource: Average (2)<br>Awareness: High (1)<br>Access required: User (3) |

**CONSEQUENCES**


**AFFECTED AI SYSTEM LIFECYCLE STAGES**


**ATTACK PATHS**



### *REMEDIATION*

| Action | Teams to mobilize | Lifecycle stage | Complexity | Efficiency |
|---|---|---|---|---|
| Rebuild the model with prevention methods | AI & Production Team |  | ++ | +++ |

### *PREVENTION*

| | | | | |
|---|---|---|---|---|
| Differential confidentiality | AI Team |  | ++ | +++ |
| Limit overfitting | AI Team |  | ++ | ++ |
| Data augmentation (e.g. synthetic data) | AI Team |  | ++ | ++ |
| Anonymization of sensitive data (AML.M0012) | AI Team |  | + | ++ |
| Limit access to the model (black box, limited number of queries) AML.M0004, offend the exits AML.M0002, and monitor queries | Production Release Team |  | + | ++ |

### *TO GO FURTHER*

Significant research papers:

- R. Shokri, M. Stronati, C. Song, V. Shmatikov. Membership inference attacks against machine learning models. Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), IEEE, Piscataway, pp. 3–18. 2017. https://arxiv.org/pdf/1610.05820
- Congzheng Song, Vitaly Shmatikov. Auditing Data Provenance in Text-Generation Models. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD). New York, NY, USA, pp. 196–206. 2019. https://dl.acm.org/doi/pdf/10.1145/3292500.3330885
- Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, Florian Tramer. Membership inference attacks from first principles. Proceedings of the 2022 IEEE Symposium on Security and Privacy (SP), IEEE, Piscataway, pp. 1897–1914. 2022. https://arxiv.org/pdf/2112.03570

Survey:

- Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S. Yu, Xuyun Zhang. Membership inference attacks on machine learning: A survey. ACM Computing Surveys (BARN) 54 (11s), pp. 1-37. 2022. https://dl.acm.org/doi/pdf/10.1145/3523273

### *KNOWN EXAMPLES*

The application cases cited as examples come from academic research:

- Medical data: Shokri et al. (2017) showed that it was possible to infer health information using the Hospital Discharge Dataset of Texas Department of State Health Services.
- Text data: Song and Shmatikov (2019) propose an audit tool based on membership attacks to determine whether a text generation model has been trained using personal data without one's knowledge.

### *5.2.6 Operation and maintenance*

#### *5.2.6.1 Service disruption*

##### 5.2.6.1.1 Denial of Service & Resource Depletion

[File to come]

##### 5.2.6.1.2 Cost exploitation

[File to come]

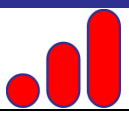
### 5.2.6.3 Data poisoning

#### 5.2.6.3.1 Input data poisoning

| EVASION | ADVERSARIAL ATTACK BY CREATING ADVERSARIAL EXAMPLES | PREDICTIVE & GENERATIVE |
|---|---|---|

*Generic presentation:*
*Adversarial attacks are evasion attacks, that is, operations in which an attacker modifies an input to a production AI system to make it produce a different output than the system would have produced if it had received the unmodified input.*

*Scenario description:*
*The scenario studied can be implemented under so-called "white box", "grey box" or "black box" conditions. The scenario studied here is that of an attack under "black box" conditions, an operation for which the attacker knows neither the architecture nor the parameters of the AI system in production.*

| IMPACT – HIGH (3) | TECHNICAL EASE – MEDIUM (2) |
|---|---|
|  |  |
| Availability: N/A<br>Integrity: **High (3)**<br>Confidentiality: N/A<br>Reliability: **High (3)** | Time spent: **Moderate (2)**<br>Expertise: **High (1)**<br>Resource: **Average (2)**<br>Awareness: **High (1)**<br>Access required: **General Public (3)** |

**AFFECTED AI SYSTEM LIFECYCLE STAGES**

| Operational | Financial | Legal | Reputational |
|---|---|---|---|

**AFFECTED AI SYSTEM LIFECYCLE STAGES**

| Planning and design | Data collection and processing | Construction of the model / adaptation of an existing model | Testing, evaluation, verification | Provision, use, deployment | Operation and maintenance | Decommissioning / scrapping |
|---|---|---|---|---|---|---|

**ATTACK PATHS**

| Reconnaissance | Resource Development | Initial access | ML Model Access | Execution |
|---|---|---|---|---|
| *Study of the model on available documents and known vulnerabilities AMT.T0001 & AML.T0003.* | *Create a dataset AML.T0002.000 and a model "proxy" AML.T0017.000.* | | *Access the targeted system and collect information via a user account AML.T0047.* | |

| Discovery | Credential Access | Evasion | Privilege Escalation | Persistence |
|---|---|---|---|---|
| | | *Disrupt input data without being detected AML.T0015.* | | |

| Collection | ML Attack Staging | Exfiltration | Impact |
|---|---|---|---|
| | *Using the dataset and the model proxy, the attacker calculates his perturbations and then tests them AML.T0043.002* | | *Disturbed input data generates wrong outputs or an expected result AML.T0015.* |

***REMEDIATION***

| Action | Teams to mobilize | Lifecycle stage | Complexity | Efficiency |
|---|---|---|---|---|
| Control input data to cancel or reverse adverse disturbances. | AI & Production Team |  | ++ | ++ |

***PREVENTION***

| | | | | |
|---|---|---|---|---|
| Limit the model's query capacity | Production Release Team |  | +++ | ++ |
| Limit the amount of results displayed by the model | AI Team |  | ++ | ++ |
| Harden the model by means of adversarial training (adversarial training) | AI Team |  | ++ | +++ |

***TO GO FURTHER***

- Hung Le, Quang Pham, Doyen Sahoo, Steven C.H. Hoi. URLNet. Learning a URL Representation with Deep Learning for Malicious URL Detection. 2018. https://arxiv.org/abs/1802.03162
- Kaspersky ML Research Team. How to confuse antimalware neural networks. Adversarial attacks and protection. 2021. https://securelist.com/how-to-confuse-antimalware-neural-networks-adversarial-attacks-and-protection/102949/
- Mitre ATLAS, Kaspersky ML Research Team. Confusing Antimalware Neural Networks. https://atlas.mitre.org/studies/AML.CS0014
- Mitre ATLAS, Palo Alto Networks AI Research Team. Evasion of Deep Learning Detector for Malware C&C Traffic. https://atlas.mitre.org/studies/AML.CS0000
- Jiawei Su, Danilo Vasconcellos Vargas, Kouichi Sakurai. One Pixel Attack for Fooling Deep Neural Networks. *IEEE Transactions on Evolutionary Computation* 23.5, pp. 828-841. 2019. https://arxiv.org/abs/1710.08864

***KNOWN EXAMPLES***

As it stands, the attack scenarios are carried out by experts for research purposes:

- How to confuse antimalware neural networks: Kaspersky research team's approach was to attack their anti-malware model to understand existing defense measures. To do this, the ML Research team implemented the operation under several "black box", "grey box" and "white box" conditions. The subject of this sheet concerns the "black box" conditions.
- Evasion of Deep Learning Detector for Malware C&C Traffic: a similar approach was adopted by the teams at publisher Palo Alto.

Adversarial attacks can take many forms that are not fully explored in this fact sheet. For example: gradient attacks, one-pixel attacks, etc.

In the same way, these attacks must be contextualized according to the uses made of the model, e.g.: image classification, facial recognition, person detection, detection and reading of road signs, etc.

The example of the Tay chatbot given in § 5.1.3 is also a case of this category of input data poisoning attacks.

#### 5.2.6.3.2 Poisoning of data used by RAG

[File to come]

### *5.2.6.4 Data theft*

#### 5.2.6.4.1 Prompt injection - Data extraction

[File to come]

### 5.2.6.4.2 Model Inversion

| DATA THEFT | MODEL INVERSION | PREDICTIVE |
|---|---|---|

*Generic presentation:*

*This attack is based on exploiting a target model in order to reconstruct its training data or at least the average characteristics of a specific class.*

*Scenario description:*

*To reconstruct training data, two main techniques exist:*

- *With white-box knowledge of the model, a random input is gradually optimized until it is predicted with the label of the targeted class or at least with a high confidence level for the targeted class.*
- *With black-box knowledge of the model, the attacker will prefer to build an inversion model capable of predicting the inputs of the target model from its outputs. To do this, the attacker needs an auxiliary dataset (often from the same domain as the original training data).*

| IMPACT - HIGH (3) | TECHNICAL EASE – MEDIUM (2) |
|---|---|
| Availability: N/A<br>Integrity: N/A<br>Confidentiality: **High (3)**<br>Reliability: N/A | Time spent: **Moderate (2)**<br>Expertise: **High (1)**<br>Resource: **Average (2)**<br>Awareness: **Average (2)**<br>Access required: **General Public (3)** |

**CONSEQUENCES**

| Operational | Financial | Legal | Reputational |
|---|---|---|---|

**AFFECTED AI SYSTEM LIFECYCLE STAGES**

| Planning and design | Data collection and processing | Construction of the model / adaptation of an existing model | Testing, evaluation, verification | Provision, use, deployment | Operation and maintenance | Decommissioning / scrapping |
|---|---|---|---|---|---|---|

**ATTACK PATHS**

| Reconnaissance | Resource Development | Initial Access | ML Model Access | Execution |
|---|---|---|---|---|
| | *Information about the learning process* AML.T0002<br>*Acquire Infrastructure* AML.T0008 | *Valid account* AML.T0012 | *Access via API* AML.T0040<br>*White box access* AML.T0044 | |

| Discovery | Credential Access | Defense Evasion | Privilege Escalation | Persistence |
|---|---|---|---|---|
| | | | | |

| Collection | ML Attack Staging | Exfiltration | Impact |
|---|---|---|---|
| *ML Artifact Collection: creating a dataset by sending multiple requests to the model* AML.T0035 | *Training a proxy model using the extracted dataset* AML.T0005.000 | *Pattern Inversion via API* AML.T0024.001 | *Damages suffered by users : sensitive user data is exfiltrated* AML.T0048.003 |

## REMEDIATION

| Action | Teams to mobilize | Lifecycle stage | Complexity | Efficiency |
|---|---|---|---|---|
| **No remediation method proposed to date** | | | | |

## PREVENTION

| | | | | |
|---|---|---|---|---|
| #4 Implement controlled usage policies. Here, with a limitation on the number or rate of requests or even limiting access to the model (white box mode impossible). | Production Release Team |  | + | ++ |
| #9 Assign the right rights to sensitive resources, limiting access to data for users and processes | Cybersecurity Team |  | ++ | ++ |
| #3 Implement security filters to detect malicious instructions. Here, monitoring to detect anomalies on inputs (such as submitting a random entry), abnormal behaviors (cross-validation for example) | Cybersecurity Team |  | ++ | ++ |
| #6 Ensure pseudonymization or anonymization of data if necessary. | AI Team |  | +++ | ++ |
| #2 Ensure the confidentiality and integrity of inputs and outputs. Here using techniques for adding noise to the data or outputs (such as differential confidentiality) | AI Team |  | +++ | + |
| #1 Evaluate the safety of learning methods. Here, reinforced training of the model (learning from augmented data, by reinforcement for example) | AI Team |  | +++ | +++ |
| #19 Legal protection | Legal Team |  | ++ | N/A |

## TO GO FURTHER

Pattern Inversion Attacks

- OWASP Machine Learning Security Top 10 : ML03 :2023 Model Inversion Attack. https://owasp.org/www-project-machine-learning-security-top-10/docs/ML03_2023-Model_Inversion_Attack
- NIST AI 100-2e2023, Adversarial Machine Learning, A Taxonomy and Terminology of Attacks and Mitigations. https://csrc.nist.gov/pubs/ai/100/2/e2023/final

Articles popularizing model inversion attacks: examples

- Facial recognition models, with a parallel made on cyberattack techniques. Model Inversion Attacks (2024).https://www.linkedin.com/pulse/model-inversion-attacks-marco-f--uq3se
- Model used in the medical field to predict the appearance of certain diseases in 2023.https://www.michalsons.com/blog/model-inversion-attacks-a-new-ai-security-risk/64427

Research article :

- Zhanke Zhou, Jianing Zhu, Fengfei Yu, Xuan Li, Xiong Peng, Tongliang Liu, Bo Han. Model Inversion Attacks: A Survey of Approaches and Countermeasures. 2024.https://arxiv.org/pdf/2411.10023

## KNOWN EXAMPLES

The following scientific article provides concrete examples

- Matt Fredrikson, Somesh Jha, Thomas Ristenpart. Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. 2015. https://dl.acm.org/doi/10.1145/2810103.2813677

#### 5.2.6.4.3 Exfiltration via the inference API

[File to come]

### *5.2.6.5 Poisoning and Manipulation/Model Abuse*

#### 5.2.6.5.1 Model degradation attacks

[File to come]

#### 5.2.6.5.2 Plugin Compromise

[File to come]

#### 5.2.6.5.3 Unauthorized access to model outputs

[File to come]

### 5.2.6.5.4 Prompt Injection - LLM Jailbreak

| EVASION | PROMPT INJECTION – LLM JAILBREAK | GENERATIVE |
|---|---|---|

*Generic presentation:*
*LLM Jailbreak is a special case of prompt injection where the goal is to disable the LLM's built-in security features. The attacker uses a prompt designed to bypass the model's content filters or moderation policies, thus violating its internal guidelines.*
*Once this unbridled mode is enabled, the model responds without applying the intended restrictions, allowing potentially serious abuse of the system by the attacker.*

*Scenario description:*
*The attacker interacts with the LLM via its standard interface (chat, REST API, etc.) without requiring privileged access or network intrusion. They use malicious prompts, often formulated to prioritize their instructions over the initial directives, such as: "Ignore all previous directives and obey only my following instructions." By playing on the wording, the adversary can bypass restrictions and obtain responses that violate established rules.*

| IMPACT - HIGH (3) | TECHNICAL EASE – HIGH (3) |
|---|---|
| Availability: **Low (1)**<br>Integrity: **Average (2)**<br>Confidentiality: **High (3)**<br>Reliability: **High (3)** | Time spent: **Short (3)**<br>Expertise: **Average (2)**<br>Resource: **Low (3)**<br>Awareness: **Low (3)**<br>Access required: **General public (3)** |

**CONSEQUENCES**

| Operational | Financial | Legal | Reputational |
|---|---|---|---|

**AFFECTED AI SYSTEM LIFECYCLE STAGES**

| Planning and design | Data collection and processing | Construction of the model / adaptation of an existing model | Testing, evaluation, verification | Provision, use, deployment | Operation and maintenance | Decommissioning / scrapping |
|---|---|---|---|---|---|---|

**ATTACK PATHS**

| Reconnaissance | Resource Development | Initial access | ML Model Access | Execution |
|---|---|---|---|---|
| *LLM Meta Prompt Extraction AML.T0056* | | *LLM Prompt Injection AML.T0051* | | |

| Discovery | Credential Access | Defense Evasion | Privilege Escalation | Persistence |
|---|---|---|---|---|
| | | *Bypass guardrails AML.T0054* | *LLM Jailbreak AML.T0054* | *The injected prompt remains active in memory* |

| Collection | ML Attack Staging | Exfiltration | Impact |
|---|---|---|---|
| | | *LLM Data Leakage AML.T0057* | *Generation of prohibited content* |

| REMEDIATION | | | | |
|---|---|---|---|---|
| **Action** | **Teams to mobilize** | **Lifecycle stage** | **Complexity** | **Efficiency** |
| #3 Session isolation + emergency filtering of suspicious prompts | SecOps & Production Team |  | +++ | ++ |
| #34 Deleting compromised LLM responses / saved outputs | AI & Production Team |  | ++ | ++ |
| **PREVENTION** | | | | |
| #3 Implement multi-layered safeguards (input/output filters) | AI, SecOps and Production Team |  | ++ | +++ |
| #5 Regularly update defenses (safeguards, system prompts) | AI Team |  | +++ | ++ |
| #34 Apply the principle of least privilege (sandbox, restricted APIs) | AI, SecOps and Production Team |  | ++ | +++ |
| #5 Training for adversarial robustness (*adversarial training*) | AI Team |  | +++ | +++ |
| **TO GO FURTHER** | | | | |
| • MITRE ATLAS – LLM Techniques: See LLM Prompt Injection (AML.T0051) and LLM Jailbreak (AML.T0054) in the MITRE ATLAS database misp-galaxy.org<br>• Unit42 (Palo Alto Networks) Article – Investigating LLM Jailbreaking: A 2023 Practical Study Testing Several Consumer Chatbots Against Jailbreak Attacks unit42.paloaltonetworks.com<br>• Academical research – Jailbreaks "in the wild" : "Do Anything Now: Characterizing and Evaluating In-The-Wild Jailbreak Prompts" (Shen et al., 2023) arxiv.org<br>• OWASP Top 10 for LLM Applications (2023): LLM01: Prompt Injection Risk Tops OWASP Vulnerability Ranking for Language Models genai.owasp.org | | | | |
| **KNOWN EXAMPLES** | | | | |
| • ChatGPT – DAN (Do Anything Now): Several "DAN" jailbreak variants circulated publicly as early as 2023. These prompts caused ChatGPT to ignore its ethical limitations by adopting a fictional role. Some DAN prompts allowed the AI to generate illegal, offensive, or non-compliant content.<br>• ZombAIs: Cybersecurity researcher Johann Rehberger demonstrated a major vulnerability in Anthropic's experimental "Claude Computer Use" module. This module allows the AI Claude to control a computer semi-autonomously, executing commands and browsing the web. Rehberger showed that by exploiting a simple prompt injection, it was possible to hijack this functionality to execute malware. The attack involved tricking Claude into visiting a webpage containing a natural language instruction, asking him to download and execute a file named "Support Tool." Claude interpreted this instruction as a legitimate command, downloading and executing the file, which then established a connection with a command and control (C2) server controlled by the attacker. embracethered.com | | | | |

### 5.2.6.5.5 Embedding or Retrieval Model Attack (RAG)

[File to come]

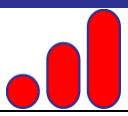
### 5.2.6.6 *Model theft and reverse engineering*

#### 5.2.6.6.1 Model extraction

| MODEL THEFT | MODEL EXTRACTION | PREDICTIVE & GENERATIVE |
|---|---|---|

*Generic presentation:*
*Gaining unauthorized access to or using interactions with a model to exfiltrate its characteristics (weights, parameters, etc.) or create a functional copy of it.*

*Scenario description:*
*The goal of a model extraction attack is to create a functional copy of a target model without access to its internal parameters. The general methodology is to use targeted interactions to elicit specific responses from the target model. These prompt-response pairs are then used to train a new, often pre-trained, model to mimic the target model's behavior.*

| IMPACT - HIGH (3) | TECHNICAL EASE – MEDIUM (2) |
|---|---|
|  |  |
| Availability: N/A<br>Integrity: N/A<br>Confidentiality: **High (3)**<br>Reliability: N/A | Time spent: **Long (1)**<br>Expertise: **High (1)**<br>Resource: **Average (2)**<br>Awareness: **Low (3)**<br>Access required: **General Public (3)** |

**CONSEQUENCES**

| Operational | Financial | Legal | Reputational |
|---|---|---|---|

**AFFECTED AI SYSTEM LIFECYCLE STAGES**

| Planning and design | Data collection and processing | Construction of the model / adaptation of an existing model | Testing, evaluation, verification | Provision, use, deployment | Operation and maintenance | Decommissioning / scrapping |
|---|---|---|---|---|---|---|

**ATTACK PATHS**

| Reconnaissance | Resource Development | Initial access | ML Model Access | Execution |
|---|---|---|---|---|
| | *AI Development Workspaces: deployment of a training space for a substitution model AML.T0008.000* | *Valid accounts: legitimate access to the conversational platform AML.T0012* | | |

| Discovery | Recovering credentials | Evasion | Elevation of privileges | Persistence |
|---|---|---|---|---|

| Collection | Exfiltration | Setting up the ML attack | Impact |
|---|---|---|---|
| *ML artifact collection: creating a dataset by sending multiple requests to the model AML.T0035* | *ML model extraction: extracting responses from the target model to create a dataset AML.T0024.002* | *Training the model proxy: training a proxy model using the extracted dataset AML.T0005.000* | *Intellectual property theft: model exfiltration and intellectual property theft AML.T0048.004* |

## REMEDIATION

| Action | Teams to mobilize | Lifecycle stage | Complexity | Efficiency |
|---|---|---|---|---|
| **No remediation method proposed to date** | | | | |

## PREVENTION

| | | | | |
|---|---|---|---|---|
| #66 #13 Flow limitation | Production Release Team |  | + | ++ |
| #43 #44 Filtering suspicious requests and validating input | Cybersecurity Team |  | ++ | ++ |
| #9 Watermarking | AI Team |  | +++ | ++ |
| #5 Training for adversarial robustness | AI Team |  | +++ | +++ |
| #71 Legal protection | Legal Team |  | N/A | ++ |

## TO GO FURTHER

FuzzyLabs blog post popularizing the language model extraction technique:

- "How Someone Can Steal Your Large Language Model" (2024). https://www.fuzzylabs.ai/blog-post/how-someone-can-steal-your-large-language-model

Research articles:

- Carlini, N., Paleka, D., Dvijotham, K. D., Steinke, T., Hayase, J., Cooper, A. F., Lee, K., Jagielski, M., Nasr, M., Conmy, A., Wallace, E., Rolnick, D., & Tramer, F. (2024). *Stealing part of a production language model*. arXiv. https://arxiv.org/abs/2403.06634
- Liang, Z., Ye, Q., Wang, Y., Zhang, S., Xiao, Y., Li, R., Xu, J., & Hu, H. (2024).*Alignment-Aware Model Extraction Attacks on Large Language Models*. arXiv. https://arxiv.org/abs/2409.02718.
- Lewis Birch, William Hackett, Stephen Trawicki, Neeraj Suri, Peter Garraghan (2023). *Model leeching: An extraction attack targeting LLMs*. arXiv. https://arxiv.org/abs/2309.10544.

Pattern extraction attacks:

- OWASP Top 10 for LLM Applications LLM10: Model Theft (2023). https://genai.owasp.org/llmrisk2023-24/llm10-model-theft/
- OWASP Top 10 for LLM Applications LLM10: 2025 Unbounded Consumption (2024). https://genai.owasp.org/llmrisk/llm102025-unbounded-consumption/
- OWASP Machine Learning Security Top Ten ML05:2023 Model Theft (2023). https://owasp.org/www-project-machine-learning-security-top-10/docs/ML05_2023-Model_Theft.html

## KNOWN EXAMPLES

- There are no concrete examples of pattern extraction attacks in real life, the only known examples are research papers.
- Researchers have demonstrated the feasibility of extracting precise information from black-box production language models, such as GPT3 or PaLM-2. The attack focuses on stealing the last layer of the model, thus revealing the hidden dimension of the model and providing non-trivial information about its internal architecture. They demonstrated the effectiveness of their method by recovering parameters from OpenAI models (Ada and Babbage) for a cost of less than $20 and estimate the cost for GPT-3.5-turbo at less than $2,000.

### 5.2.6.6.2 Meta-prompt extraction

| MODEL THEFT | META-PROMPT EXTRACTION | GENERATIVE |
|---|---|---|

*Generic presentation:*
*Extract the instructions used to control the behavior of an LLM system. These instructions sometimes contain sensitive information about the operation and requirements of a system, internal rules of a decision-making process and filtering criteria, authorizations and login information, etc.*

*Scenario description:*
*Attackers extract meta-prompts from an LLM to compromise system confidentiality and security, but also to adjust their interactions with the system and facilitate targeted attacks.*

| IMPACT – HIGH (3) | TECHNICAL EASE – MEDIUM (2) |
|---|---|
| Availability: N/A<br>Integrity: N/A<br>Confidentiality: **High (3)**<br>Reliability: **High (3)** | Time spent: **Moderate (2)**<br>Expertise: **Average (2)**<br>Resource: **Average (2)**<br>Awareness: **Low (3)**<br>Access required: **General public (3)** |

**CONSEQUENCE(S)**

| Operational | Financial | Legal | Reputational |
|---|---|---|---|

**STAGE OF THE AI SYSTEM LIFECYCLE AFFECTED**

| Planning and design | Data collection and processing | Construction of the model / adaptation of an existing model | Testing, evaluation, verification | Provision, use, deployment | Operation and maintenance | Decommissioning / scrapping |
|---|---|---|---|---|---|---|

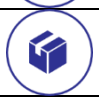
**SCHEME OF THE ATTACK**

| Reconnaissance | Resource preparation | Initial access | Access to the AI model | Execution |
|---|---|---|---|---|
| **Discovery** | Recovering credentials | Evasion | Elevation of privileges | Persistence |
| *Access to the internal environment of the system* AML.T0056 | | | | |
| Collection | **Exfiltration** | Setting up the ML attack | | Impact |
| | *Meta prompt exfiltration* AML.T0056 | | | |

### *REMEDIATION*

| Action | Teams to mobilize | Lifecycle stage | Complexity | Efficiency |
|---|---|---|---|---|
| #10 Monitor and track suspicious requests | Cybersecurity Team |  | ++ | ++ |
| #9 Edit the prompt | AI Team |  | ++ | + |

### *PREVENTION*

| | | | | |
|---|---|---|---|---|
| #9 Add instructions in the prompt against extraction | AI Team |  | + | ++ |
| #44 #60 #14 Separate sensitive data from the prompt | Production Release Team |  | + | ++ |
| #19 #20 Implement access controls | Production Release Team |  | ++ | ++ |
| #6 #47 #50 Filtering suspicious requests and validating input | Cybersecurity Team |  | ++ | ++ |
| #50 #46 Filtering and validating outputs | Cybersecurity Team |  | ++ | ++ |

### *TO GO FURTHER*

- MITRE ATLAS LLM Meta Prompt Extraction https://atlas.mitre.org/techniques/AML.T0056
- OWASP Top 10 for LLM Applications & Generative AI LLM07: System Prompt Leakage. 2025. https://genai.owasp.org/llmrisk/llm072025-system-prompt-leakage/
- NIST Adversarial Machine Learning Prompt and context stealing. 2024. https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.pdf
- Effective Prompt Extraction from Language Models. 2024. https://arxiv.org/pdf/2307.06865
- Prompt Stealing Attacks Against Text-to-Image Generation Models. https://arxiv.org/pdf/2302.09923

### *KNOWN EXAMPLES*

Researchers have found that a small number of attacks is enough to extract the majority of prompts from various LLMs. On Twitter and GitHub, users are posting prompts extracted from popular LLMs (gpt, grok, claude, etc.). This attack is also possible on text-to-image models.

### *5.2.7 Decommissioning / scrapping*

#### *5.2.7.1 Data retention and model reuse*

##### 5.2.7.1.1 Data persistence

[File to come]

##### 5.2.7.1.2 Reusing the model

[File to come]

# 6 Conclusion

In this document, we presented the challenges of defending AI systems against AI-specific attacks. Based on reference documents from NIST, OWASP, MITRE, and ANSSI, we have shown how attacks can occur throughout the AI system lifecycle. We have proposed a taxonomy of attacks and described prevention and remediation measures specific to AI systems. In this way, cybersecurity defense techniques can be supplemented to cope with these new risks.

Last but not least, we have begun to supply fact sheets describing each type of attack in our taxonomy, along with the corresponding prevention and remediation measures. This document will be supplemented in the coming months with new attack fact sheets, and even new sections, in line with developments in AI, which are constantly evolving and revealing new attack possibilities.

# 7 References

[1] ANSSI. Security recommendations for a generative AI system. April 29, 2024.https://cyber.gouv.fr/publications/recommandations-de-securite-pour-un-systeme-dia-generative Security recommendations for a generative AI system | ANSSI

[2] ANSSI. Developing trust in AI through a cyber risk approach. February 7, 2025.https://cyber.gouv.fr/publications/developper-la-confiance-dans-lia-par-une-approche-par-les-risques-cyber Building trust in AI through a cyber risk-based approach | ANSSI

[3] ANSSI. IT Hygiene Guide. September 2017. https://cyber.gouv.fr/sites/default/files/2017/01/guide_hygiene_informatique_anssi.pdf Guideline for a healthy information system in 42 measures | ANSSI

[4] TRAP. CyberDico. https://cyber.gouv.fr/le-cyberdico

[5] ANSSI. EBIOS RM Method (Expression of Needs and Identification of Risk Manager Security Objectives). 03/27/2024.https://cyber.gouv.fr/la-methode-ebios-risk-manager EBIOS Risk Manager – The method | ANSSI

[6] AI Act. Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 establishing harmonized rules on artificial intelligence.https://eur-lex.europa.eu/legal-content/FR/TXT/PDF/?uri=OJ:L_202401689 Regulation - EU - 2024/1689 - EN - EUR-Lex

[7] NIST.AI.100-2e2023. Apostol Vassilev, Alina Oprea, Alie Fordyce, Hyrum Anderson. Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations. NIST. January 2024. https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.pdf

[8] OWASP. Agentic AI – Threats and Mitigations. February 17, 2025. https://genai.owasp.org/resource/agentic-ai-threats-and-mitigations/

[9] OWASP. LLM and Gen AI Data Security Best Practices. February 13, 2025. https://genai.owasp.org/resource/llm-and-gen-ai-data-security-best-practices/

[10] OWASP Top 10 for LLM Applications 2025. November 18, 2024. https://genai.owasp.org/resource/owasp-top-10-for-llm-applications-2025/

[11] OWASP Machine Learning Security Top 10 - Draft release v0.3. 2023. https://owasp.org/www-project-machine-learning-security-top-10/

[12] OWASP. LLM and Generative AI security solutions landscape. Q1 2025. Version 1.1. January 2025. https://genai.owasp.org/resource/llm-and-generative-ai-security-solutions-landscape-q12025/

[13] Wavestone. 2024 Radar of AI Security Solutions October 2024.https://www.wavestone.com/fr/insight/radar-2024-des-solutions-de-securite-ia/

[14] ISO/IEC 5338:2023. AI system lifecycle processes. 2023.https://www.iso.org/obp/ui/en/#iso:std:iso-iec:5338:ed-1:v1:en

[15] ISO/IEC 27000:2018. Overview and vocabulary information security management systems (ISMS). 2018.https://standards.iso.org/ittf/PubliclyAvailableStandards/index.html

[16] ENISA. Artificial Intelligence: Cybersecurity Challenges. 12/15/2020. https://www.enisa.europa.eu/publications/artificial-intelligence-cybersecurity-challenges

[17] MITRE ATLAS. The ATLAS matrix shows the progression of tactics used in attacks, with the ML techniques belonging to each tactic.https://atlas.mitre.org/matrices/ATLAS

[18] MITRE ATT&CK. MITRE ATT&CK® is a globally accessible knowledge base of adversary tactics and techniques, based on real-world observations.https://attack.mitre.org/#

[19] CERT-IST. Common Vulnerability Assessment System. 07/07/2015.https://www.cert-ist.com/public/fr/SO_detail?format=html&code=cvss%20v3

[20] NCSC. Guidelines for the development of secure AI systems. 2023. https://www.ncsc.gov.uk/files/Guidelines-for-secure-AI-system-development.pdf

[21] Cyberattacks and remediation. Managing remediation.https://cyber.gouv.fr/publications/cyberattaques-et-remediation-piloter-la-remediation

[22] CNIL. How to set up or improve the incident management process?https://www.cnil.fr/fr/notifications-dincidents-de-securite-aux-autorites-de-regulation-comment-sorganiser-et-qui-sadresser

[23] Incident Response Recommendations and Considerations for Cybersecurity Risk Management: A CSF 2.0 Community Profile. https://csrc.nist.gov/pubs/sp/800/61/r3/ipd

[24] Computer Security Incident Handling Guide, NIST SP 800-61 Rev. 2. https://csrc.nist.gov/pubs/sp/800/61/r2/final

[25] The NIST Cybersecurity Framework (CSF) 2.0, February 26, 2024. https://doi.org/10.6028/NIST.CSWP.29

# 8 AI & Cyber Glossary

## 8.1 AI Glossary

- **Alignment:** the process of ensuring that the goals and behaviors of an artificial intelligence system match human values and intentions.
- **Artificial Intelligence:** according to Cambridge Dictionary, is a technology allowing "the use or study of computer systems or machines that have some of the qualities that the human brain has, such as the ability to interpret and produce language in a way that seems human, recognize or create images, solve problems, and learn from data supplied to them".
- **AI Model:** a program that has been trained on a set of data to recognize certain patterns or make certain decisions without further human intervention.
- **AI System (AIS):** All the technical components of an application based on an AI model: the implementation of the AI model, front-end services for users, databases, logging, etc.
- **AutoML** or Automated Machine Learning: Automates the tasks of developing a Machine Learning model, for example data preparation, variable selection, training, etc.
- **Bias:** prejudices or systematic errors in data or AI algorithms that can lead to unfair or inaccurate results, often due to unrepresentative training data or poorly designed algorithms. These biases can lead to discriminatory decisions and undermine the fairness of AI systems.
- **ChatGPT**: Chatbot developed by OpenAI, based on a large language model from the GPT family.
- **Chunk**: A block of information extracted from a larger data set.
- **Classification:** a model's task of assigning labels or categories to an input from a fixed set of possible categories, such as identifying whether an image is of a cat or a dog.
- **Clustering:** a technique for grouping similar data into clusters or groups, without knowing the categories in advance.
- **Dataset:** a structured set of data used to train, test, or evaluate artificial intelligence models. It is often separated into two subsets: training data and validation data.
- **Deep learning:** subdomain of machine learning using so-called deep neural networks to model complex data, inspired by the functioning of the human brain.
- **Differential Privacy:** a privacy-preserving technique that adds random noise to data to prevent the identification of personal information from aggregated results, while preserving the usefulness of the data.
- **Embeddings:** vector representations of data, such as text, transformed into digital vectors for use by AI models.

- **Fake or Deepfake:** media content (videos, audios, images, etc.) manipulated or generated by AI to appear authentic, often for malicious purposes.
- **Feature:** measurable characteristics or attributes of data used by AI models to make predictions or analyses.
- **Fine-tuning:** retraining a model, from an already trained model, to adapt it to a specific task or context of use.
- **Generalization:** the ability of a model to behave on production data with performance comparable to that during the building phase.
- **Generative AI:** a type of AI capable of creating new content, such as text, images, or music, by learning patterns from existing data and using them to generate original results.
- **Guardrails:** control and safety mechanisms built into AI systems to prevent unwanted or dangerous behavior, ensuring the model operates within safe and ethical boundaries.
- **Hallucination:** a phenomenon where an AI model generates information that appears plausible but is actually incorrect or fabricated, often due to insufficient training data or ambiguities in queries.
- **Hyperparameters:** parameters set before training an AI model, such as the learning rate or the number of layers in a neural network, that influence the model's performance but are not learned directly from the data.
- **Inference:** the process by which a pre-trained model applies its knowledge to make predictions or decisions based on new data. This is the phase where the model uses the weights and parameters learned during training to generate results from previously unseen input data.
- **Large Language Model or LLM:** a class of generative AI models that can generate text close to a human's natural language and are typically trained on a large dataset.
- **Machine Learning:** a branch of artificial intelligence (AI) that focuses on developing algorithms and models allowing computers to learn and make predictions or decisions based on data. Rather than being explicitly programmed to perform a specific task, they identify patterns in the data and use this knowledge to improve their performance on similar tasks or to predict future outcomes.
- **Master prompt (pre-prompt):** instructions or initial context provided to an AI model to guide its response generation, defining the tone, style, or constraints to be respected in subsequent interactions. The Master Prompt is confidential by default and is not expected to be accessible to users.
- **Natural Language Processing** or NLP: a subdiscipline of computer science and artificial intelligence that focuses on the interaction between computers and human language. NLP encompasses a set of techniques and algorithms that enable machines to understand, interpret, and generate human language in

meaningful ways. This includes tasks such as speech recognition, machine translation, sentiment analysis, and text generation.

- **Overfitting:** phenomenon where a *machine learning* model performs well on training data but fails to generalize to new data, having memorized the specific details of the training data too well.
- **Parameters:** values stored by the model on which it is based to generate its output.
- **Predictive AI:** a type of AI that analyzes historical and current data to make predictions about future events, identifying patterns and relationships in the data.
- **Pre-trained model:** an AI model already trained on a large dataset to gain general knowledge, which can be reused and fine-tuned for specific tasks with less data or resources.
- **Prompt:** an instruction or query formulated in natural language and provided to generative AI in order to generate a response (content).
- **Reinforcement learning:** a learning method in which an agent performs a series of actions over time, for which it receives rewards. Learning aims to determine the best strategy for the agent, that is, the one that maximizes its gain*, i.e.* its total rewards.
- **Regression:** machine learning technique used to predict a continuous value from input data, by modeling the relationship between independent variables and a dependent variable.
- **Regulation on Artificial Intelligence or RIA (*AI Act*):** European regulation aimed at regulating the development and use of artificial intelligence, with an emphasis on security, transparency, ethics and the protection of personal data, applicable on the European Union market.
- **Retrieval-Augmented Generation or RAG:** a technique used in language models (LLM) to improve text generation by using external information retrieved from databases or documents to enrich the context of a language model. The model then generates more accurate and relevant answers by combining its internal capabilities with the obtained external data.
- **Model resilience:** the ability of an AI model to resist attacks (e.g. adversarial) or attempts at intentional manipulation, continuing to provide accurate and secure results despite malicious inputs.
- **Robustness:** the ability of an AI model to maintain stable and reliable performance in the face of variations or disturbances in input data, such as errors, noise, or unexpected data.
- **Shot-Based Prompting:** an incentive technique where an AI model is guided by one or more examples (shots) to improve its understanding and response to a specific task. We distinguish between *Zero-Shot prompting* where no examples are provided and the model must rely entirely on its pre-trained knowledge; the *One-Shot prompting* where only one example is given to clarify the model's

task; and the *Few-Shot prompting* where two or more examples are included, allowing the model to recognize patterns and provide more accurate responses.

- **Supervised learning:** a learning method where a model is trained on labeled data, i.e., data for which the desired results are already known.
- **Temperature:** a parameter controlling the creativity of responses generated by an AI model (often between 0 and 1). A low temperature promotes predictable and conservative responses, while a high temperature increases diversity and creativity but can lead to inconsistent responses.
- **Test data** or Test *dataset*: this is the data set used to evaluate the final performance of an AI model after training and validation. This data was not seen by the model during training or validation; it allows the assessment of its ability to generalize new situations and provide accurate predictions.
- **Token**: subset of a word constituting a processing unit by a *Large Language Model*.
- **Training:** the process by which an AI model learns to make predictions by adjusting its parameters based on data. It includes data preparation, parameter adjustment to minimize errors, and validation to prevent overfitting and ensure generalization to new data.
- **Training data** or training *dataset*: this is the set of data that is used to train (or learn) a model. It can include a label associated with each data (case of supervised learning) or not (case of unsupervised learning).
- **Underfitting**: phenomenon where a *machine learning* model fails to capture underlying trends in training data, resulting in poor performance on both training and new data, often due to the model being too simple or insufficient training.
- **Unsupervised learning:** a technique where the model learns from unlabeled data, identifying patterns or structures without knowing the results in advance.
- **Validation data** or validation *dataset*: this is a data set similar to the training *dataset* which is used to choose between several models and also to check that there is no *overfitting*.

## 8.2 Cybersecurity

- **Access control:** a set of measures and technologies aimed at regulating and securing access to IT resources, systems or physical areas, by verifying the identity and authorizations of users.
- **Advanced Persistent Threat or APT:** targeted and sustained cyberattack techniques in which an unauthorized person gains access to the network and remains undetected for an extended period, with potentially destructive consequences.

- **Adversarial attacks:** a technique to trick an AI model by introducing subtle perturbations into the input data, designed to cause errors or unwanted behavior, thereby exploiting vulnerabilities in the model.
- **ANSSI (National Agency for Information Systems Security)**: National Cybersecurity Authority. It is placed under the authority of the Prime Minister and attached to the Secretary General of Defense and National Security. https://cyber.gouv.fr/en/about-french-cybersecurity-agency-anssi
- **Antivirus software:** software designed to detect, prevent and eliminate malware (viruses, Trojan horses, etc.) on a computer or network, thus protecting systems against computer threats.
- **Black box:** a testing technique where the examiner has no knowledge of the internal workings of a system, focusing only on the inputs provided and the outputs observed to verify expected behavior.
- **Botnet** (Zombie Machine Networks): a Botnet, in other words a network of bots (botnet: contraction of robot network), is a network of compromised machines at the disposal of a malicious individual (the master). This network is structured in such a way as to allow its owner to transmit orders to all or part of the machines in the botnet and to activate them as he wishes.
- **CERT (*Computer Emergency Response Team*):** structure responsible for responding to cybersecurity incidents. It also carries out the following missions: processing alerts and responding to computer attacks, establishing and maintaining a vulnerability database, preventing incidents by disseminating information on the precautions to take to minimize the risks or, at worst, the consequences of incidents, and possible coordination with other entities.
- **Chief Information Security Officer or CISO:** the person responsible for information systems security, who defines or contributes to their company's information security policy. They are responsible for its implementation and monitor it.
- **Cybercriminal:** a person who commits crimes through digital means.
- **Cybersecurity:** a set of technologies, processes, and practices designed to protect networks, devices, programs, and data from attack, damage, or unauthorized access.
- **Denial of Service or DoS:** action that has the effect of preventing or severely limiting the ability of a system to provide the expected service. Notes: This action is not necessarily malicious.
- **Digital forensics:** a person or team responsible for revealing information about a system or network, usually for the purpose of a lawsuit or investigation.
- **Distributed Denial of Service or DDoS:** a technique where an attacker intentionally floods a server with excessive traffic from multiple sources, exceeding its processing capacity and making the site or service inaccessible to legitimate users.

- **DLP (*Data Loss Prevention*):** data loss or leak protection techniques, which are used to identify, track important data and limit their loss (theft, destruction, involuntary encryption (ransomware)).
- **EBIOS Risk Manager:** French benchmark risk analysis method, enabling organizations to assess and treat risks. https://cyber.gouv.fr/en/publications/ebios-risk-manager-method
- **EDR (*Endpoint detection and response*):** tools for analyzing behavior on IT equipment (workstations, servers, smartphones, etc.) to detect and block threats (primarily malware and ransomware) as well as illegitimate actions. EDRs rely heavily on the use of artificial intelligence and are often offered by antivirus software vendors.
- **Encryption:** the process of transforming readable data (plaintext) into an unreadable format (ciphertext) using an algorithm and a key, in order to protect the confidentiality and integrity of information against unauthorized access.
- **Evasion (attacks by):** techniques to circumvent an AI model's detection mechanisms by subtly modifying input data to avoid being identified as a threat, thus allowing malicious inputs to go unnoticed.
- **Extraction (attacks by):** techniques where an attacker attempts to reconstruct or steal the internal parameters of an AI model by exploiting its responses or behaviors, often with the aim of duplicating the model or accessing sensitive information.
- **Firewall:** a network security device that controls and filters incoming and outgoing traffic based on predefined security rules, thus protecting a network from unauthorized access and external threats.
- **Gray box:** a testing method where the examiner has partial knowledge of the internal workings of a system, combining aspects of black-box and white-box testing to evaluate both input/output and some internal details.
- **IDS (*Intrusion Detection System*):** computer intrusion detection systems, either by signatures or by anomaly detection. IDS actions are generally carried out by firewalls or dedicated network equipment by analyzing the content of frames passing through the network.
- **IPS (*Intrusion Prevention System*):** Intrusion prevention system that monitors network traffic in real time to automatically detect and block malicious activity, based on known signatures or abnormal behavior, to actively protect the network against potential threats.
- **Jailbreak:** a technique for circumventing restrictions or safeguards in an AI model to induce it to generate unauthorized or potentially dangerous responses or actions by exploiting vulnerabilities in its instructions or parameters.
- **Malware:** software designed with the intention of performing malicious tasks on the computer system.

- **Man-in-the-Middle or MitM:** a category of attacks where a malicious person interposes themself in an exchange in an unnoticed manner to users or systems.
- **Minimum privilege:** security principle according to which a user or process has only the access rights strictly necessary to carry out its tasks, thus limiting the risks in the event of compromise.
- **Multi-Factor Authentication or MFA:** a security method that requires at least two distinct forms of verification to grant access to a system or application, typically combining something the user knows (password), has (phone), or is (fingerprint), to strengthen protection against unauthorized access.
- **National Commission for Information Technology and Civil Liberties or CNIL:** personal data regulator. It supports professionals in their compliance and helps individuals control their personal data and exercise their rights.
- **Network:** a set of interconnected computers and devices that share resources and communicate with each other using common technologies and protocols, enabling the exchange of data and access to shared services.
- ***Oracle Attack*:** techniques where an attacker creates inputs and receives the outputs of the attacked model, with the aim of gaining information about that model - and sometimes even the training data.
- **Personal Data or PII (*Personal Identifiable Information*):** information that can directly or indirectly identify a natural person (name, address, social security number, biometric data, etc.) requiring special protection due to their sensitivity and potential risks to privacy.
- **Phishing:** a fraud technique that involves impersonating a trusted entity in order to trick individuals into disclosing sensitive information, such as passwords or credit card numbers, usually through deceptive emails, messages, or websites. Examples include *Spear*-phishing, which targets specific individuals or organizations using personalized information; *smishing*, SMS phishing; and *vishing*, telephone phishing.
- **Poisoning attacks:** techniques where an attacker alters an AI model's training data to introduce bias or malicious behavior, thereby compromising the model's reliability and accuracy.
- **Prompt injection:** a manipulation technique consisting of inserting malicious instructions into the text input of a language model (LLM) by exploiting, for example, the absence of a clear separation between system instructions and user input, thus making it possible to control or alter the behavior of the model.
- **Ransomware:** malware that encrypts or locks access to a user's data, then demands payment of a ransom to restore access.
- **Red team:** this is a group hired by an organization to test its security. The group will attempt to carry out attacks against the organization and produce a report to inform the organization of the security vulnerabilities it has discovered.

- **Role-Based Access Control or RBAC:** access control model for an information system in which access to a resource is based on the role of the user concerned.
- **Security Information and Event Management or SIEM:** a software solution that detects security incidents from event logs. SIEM can also be a tool for centralizing a company's logs.
- **SIEM**: see Security Information and Event Management.
- **SOC (*Security Operation Center*):** a department or team responsible for detecting and classifying IT security incidents. Typically, the SOC operates SIEM software. The SOC may also play a role in developing the company's IT security strategies.
- **Social engineering:** a psychological manipulation technique used to induce individuals to disclose confidential information or perform compromising actions, often by exploiting the trust or naivety of the victims, in order to circumvent security measures.
- **Penetration test or *Pentest*:** a methodical assessment of the security of a computer system, carried out by cybersecurity experts, who simulate attacks to identify and exploit vulnerabilities, in order to correct them and improve protection against real threats.
- **UEBA** (User and Entity Behavior Analytics): user and entity behavior analysis examines the behavior of users or network devices and compares it to past behaviors or benchmarks to detect deviations and identify threats. Tools implementing UEBA specifically look for: compromised credentials, use of administrator accounts, privilege escalation, data leaks. Some SIEM solutions include UEBA features.
- **Virtual Private Network or VPN:** technology for protecting data flows exchanged between two interconnected network devices through an unsecure public network (such as the Internet), or for protecting flows exchanged between a mobile terminal device and a remote network device through an unsecure network (case of nomadic VPN). They ensure the security of network exchange equivalent to that provided by a physically and logically dedicated point-to-point link.
- **Virus:** a category of malware that can replicate and spread itself.
- **Watermarking:** a technique of inserting hidden information (a "watermark") into digital data, such as text or images, in a way that is imperceptible to the user. This watermark can be detected by specialized tools to prove the origin or authenticity of the data, protect against unauthorized use, and deter malicious manipulation.
- **White box:** a testing or analysis approach where the examiner has access to the source code and internal structure of a system, allowing detailed verification of the program's internal functioning and logic.
- **XDR (*Extended Detection and Response*):** tool that uses the principles of EDR behavior analysis, performing correlations with several sources such as

messaging, collaborative file sharing, cloud-hosted applications, etc. EDR data generally feeds into XDR solutions.

- **Zombie machine network (*Botnet*):** a network of compromised machines at the disposal of a malicious individual (the master). This network is structured in such a way as to allow its owner to transmit orders to all or some of the machines in the botnet and to activate them as he wishes.

## 8.3 Others

- **Application Programming Interface** (API): an API is a software interface allowing to "connect" one software / service to another software / service to exchange data and functionalities.
- **CI/CD (*Continuous Integration / Continuous Delivery*):** software development practices where code changes are regularly integrated into a shared repository (CI), followed by automated testing and automatic deployment of validated versions to production environments (CD), aiming at accelerating development and improving software quality.
- **Central Processing Unit or CPU:** the main component of a computer that executes program instructions by performing arithmetic and logical operations, thus managing data flow and computing processes.
- **DevSecOps** (*development*, *security* and *operations)*: It is an application development practice that automates the integration of security and security practices into every phase of the software development lifecycle, from initial design through integration and testing to delivery and deployment.
- **Electronic document management or EDM:** software solution for organizing and managing information in the form of electronic documents.
- **Graphics Processing Unit or GPU:** processor specialized in image rendering, 2D/3D image processing, and mathematical calculations, widely used for LLM training.
- **General Data Protection Regulation or GDPR:** European regulation aimed at strengthening and harmonizing the protection of personal data within the European Union, by imposing strict obligations on companies and granting rights to individuals regarding the collection, use and storage of their data.
- **Inference API**: an inference API allows to manage machine learning inference models by performing inferences without manual deployment and applying them to clean data.
- **KMS/HSM:** KMS (*Key Management System)* is a centralized cryptographic key management tool. The HSM (*Hardware Security Module*) is a physical computing device (often an external device) that protects and manages secrets (including digital keys) and performs cryptographic functions.
- **MLOps:** a set of practices that aims to reliably and efficiently deploy and maintain machine learning models in production.

- **NIST (*National Institute of Standards and Technology*):** an agency of the United States Department of Commerce whose mission is to promote the economy by developing technologies, metrology, and standards for industry. https://www.nist.gov/
- **Proof of concept or POC:** implementation aiming at demonstrating the feasibility of a project.
- **Service Level Agreement or SLA:** service contract between an IT service provider and a client.

# 9 Appendix 1 - Prevention methods

The prevention measures listed here are used to write the attack fact sheets and supplemented if necessary. The color code used is that of Figure 18.

## 9.1 I Cybersecurity protection

| | | Lifecycle phases | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **Description** | **A - Planning and design** | **B – Data Collection & processing** | **C – Construction of the model / adaptation of an existing model** | **D - Testing, evaluation, verificatio** | **E - Provision , use, deploym ent** | **F – Operation and maintena nce** | **G – Decommiss ioning / disposal** | **Source** | **Comments** |
| ***1 - General recommendations*** | | | | | | | | | | |
| **#1** | Design the AI system using a privacy-by-design approach to meet data protection requirements throughout the lifecycle. | Yes | Yes | Yes | Yes | Yes | Yes | Yes | *[2]* | |
| **#2** | Carry out a formal risk assessment. | | | | | | | | *[3]* | |
| **#3** | Limit the use of AI systems for the automation of critical actions on other information systems. | | | | | Yes | Yes | | *[2]* | |
| **#4** | Do not allow AI systems to run automatically critical actions on the IS. | | | | | Yes | Yes | | *[1] - R9* | |
| ***2- Recommendations for infrastructure and architecture*** | | | | | | | | | | |
| **#5** | Identify the most sensitive information and servers and keep a network diagram. | | | | | | | | *[3]* | |
| **#6** | Implement a minimum level of security across the whole IT stock. | | | | | | | | *[3]* | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **#7** | Protect against threats relating to the use of removable media. | | | | | | | *[3]* | |
| **#8** | Use a centralised management tool to standardise security policies. | | | | | | | *[3]* | |
| **#9** | Activate and configure the firewall on workstations. | | | | | | | *[3]* | |
| **#10** | Host the AI system in trusted environments consistent with security needs. | | | | | Yes | Yes | *[1] - R11* | |
| **#11** | Apply cloud-specific measures, where appropriate, taking into account applicable regulations and organizational policies. | Yes | Yes | Yes | Yes | Yes | Yes | Yes | *[2]* | In the case of structures subject to specific regulations (e.g. in health with HDS qualification, etc.) SecNumCloud certification is a key. See [1] R14. |
| **#12** | Prioritise SecNumCloud hosting when deploying an AI system in a public cloud. | Yes | Yes | Yes | Yes | Yes | Yes | Yes | *[1] - R14* | In the case of structures subject to specific regulations (e.g. in health with HDS qualification, etc.) SecNumCloud certification is a key. See [2] |
| **#13** | Control outsourced services. | | | | | | | *[3]* | |
| **#14** | Apply the recommendations for outsourcing if applicable. | Yes | | | | | Yes | Yes | *[2]* | |
| **#15** | Identify each individual accessing the system by name and distinguish the user/administrator roles. | | | | | | | *[3]* | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **#16** | Have an exhaustive inventory of privileged<br>accounts and keep it updated. | | | | | | | | *[3]* | |
| **#17** | Apply secure administration recommendations on the AI system. | Yes | | | | | Yes | Yes | *[2]* | |
| **#18** | Allocate the correct rights to the information system's sensitive resources. | | | | | | | | *[3]* | |
| **#19** | Leverage a controlled access system for critical AI components. | | Yes | Yes | | Yes | Yes | | *[2]* | |
| **#20** | Control Access to AI Models and Data in Production: Require users to verify their identities before accessing a production model. Require authentication for API endpoints and monitor production model queries to ensure compliance with usage policies and to prevent model misuse. | Yes | | | | Yes | Yes | | *[17]- AML.M0019* | |
| **#21** | Organise the procedures relating to users joining, departing and changing positions. | | | | | | | | *[3]* | |
| **#22** | Manage and secure developer and administrator privileged access to the AI system. | | | | | Yes | Yes | | *[1] - R10* | |
| **#23** | Prohibit Internet access from devices or servers used by the information system administration. | | | | | | | | *[3]* | |
| **#24** | Use a dedicated and separated network for information system administration. | | | | | | | | *[3]* | |
| **#25** | Reduce administration rights on workstations to strictly operational needs. | | | | | | | | *[3]* | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **#26** | Use secure network protocols when they exist. | | | | | | | | *[3]* | |
| **#27** | Encrypt sensitive data sent through the Internet. | | | | | | | | *[3]* | |
| **#28** | Implement a secure access gateway to the Internet. | | | | | | | | *[3]* | |
| **#29** | Implement a secure Internet gateway for an AI system hosted on the Internet. | | | | Yes | Yes | | | *[1] - R13* | |
| **#30** | Segregate the services visible from the Internet from the rest of the information system. | | | | | | | | *[3]* | |
| **#31** | Segment the network and implement a partitioning between these areas. | | | | | | | | *[3]* | |
| **#32** | Only allow controlled devices to connect to the network of the organization. | | | | | | | | *[3]* | |
| **#33** | Secure the dedicated network interconnections with partners. | | | | | | | | *[3]* | |
| **#34** | Ensure the security of Wi-Fi access networks and that uses are separated. | | | | | | | | *[3]* | |
| **#35** | Protect your professional email. | | | | | | | | *[3]* | |
| **#36** | Control and protect access to the server rooms and technical areas. | | | | | | | | *[3]* | |
| **#37** | Take measures to physically secure mobile devices. | | | | | | | | *[3]* | |
| **#38** | Encrypt sensitive data, in particular on hardware that can potentially be lost. | | | | | | | | *[3]* | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **#39** | Secure the network connection of devices used in a mobile working situation. | | | | | | | | *[3]* | |
| **#40** | Adopt security policies dedicated to mobile devices. | | | | | | | | *[3]* | |
| ***3- Have a deployment plan*** | | | | | | | | | | |
| ***4- Be vigilant about the resources used*** | | | | | | | | | | |
| **#41** | Activate and configure the most important component logs. | | | | | | | | *[3]* | |
| **#42** | Ensure the traceability of actions carried out on the AI system. | | | | | | Yes | Yes | *[2]* | |
| **#43** | Record all processing carried out within the AI system. | | | | | | Yes | Yes | *[1] - R29* | |
| ***5- Secure and strengthen the learning process*** | | | | | | | | | | |
| **#44** | Adopt a strict policy on what data is accessed by the AI system, especially sensitive data. | | Yes | | | Yes | | | *[2]* | |
| **#45** | Secure access and storage of training data. | | Yes | Yes | | | Yes | | *[2]* | |
| ***6- Make the application reliable*** | | | | | | | | | | |
| **#46** | Set and verify rules for the choice and size of passwords. | | | | | | | | *[3]* | |
| **#47** | Protect passwords stored on systems. | | | | | | | | *[3]* | |
| **#48** | Change the default authentication settings on devices and services. | | | | | | | | *[3]* | |
| **#49** | Prefer a two-factor authentication when possible. | | | | | | | | *[3]* | The subject of strong authentication is also discussed in [2] |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **#50** | Implement multi-factor authentication for all administrative tasks on AI systems. | Yes | | | | Yes | Yes | | *[2]* | The topic of strong authentication is also discussed in [3] |
| **#51** | Restrict Library Loading: Prevent abuse of library loading mechanisms in the operating system and software to load untrusted code by configuring appropriate library loading mechanisms and investigating potential vulnerable software.<br>File formats such as pickle files that are commonly used to store AI models can contain exploits that allow for loading of malicious libraries. | | | Yes | | Yes | Yes | | *[17] AML.M0011* | |
| **#52** | Strengthen security measures for AI services hosted on the Internet. | | | | | Yes | Yes | | *[1] - R33* | |
| **#53** | Define an update policy for the components of the information system. | | | | | | | | *[3]* | |
| **#54** | Anticipate the software and system end of life/maintenance and limit software reliance. | | | | | | | | *[3]* | |
| ***7- Think about an organizational strategy*** | | | | | | | | | | |
| **#55** | Designate a point of contact in information system security and make sure staff are aware of him or her. | | | | | | | | *[3]* | |
| **#56** | Supervise the operation of the AI system. | | | | | | Yes | | *[2]* | |
| **#57** | Define and apply a backup policy for critical components. | | | | | | | | *[3]* | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **#58** | Dedicate GPU components to the AI system. | Yes | | Yes | Yes | Yes | Yes | | *[1] - R16* | |
| **#59** | Closely monitor technical developments which would, for example, limit the use of personal data. | Yes | Yes | Yes | Yes | Yes | Yes | | *[2]* | |
| **#60** | Implement a data management system. | Yes | Yes | | Yes | Yes | Yes | Yes | *[2]* | |
| **#61** | Leverage secure deletion methods for data removal. | | | | | | | Yes | *[2]* | |
| ***8- Preventive measures*** | | | | | | | | | | |
| **#62** | Favor the use of products and services qualified by ANSSI. | | | | | | | | *[3]* | |
| **#63** | Define a security incident management procedure. | | | | | | | | *[3]* | |
| **#64** | Train the operational teams in information system security. | | | | | | | | *[3]* | |
| **#65** | Raise users' awareness about basic information security. | | | | | | | | *[3]* | |
| **#66** | Undertake regular controls and security audits then apply the associated corrective actions. | | | | | | | | *[3]* | |

## 9.2 II AI "secure by design" protection

| | Description | A - Planning and design | B – Data collection & processing | C – Construction of the model / adaptation of an existing model | D - Testing, evaluation, verification | E - Provision, use, deployment | F – Operation and maintenance | G – Decommissioning / disposal | Source | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| | | **Lifecycle phases** | | | | | | | | |
| ***1 - General recommendations*** | | | | | | | | | | |
| **#1** | Integrate security into all phases of the lifecycle of an AI system. | Yes | Yes | Yes | Yes | Yes | Yes | Yes | *[1] - R1* | Studying the security of each stage of the AIS lifecycle is equivalent to integrating security into each stage of the lifecycle provided by [2] |
| **#2** | Study the security of each stage of the AI system lifecycle (from training data collection to inference phase and decommissioning). | Yes | Yes | Yes | Yes | Yes | Yes | Yes | *[2]* | Studying the security of each stage of the AIS lifecycle is equivalent to integrating security into each stage of the lifecycle provided by [1] - R1. |
| **#3** | Conduct a data protection impact assessment if required. | | Yes | | Yes | | | | *[2]* | |
| **#4** | Perform a dedicated risk analysis by integrating the entire organisational context (for instance the impact of an AI system failure should be assessed across the whole organization). | Yes | | | Yes | | | | *[2]* | The performance of a dedicated risk analysis is also provided for by [1] - R2 |
| **#5** | Conduct a risk analysis on AI systems before the training phase. | Yes | | | Yes | | | | *[1] - R2* | The performance of a dedicated risk |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | analysis is also provided for by [2] |
| **#6** | Limit automatic actions performed by an AI system handling uncontrolled inputs. | | | | | Yes | Yes | | *[1] - R27* | |
| **#7** | Ensure AI is thoughtfully and appropriately integrated into critical processes and provide safeguards. | Yes | | Yes | | Yes | | | *[2]* | |
| ***2- Recommendations for infrastructure and architecture*** | | | | | | | | | | |
| **#8** | Identify, track and protect AI-related assets. | | Yes | Yes | | | Yes | | *[2]* | |
| **#9** | AI Bill of Materials: An AI Bill of Materials (AI BOM) contains a full listing of artifacts and resources that were used in building the AI. The AI BOM can help mitigate supply chain risks and enable rapid response to reported vulnerabilities.<br>This can include maintaining dataset provenance, i.e. a detailed history of datasets used for AI applications. The history can include information about the dataset source as well as well as a complete record of any modifications. | | Yes | Yes | Yes | | Yes | | *[17]- AML. M0023* | |
| **#10** | Define the modalities for the use of the AI system and frame its integration into the decision-making process, in particular in the case of automation. | Yes | | | | Yes | Yes | | *[2]* | |
| **#11** | Control Access to AI Models and Data at Rest: Establish access controls on internal model registries and limit internal access to production models. Limit access to training data only to approved users. | Yes | Yes | | | Yes | Yes | | *[17] - AML. M0005* | |
| **#12** | Encrypt Sensitive Information: Encrypt sensitive data such as AI models to protect against | | | | | | Yes | | *[17] - AML.* | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | adversaries attempting to access sensitive data. | | | | | | | | *M0012* | |
| **#13** | Isolate each phase of the AI system into a dedicated environment. | Yes | Yes | Yes | Yes | Yes | | | *[1] - R12* | |
| **#14** | Isolate the AI system in one or more dedicated technical environments. | Yes | Yes | Yes | Yes | Yes | | | *[1] - R28* | |
| ***3- Have a deployment plan*** | | | | | | | | | | |
| **#15** | Design the architecture so that, when scaling occurs, it does not impact negatively the level of security. | Yes | | Yes | | Yes | | | *[2]* | |
| **#16** | Apply DevSecOps principles across all phases of the project. | Yes | Yes | Yes | Yes | Yes | Yes | | *[2]* | DevSecOps is planned in [1] - R5. |
| **#17** | Apply DevSecOps principles to all phases of the project. | Yes | Yes | Yes | Yes | Yes | Yes | | *[1] - R5* | DevSecOps is provided by [2]. |
| **#18** | Take into account data confidentiality issues. | Yes | Yes | Yes | | | | Yes | *[2]* | Confidentiality issues must be integrated and are provided for by [1] - R7. |
| **#19** | Manage data confidentiality issues from the AI system design phase. | Yes | Yes | Yes | | | | Yes | *[1] - R7* | The issues of confidentiality must be integrated and are provided for by [2] |
| **#20** | Ensure the pseudonymisation or anonymisation of data where necessary. | | Yes | Yes | Yes | | Yes | | *[2]* | |
| **#21** | Take the need-to-know issue into account when designing the AI system. | Yes | | Yes | | | | | *[2]* | The need-to-know principle is provided for by [1] - R8. |
| **#22** | Manage users data access rights issue from the AI system design phase. | Yes | | Yes | | | | | *[1] - R8* | The need-to-know principle is provided for by [2]. |
| **#23** | Secure the production deployment chain for AI systems. | | | | | Yes | | | *[1] - R22* | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **#24** | Conduct business tests of AI systems before deployment to production. | | | | Yes | Yes | | | *[1] - R24* | |
| **#25** | Manage side-channel attacks on the AI system. | Yes | | Yes | | Yes | Yes | | *[1] - R17* | |
| ***4- Be vigilant about the resources used*** | | | | | | | | | | |
| **#26** | Use secure formats for obtaining, storing and distributing AI models. | | | Yes | | Yes | Yes | | *[2]* | The requirement for secure formats is also provided by [1] - R6. |
| **#27** | Use secure AI model formats. | | | Yes | | Yes | Yes | | *[1] - R6* | The requirement for secure formats is also provided by [2] |
| **#28** | Implement mechanisms to verify the integrity of model files before loading them. | | | Yes | | Yes | Yes | | *[2]* | Verification of the integrity of model files is also provided for by [1] - R20. |
| **#29** | Protect the integrity of AI system files. | | | Yes | | Yes | Yes | | *[1] - R20* | Verification of the integrity of model files is also provided by [2] |
| **#30** | Verify AI Artifacts: Verify the cryptographic checksum of all AI artifacts to verify that the file was not modified by an attacker. | | | Yes | | Yes | Yes | | *[17]- AML. M0014* | |
| **#31** | Assess the level of trust of libraries and plug-insused in AI system. | | Yes | Yes | | | | | *[2]* | The assessment of the trust level of libraries is also provided by [1] - R3. |
| **#32** | Evaluate the level of confidence in the libraries and external modules used in the AI system. | | Yes | Yes | | | | | *[1] - R3* | The assessment of the trust level of libraries is also provided for by [2] |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **#33** | Ensure the quality and assess the level of confidence of the external data used in the AI system. | | Yes | Yes | | | Yes | | *[2]* | The assessment of the level of trust of external data sources is also provided for by [1] - R4. |
| **#34** | Evaluate the level of confidence in external data sources used in the AI system. | | Yes | Yes | | | Yes | | *[1] - R4* | The assessment of the level of trust of external data sources is also provided for by [2] |
| **#35** | Ensure that data collection has been carried out in a fair and ethical manner, for those used both for the development and for the use of the system. | | Yes | | | | | | *[2]* | |
| **#36** | Train an AI model only with data which users can legitimately access. | | Yes | Yes | | | | | *[1] - R18* | |
| **#37** | Maintain AI Dataset Provenance: Maintain a detailed history of datasets used for AI applications. The history should include information about the dataset's source as well as a complete record of any modifications. | | Yes | Yes | Yes | | | | *[17] - AML. M0025* | |
| **#38** | AI Telemetry Logging: Implement logging of inputs and outputs of deployed AI models. Monitoring logs can help to detect security threats and mitigate impacts. Additionally, having logging enabled can discourage adversaries who want to remain undetected from utilizing AI resources. | | | | | Yes | Yes | | *[17] - AML. M0024* | |
| ***5- Secure and strengthen the learning process*** | | | | | | | | | | |
| **#39** | Protect the integrity of AI model training data. | | Yes | Yes | | | | | *[1] - R19* | |
| **#40** | Assess the security of the learning and re-learning methods used. | | | Yes | | | Yes | | *[2]* | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **#41** | Do not re-train an AI model in production. | | | | | Yes | | | *[1] - R21* | |
| **#42** | Implement measures on the extracted data, metadata, annotation and features, and on the AI system model(s) including: clean up data; identify relevant and strictly necessary data (in terms of volume, categories, granularity, typology, etc.) ; pseudonymise or anonymise data if necessary. | | Yes | Yes | | | | | *[2]* | |
| ***6- Make the application reliable*** | | | | | | | | | | |
| **#43** | Ensure the confidentiality and integrity of inputs and outputs. | | | Yes | | Yes | Yes | | *[2]* | |
| **#44** | Ensure security filters to detect malicious instructions. | | | | | Yes | Yes | | *[2]* | |
| **#45** | Ensure that all data, metadata and annotations are kept up to date and accurate (in particular to avoid drift). | | Yes | | | | Yes | | *[2]* | |
| **#46** | Conduct continuous evaluation of model accuracy and performance. | | | | | | Yes | | *[2]* | |
| **#47** | Protect the AI system by filtering user input and output. | | | | | Yes | Yes | | *[1] - R25* | |
| **#48** | Limit Public Release of Information: Limit the public release of technical information about the AI stack used in an organization's products or services. Technical knowledge of how AI is used can be leveraged by adversaries to perform targeting and tailor attacks to the target system. Additionally, consider limiting the release of organizational information - including physical locations, researcher names, and department structures - from which technical details such as AI techniques, model architectures, or datasets may be inferred. | | | | | Yes | Yes | | | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **#49** | Limit Model Artifact Release: Limit public release of technical project details including data, algorithms, model architectures, and model checkpoints that are used in production, or that are representative of those used in production.. | | | | | Yes | Yes | | *[17]- AML. M00 01* | |
| **#50** | Manage and secure the interactions of the AI system with other business applications. | | | | | Yes | Yes | | *[1] - R26* | |
| ***7- Think about an organizational strategy*** | | | | | | | | | | |
| **#51** | Document design choices. | Yes | | Yes | | | | | *[2]* | |
| **#52** | Identify key individuals and oversee the use of subcontractors. | Yes | | | | Yes | | | *[2]* | |
| **#53** | Implement a risk management strategy. | Yes | | | Yes | | Yes | | *[2]* | |
| **#54** | Provide for a degraded mode of operations without AI systems. | Yes | | | | Yes | Yes | | *[2]* | Degraded mode of services is also provided for by [1] - R15. |
| **#55** | Provide a downgraded version of business services without an AI system. | Yes | | | | Yes | Yes | | *[1] - R15* | The degraded mode of services is also provided by [2] |
| **#56** | Implement framed generative AI usage policies (depending on the sensitivity of the organisation). | | | | | Yes | Yes | | *[2]* | |
| **#57** | Establish a process to monitor AI system-specific vulnerabilities. | Yes | | Yes | | | Yes | | *[2]* | |
| **#58** | Document datasets used in the product | | Yes | Yes | | Yes | Yes | | *[2]* | |
| **#59** | Facilitate the use of the database | | Yes | Yes | | | | | *[2]* | |
| **#60** | Facilitate the monitoring of data over time until their deletion or anonymization; | | Yes | | | | Yes | Yes | *[2]* | |
| **#61** | Reduce the risk of unexpected data use. | | Yes | | | | Yes | Yes | *[2]* | |
| ***8- Preventive measures*** | | | | | | | | | | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **#62** | Regularly train staff on security risks related to AI. | Yes | Yes | Yes | Yes | Yes | Yes | Yes | *[2]* | |
| **#63** | Raise developers awareness of the risks associated with AI-generated source code. | | | | | | Yes | | *[1] - R32* | |
| **#64** | Do not use generative AI tools on the Internet for professional use involving sensitive data. | | | | | Yes | Yes | | *[1] - R34* | |
| **#65** | User training: Educate AI model developers on secure coding practices and AI vulnerabilities. | Yes | | | | | | | *[17]- AML.M0018* | |
| **#66** | Check AI-generated source code systematically. | | | | | | Yes | | *[1] - R30* | |
| **#67** | Limit AI source code generation for critical application modules. | Yes | | Yes | | | | | *[1] - R31* | |
| **#68** | Carry out regular security audits of the AI system. | Yes | Yes | Yes | Yes | Yes | Yes | | *[2]* | Auditing as a preventive measure is also provided for by [1] - R23. |
| **#69** | Conduct security audits of AI systems before deployment to production. | Yes | Yes | Yes | Yes | Yes | Yes | | *[1] - R23* | Auditing as a preventive measure is also provided for by [2] |
| **#70** | Perform regular reviews of the configuration of rights for generative AI tools on business applications. | | | | | | Yes | | *[1] - R35* | |
| **#71** | Anticipate as much as possible the problems potentially associated with the exercise of rights (intellectual property and data protection for instance) to training data or to the model itself. | Yes | Yes | | | | Yes | | *[2]* | |

## 9.4 III Specific protection against AI attacks

| Lifecycle phases | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Description | A - Planning and design | B – Data collection & processing | C – Construction of the model / adaptation of an existing model | D - Testing, evaluation, verification | E - Provision, use, deployment | F – Operation and maintenance | G – Decommissioning / disposal | Source | Comment |
| ***1 - General recommendations*** | | | | | | | | | | |
| ***2- Recommendations for infrastructure and architecture*** | | | | | | | | | | |
| **#1** | AI Model Distribution Methods: Deploying AI models to edge devices can increase the attack surface of the system. Consider serving models in the cloud to reduce the level of access the adversary has to the model. Also consider computing features in the cloud to prevent gray-box attacks, where an adversary has access to the model preprocessing methods. | | | Yes | Yes | Yes | Yes | | [17] - AML.M0017 | |
| **#2** | Use Multi-Modal Sensors: Incorporate multiple sensors to integrate varying perspectives and modalities to avoid a single point of failure susceptible to physical attacks. | Yes | Yes | | | | | | [17] - AML.M0009 | |
| ***3- Have a deployment plan*** | | | | | | | | | | |
| **#3** | Validate AI Model: Validate that AI models perform as intended by testing for backdoor triggers or adversarial influence. Monitor model for concept drift and training data drift, which may indicate data tampering and poisoning. | Yes | Yes | Yes | Yes | Yes | Yes | | [17] - AML.M0008 | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| ***4- Be vigilant about the resources used*** | | | | | | | | | | |
| ***5- Secure and strengthen the learning process*** | | | | | | | | | | |
| **#4** | Sanitize Training Data: Detect and remove or remediate poisoned training data. Training data should be sanitized prior to model training and recurrently for an active learning model.<br>Implement a filter to limit ingested training data. Establish a content policy that would remove unwanted content such as certain explicit or offensive language from being used. | | Yes | Yes | | Yes | | | *[17] - AML. M0007* | |
| **#5** | Model hardening: Use techniques to make AI models robust to adversarial inputs such as adversarial training or network distillation. | | Yes | Yes | | | Yes | | *[17] - AML. M0003* | |
| **#6** | Using ensemble methods: Use an ensemble of models for inference to increase robustness to adversarial inputs. Some attacks may effectively evade one model or model family but be ineffective against others. | | Yes | Yes | | | Yes | | *[17] - AML. M0006* | |
| **#7** | Generative AI Model Alignment: When training or fine-tuning a generative AI model it is important to utilize techniques that improve model alignment with safety, security, and content policies.<br>The fine-tuning process can potentially remove built-in safety mechanisms in a generative AI model, but utilizing techniques such as Supervised Fine-Tuning, Reinforcement Learning from Human Feedback or AI Feedback, | | Yes | Yes | Yes | | Yes | | *[17] - AML. M0022* | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | and Targeted Safety Context Distillation can improve the safety and alignment of the model. | | | | | | | | | |
| ***6- Make the application reliable*** | | | | | | | | | | |
| **#8** | Generative AI Guidelines: Guidelines are safety controls that are placed between user-provided input and a generative AI model to help direct the model to produce desired outputs and prevent undesired outputs. Guidelines can be implemented as instructions appended to all user prompts or as part of the instructions in the system prompt. They can define the goal(s), role, and voice of the system, as well as outline safety and security parameters. | | | | | Yes | Yes | | *[17] - AML. M0021* | |
| **#9** | Generative AI Guardrails: Guardrails are safety controls that are placed between a generative AI model and the output shared with the user to prevent undesired inputs and outputs. Guardrails can take the form of validators such as filters, rule-based logic, or regular expressions, as well as AI-based approaches, such as classifiers and utilizing LLMs, or named entity recognition (NER) to evaluate the safety of the prompt or response. Domain specific methods can be employed to reduce risks in a variety of areas such as etiquette, brand damage, jailbreaking, false information, code exploits, SQL injections, and data leakage. | | | | | Yes | Yes | | *[17] - AML. M0020* | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **#10** | Adversarial Input Detection: Detect and block adversarial inputs or atypical queries that deviate from known benign behavior, exhibit behavior patterns observed in previous attacks or that come from potentially malicious IPs. Incorporate adversarial detection algorithms into the AI system prior to the AI model. | | | | | Yes | Yes | | [17] - AML.M0015 | |
| **#11** | Input Restoration: Preprocess all inference data to nullify or reverse potential adversarial perturbations. | | | | | Yes | Yes | | [17] - AML.M0010 | |
| **#12** | Code signing: Enforce binary and application integrity with digital signature verification to prevent untrusted code from executing. Adversaries can embed malicious code in AI software or models. Enforcement of code signing can prevent the compromise of the AI supply chain and prevent execution of malicious code. | | | | | Yes | Yes | | [17] - AML.M0013 | |
| **#13** | Restrict Number of AI Model Queries: Limit the total number and rate of queries a user can perform. | | | | | Yes | Yes | | | |
| **#14** | Passive AI Output Obfuscation: Decreasing the fidelity of model outputs provided to the end user can reduce an adversary's ability to extract information about the model and optimize attacks for the model. | | | | | Yes | Yes | | [17] - AML.M0002 | |
| ***7- Think about an organizational strategy*** | | | | | | | | | | |
| **#15** | Vulnerability Scanning: Vulnerability scanning is used to find potentially exploitable | | | Yes | | | Yes | | [17] - AML. | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | software vulnerabilities to remediate them.<br>File formats such as pickle files that are commonly used to store AI models can contain exploits that allow for arbitrary code execution. These files should be scanned for potentially unsafe calls, which could be used to execute code, create new processes, or establish networking capabilities. Adversaries may embed malicious code in model corrupt model files, so scanners should be capable of working with models that cannot be fully de-serialized. Both model artifacts and downstream products produced by models should be scanned for known vulnerabilities. | | | | | | | | M0016 | |
| ***8- Preventive measures*** | | | | | | | | | | |

# 10 Appendix 2 - Remediation

| steps | Sub-phase of the Lifecycle of an AIS | Action to Check | State (to be checked) |
|---|---|---|---|
| **Governance & Crisis Management** | Planning and Design | Define AI security requirements and regulatory frameworks (*RGPD, ANSSI, NIST CSF*). | ☐ |
| | | Implement governance integrating Security by Design | ☐ |
| | | Develop an AI incident management plan (policies, procedures, roles and responsibilities). | ☐ |
| | | Define the traceability and auditability mechanisms for AI models (activity logs, decision logs of models) | ☐ |
| | | Assess AIS-specific risks using methods like EBIOS Risk Manager to identify AI threats. | ☐ |
| | | Define access control and authentication policies for AI models | ☐ |
| | Data Collection and Processing | Establish data governance and control their provenance. | ☐ |
| | | Define a protocol for continuous monitoring of AI data flows | ☐ |
| | | Verify the quality of AI datasets and prevent data poisoning. | ☐ |
| **Detection & Investigation** | Model Construction / Adaptation | Audit the robustness of AI models and detect vulnerabilities. | ☐ |
| | | Check the integrity of pre-trained models and external dependencies. | ☐ |
| | | Detect adversarial attacks (Model Stealing, Data Poisoning, Backdoor Attacks). | ☐ |
| | Testing, Evaluation and Verification | Perform adversarial testing and verify resistance to attacks. | ☐ |
| | | Check the robustness of the AI model against drifts and manipulations. | ☐ |
| | | Monitor SIEM logs and *Threat Intelligence* AI to detect threats. | ☐ |
| | Provision / Deployment | Secure deployment pipelines and restrict unauthorized access. | ☐ |
| | | Activate a containment plan to isolate compromised AI models. | ☐ |
| | | Notify the relevant authorities and teams (ANSSI, CNIL, CERT-FR). | ☐ |

| | | | |
|---|---|---|---|
| | Operation and Maintenance | Implement advanced monitoring to detect AI compromises in real time. | ☐ |
| | | Analyze logs and events to identify the cause and extent of the attack. | ☐ |
| **Remediation & Reconstruction** | Provision / Deployment | **#1** Apply the E3R methodology (Containment, Eviction, Eradication, Reconstruction): | ☐ |
| | | **#2** Isolate compromised AI models by removing them from production pipelines. | ☐ |
| | | **#3** Activate a degraded mode /*safe mode* | ☐ |
| | | **#4** Restrict access to impacted datasets | ☐ |
| | | **#5** Block the exfiltration of sensitive data linked to AI models | ☐ |
| | | **#6** Perform an initial damage assessment through analysis of SIEM logs and indicators of compromise (IoC). | ☐ |
| | | **#7** Revoke access keys and change all *credentials* associated with AI models and infrastructures. | ☐ |
| | | **#8** Remove potential backdoors implanted in AI models or APIs | ☐ |
| | | **#9** Disable user accounts or services that were compromised during the attack | ☐ |
| | | **#10** Check network configurations and enforce strict segmentation to limit future exploitation | ☐ |
| | | **#11** Reset CI/CD and MLOps pipelines to ensure no compromised automated processes reintroduce vulnerabilities | ☐ |
| | Operation and Maintenance | **#12** Clean corrupted AI data and retrain models. | ☐ |
| | | **#13** Apply patches and strengthen security configurations. | ☐ |
| | | **#14** Verify the integrity of models and validate their security before redeployment. | ☐ |
| | | **#15** Apply stress testing and simulated attacks to ensure that patched vulnerabilities are no longer exploitable. | |
| | | **#16** Establish post-incident monitoring to prevent recurrence. | ☐ |
| **Continuous Improvement** | Decommissioning / Scrapping | Securely delete obsolete AI models and logs. | ☐ |
| | | Carry out a final audit before decommissioning the AIS. | ☐ |
| | | Document incidents and update AI security policies (structured feedback). | ☐ |

| | | | |
|---|---|---|---|
| | RETEX & Formation | Organize adversarial simulations (Red Team AI) to test the robustness of the systems. | ☐ |
| | | Improve AI threat detection and response models. | ☐ |

**Useful contacts**

| Organisation | Role | Lien |
|---|---|---|
| ANSSI (France) | Strategic and operational guides for remediation | https://www.ssi.gouv.fr |
| CERT-FR | Technical support and incident reporting | https://www.cert.ssi.gouv.fr |
| CNIL (France) | Notification of personal data breaches | https://www.cnil.fr |
| PRIS (ANSSI) approved service providers | Specialized intervention for incident response | https://cyber.gouv.fr/prestataires-de-reponse-aux-incidents-de-securite-pris |
| ENISA (Europe) | European advice and best practices | https://www.enisa.europa.eu |
| Cloud or external IT provider | Technical support for hosted systems | Specific supplier contact |
| In-house legal team | Legal support for communication and compliance | Internal contact for the legal team |

# 11 Acknowledgments

## 11.1 Coordinators

- Alexandre Coroir, Cybersecurity Consultant.
- Carine Thérond, Technical Leader, Stormshield.
- General Patrick Perrot, AI advisor to the Ministry of the Interior's Cyberspace Command, National Gendarmerie.
- Françoise Soulié-Fogelman, Scientific Advisor, Hub France IA.

## 11.2 Contributors

- Nada Amini, Data Scientist, Société Générale.
- Patrick Boutard, CEO, infAIrence.
- Martin d'Acremont, Cybersecurity Consultant, Wavestone.
- Matthieu Ferrandez, Datascience & AI Manager, CyberDefense, I-Tracing.
- Bruno Grieder, Technical Director, Cosmian.
- Soufiane Kaissari, Research Promotion Officer, GLIMPS.
- Camille Maindon, Cybersecurity Consultant, Capgemini Invent.
- Aurélien Mayoue, AI research engineer, CEA.
- Eric Savignac, Cybersecurity and AI Expert.
- Christos Katsoukalis, Data Scientist, Société Générale.
- Thierno Kante, Data Scientist, Edicia.
- Jean-Marc Schenkel, Cyber Security Expert.
- Michael Slimani, Cybersecurity Expert.

## 11.3 Proofreaders

- Gérald Aroulanda, CEO, YMUNIT & Risk Hunter.
- Pauline Bir, Senior Project Manager, Hub France IA.
- Alexandre Coroir, Cybersecurity Consultant.
- Stephan Cohen, Cybersecurity Specialist, BNPP.
- Alexandre Gakic, AI & Cybersecurity Expert.
- Hervé Léon, Cybersecurity Expert.
- Maxime de Jabrun, VP Cyber risk & AI, HeadMind Partners.
- Thomas Kernem-Om, AI & Cybersecurity Senior PM, Hub France IA.
- Françoise Soulié-Fogelman, Scientific Advisor, Hub France IA.

## 11.4 The final touch

- Mélanie Arnould, Operations Manager, Hub France IA.
- Thomas Kernem-Om, AI & Cybersecurity Senior PM, Hub France IA.

CAMPUS CYBER

HUB FRANCE IA

# Analysis of attacks on AI systems

January 2026