RÉPUBLIQUE
FRANÇAISE
Liberté
Égalité
Fraternité

SAFETY RECOMMENDATIONS FOR A GENERATIVE IA SYSTEM

ANSSI-PA-102
29/04/2024

ANSSI GUIDE

INTENDED AUDIENCE :

Developer Administrator CISO CIO User

Information

Attention

This document, written by ANSSI, is entitled **"Recommandations de sécurité pour un sys- tème d'AI générative" (Security recommendations for a generative AI system)**. It can be downloaded from `cyber.gouv.fr`.

It is an original ANSSI production, licensed under the "Open License v2.0" published by the Etalab mission.

In accordance with the Open License v2.0, the document may be freely reused, provided that its authorship is acknowledged (source and date of last update). Reuse includes the right to communicate, disseminate, redistribute, publish, transmit, reproduce, copy, adapt, modify, extract, transform and exploit, including for commercial purposes. Unless otherwise stipulated by law, the recommendations are not prescriptive; they are provided "as is" and are adapted to the threats present at the time of publication. Given the diversity of information systems, ANSSI cannot guarantee that this information can be used without adaptation on the target information systems. In all cases, the relevance of the implementation of the elements proposed by ANSSI must first be validated by the system administrator and/or the people in charge of information systems security.

Document evolution :

VERSION	DATE	NATURE OF CHANGES
1.0	29/04/2024	Initial version

Table of contents

1 Background 3

1.1 Introduction......3

1.2 Definitions 4

1.3 Perimeter 5

2 Summary 6

3 Description of a generative AI system 7

3.1 Life cycle of a generative AI system......7

3.2 Architecture of a generative AI system 10

4 Attack scenarios on generative AI 11

5 Recommendations 14

5.1 General recommendations 14

5.2 Recommendations for the training phase......21

5.3 Recommendations for the deployment phase......22

5.4 Recommendations for the production phase......23

5.5 The special case of AI-assisted source code generation......26

5.6 Special case of AI services for the general public on the Internet 27

5.7 Special case of using third-party generative AI solutions 28

List of recommendations 30

Bibliography 32

1
Context

1.1 Introduction

Artificial intelligence (AI) has long been a theme in the field of re- search, but the possibilities offered by computing power and massive data processing have opened up new opportunities. Among these, there has been a boom in products that can generate an answer to a question formulated in natural language from a model trained on very large volumes of data. These AI models are generally referred to as *Large Language Models* (LLMs) and fall into the category of *generative AI* (see definitions in section 1.2).

The recent craze for these products and services, some of which are now readily available to the general public, has prompted organizations (companies, administrations) to consider the potential productivity gains that could result.

While this technology opens up new perspectives in the organization of work, it requires vigilance and caution when it comes to deployment and integration into existing information systems. Indeed, the deployment of generative AI tools generates new threats that can have a significant impact, for example on the confidentiality of the data they process, but also on the integrity of the information systems with which they are connected.

The purpose of this document is to provide safety recommendations for the implementation of generative AI solutions based on LLMs within public and private entities.

1.2 Definitions

Generative AI

Generative AI is a subset of artificial intelligence, focused on the creation of models that are trained to generate content (text, images, videos, etc.) from a specific corpus of training data.

Large Language Model

A category of generative AI models that can generate text close to the natural speech of a human being, and which are generally trained on a large en- sembly of data.

AI model

In the context of this guide, an AI model refers to a neural network and its parameters (weights, bias [1]).

AI system

An AI system encompasses all the technical components of an application based on an AI model: implementation of this AI model, front-end services for users, databases, logging, etc.

Request

A *prompt* is the text instruction sent by the user to the AI system.

Opposing attack

An *adversarial attack*, sometimes also called an "antagonistic attack" or "attack by contradictory examples", aims to send one or more malicious requests to an AI system, with the aim of deceiving or altering its proper functioning.

[1] In a neural network, a weight is a power coefficient of the connection between 2 neurons, which is adjusted throughout the training phase. A bias is a constant linked to a neuron, allowing "compensation" in the calculation of the result.

1.3 Perimeter

This document deals mainly with the following use cases:

- synthesis or summary of a body of documentation;
- information extraction or text generation from a corpus of documents;
- conversational agents [2] (also known as *Chatbot*);
- source code generation for application developers.

The documentary corpus identified may be "multimodal", i.e. it may involve various categories of input data: text, images, sound, video, etc. However, the guide focuses mainly on textual output generation, and does not deal specifically with image or video generation (even though most of the recommendations are applicable to these use cases).

This corpus of documentation integrates the model's training data, but can also be based on additional data or documents supplied directly as input by the user.

This document only deals with securing a generative AI system architecture based on an LLM.

Safety issues related to the data *quality* [3] and *performance* [4] of an AI model are not covered in this document.

Similarly, while other issues such as ethics, privacy, intellectual property, protection of business secrecy and protection of personal data also need to be taken into account when designing an AI model, they do not fall within ANSSI's area of expertise and are therefore not addressed in this guide.

For all these topics, we refer to the work of ENISA [6, 7], BSI [1], NIST [15, 16] and CNIL [2].

ANSSI has also co-signed an NCSC-UK paper [14] on securing AI in no- vember 2023.

[2] A conversational agent is defined here as an application that enables a written exchange between the user and the AI system, rather than an oral exchange.

[3] Data quality generally refers to business criteria. Data quality criteria from a business point of view can be, for example, origin, quantity, completeness, relevance, accuracy, representativeness (in the statistical sense), or enc or compliance with a given structure.

[4] The performance of an AI model is also a business concept highly dependent on the objectives set when the model was designed. It can include a number of factors, such as accuracy, relevance or the speed of responses generated for users, for example.

2

Summary

The implementation of a generative AI system can be broken down into 3 cyclical phases: an initial phase of training the AI model from specifically selected data, then an integration and deployment phase, and finally an operational production phase in which users can access the trained AI model, via the AI system.

Each of these 3 phases requires specific security measures, which depend in part on the choice of subcontractor for each component (hosting, model training, performance testing, etc.), as well as on the sensitivity of the data used at each stage, and the criticality of the AI system in terms of its purpose.

In addition to the classic threats inherent in any information system, a generative AI system may be subject to specific attacks aimed, for example, at disrupting its proper operation (adversarial attacks) or exfiltrating data processed by it.

The issue of data protection, particularly with regard to training data, is therefore an essential part of a generative AI system, with the corollary problem of the need to know about users when they query the model. Indeed, the model is designed to generate an answer from all the data it has accessed during training, as well as additional data that may come from sensitive internal sources.

The use of a generative AI system must therefore meet confidentiality requirements (*it should be remembered that sending sensitive data to consumer tools on the Internet*[5] *is to be avoided*), as well as integrity and availability requirements. The AI system's interactions with other applications or IS components must therefore be secure, limited to strictly operational requirements, and must be capable of being controlled by a human when they are critical to the organization.

Certain specific uses, such as AI-assisted application development, raise major issues and must therefore be carefully framed (with great vigilance over sensitive modules or applications), controlled by humans and regularly tested (with automatic source code analysis tools).

Finally, the protection of AI models can be just as much of an issue as data protection, not only for reasons of protecting the nation's scientific and technical potential (academic research, models used for national security, etc.), but also because an attacker with knowledge of a model's architecture and parameters can potentially improve his attack capabilities for other purposes (data exfiltration, etc.).

[5] Examples include *ChatGPT*, *Gemini* and *DeepL* for translation.

3

Description of a generative AI system

Attention

The life cycle and architecture presented in this chapter are given as examples to facilitate understanding of the recommendations. They are not intended to be prescriptive. In particular, the sequencing of the functions presented may vary and, depending on the use case, these functions are not always implemented in an AI system.

3.1 Life cycle of a generative AI system

Figure 1 shows an example of the life cycle of a generative AI system.

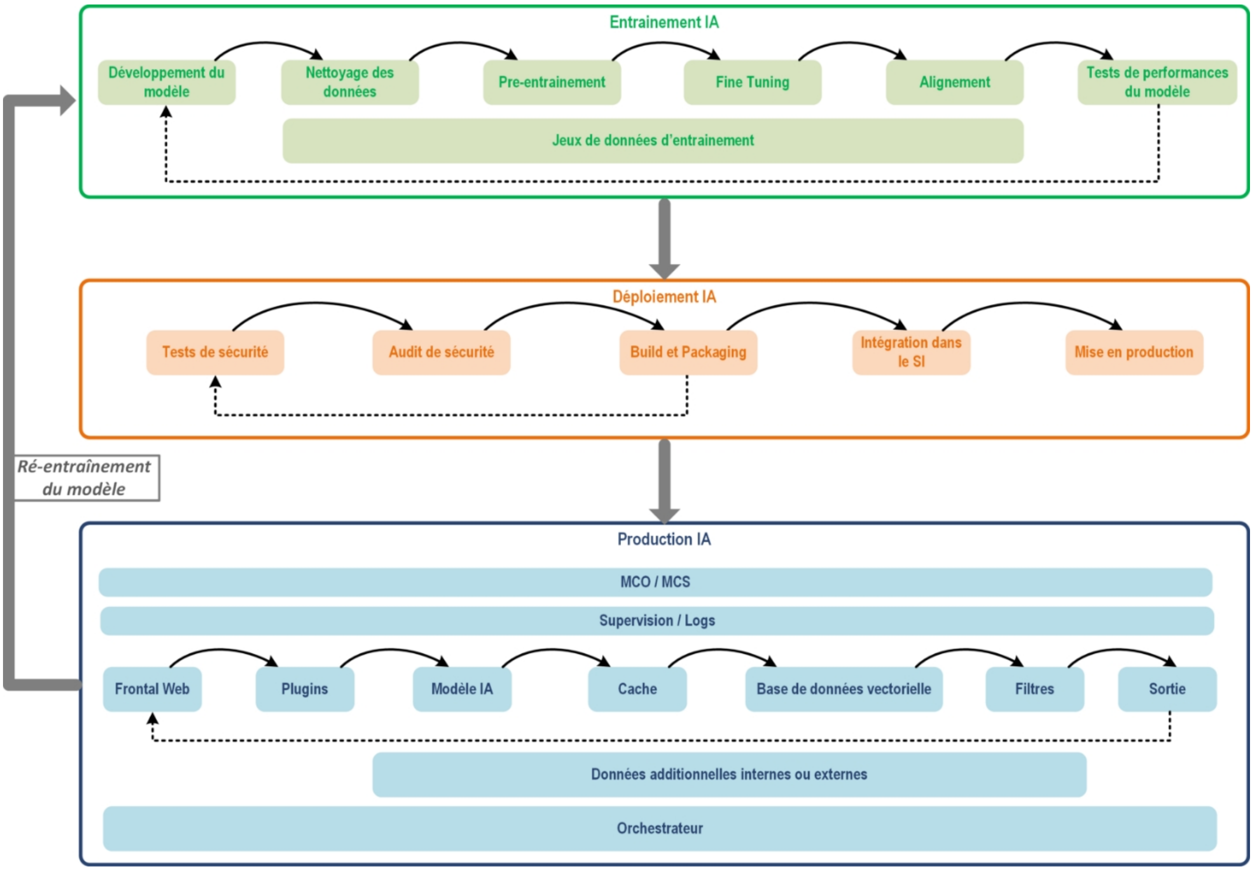

FIGURE 1 - Example of the life cycle of a generative AI system

The 3 phases of training, deployment and production [6] potentially involve different environments and different users. It is important that these 3 phases in the lifecycle of a generative AI system each receive special attention from a safety point of view.

These 3 phases can be carried out in distinct environments, for example, the training phase in a public *cloud* and the deployment and production phase in-house. Nevertheless, appropriate security measures must be applied regardless of the chosen environment.

The re-training of an AI model presented in this diagram does not generally involve repeating all the steps presented in the training phase (very often, only the *fine-tuning* or alignment steps are performed).

Figure 2 shows examples of how responsibilities are shared throughout the design phases of a generative AI system.

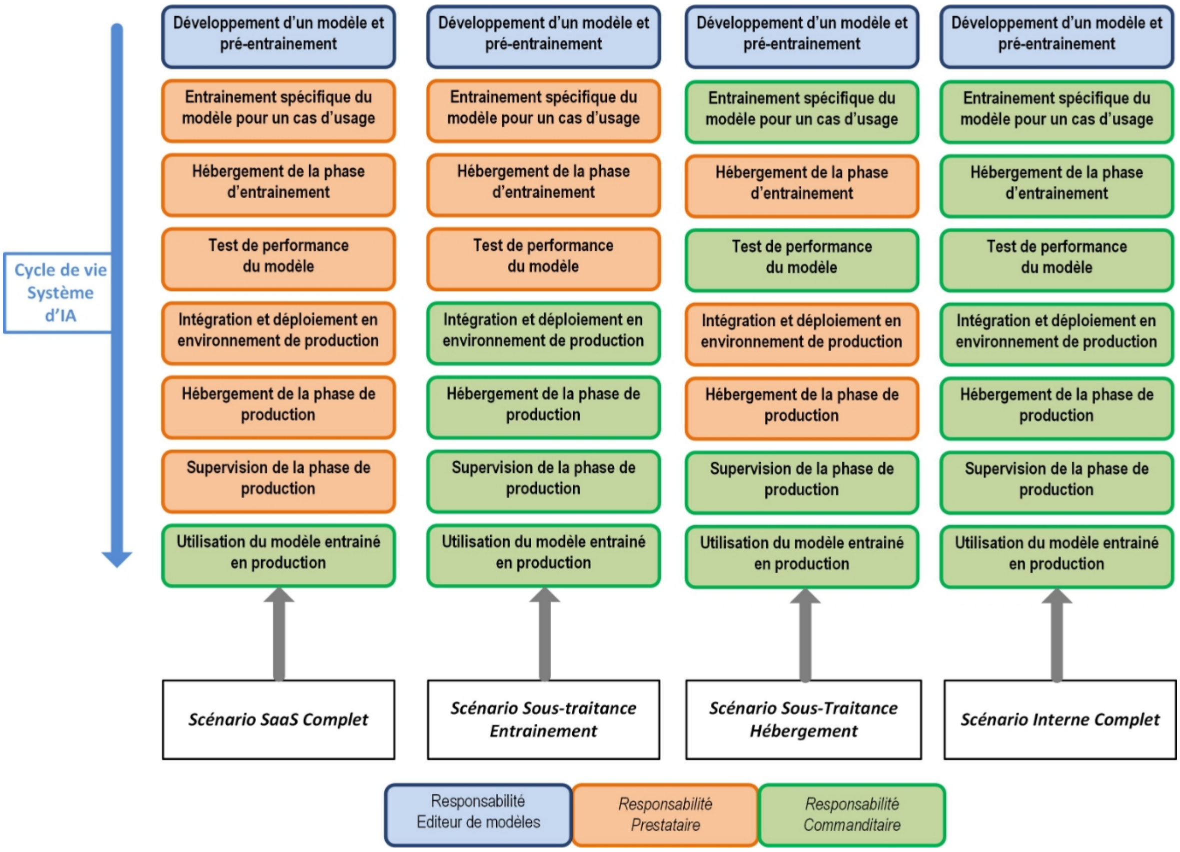
Cycle de vie Système d'IA
Développement d'un modèle et pré-entrainement
Entrainement spécifique du modèle pour un cas d'usage
Hébergement de la phase d'entrainement
Test de performance du modèle
Intégration et déploiement en environnement de production
Hébergement de la phase de production
Supervision de la phase de production
Utilisation du modèle entrainé en production
Scénario SaaS Complet
Scénario Sous-traitance Entrainement
Scénario Sous-Traitance Hébergement
Scénario Interne Complet
Responsabilité Editeur de modèles
Responsabilité Prestataire
Responsabilité Commanditaire

FIGURE 2 - Shared responsibility scenarios in a generative AI system

Safety risks and impacts will be assessed according to the scenario chosen by the organization.

[6] This production phase can sometimes be called the AI model inference phase, i.e. the model makes predictions for given users.

Figure 3 describes the integration of a generative AI system into an IS, and the points of attention to be taken into account with regard to internal and external interactions.

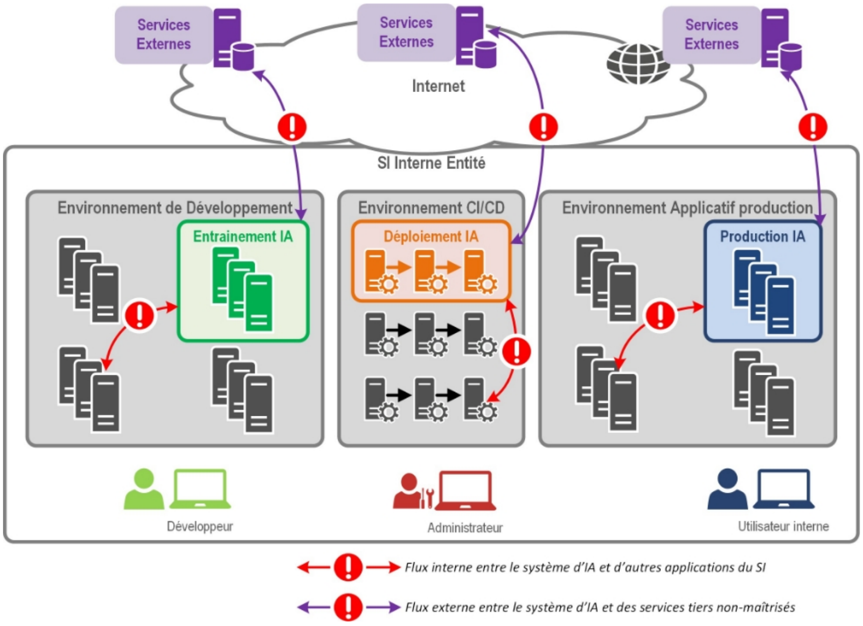
Services Externes
Services Externes
Services Externes
Internet
SI Interne Entité
Environnement de Développement
Entrainement IA
Environnement CI/CD
Déploiement IA
Environnement Applicatif production
Production IA
Développeur
Administrateur
Utilisateur interne
Flux interne entre le système d'IA et d'autres applications du SI
Flux externe entre le système d'IA et des services tiers non-maîtrisés

FIGURE 3 - Integrating a generative AI system into an existing information system

Particular attention must be paid to these interactions, and they must be included in the scope of analysis at every stage of the project.

R1

Integrating safety into all phases of the AI system lifecycle

Safety measures must be identified and applied in each of the 3 phases of the AI system lifecycle: training, deployment and production. These measures are highly dependent on the responsibility-sharing scenario adopted and the associated subcontracting. They must also take into account interactions with other applications or components, both internal and external to the IS.

You can refer to the ANSSI hygiene guide [17] for a basic security foundation to be applied.

3.2 Architecture of a generative AI system

Figure 4 shows an example of the generic architecture of a generative AI system.

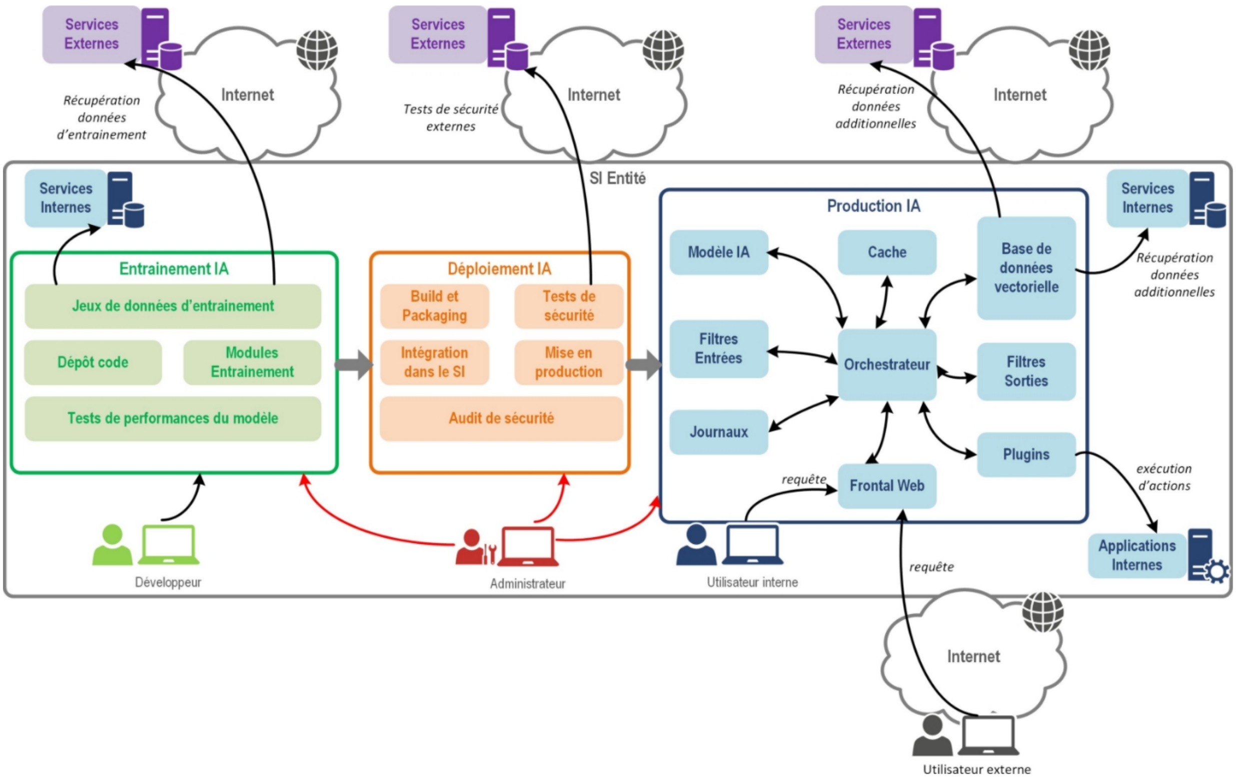

FIGURE 4 - Example of the generic architecture of a generative AI system

This architecture does not exhaustively cover all the components of a generative AI system, but aims to identify potential attack paths targeting an entity.

There are several important elements in this diagram:

- the different populations who can access an AI system at different stages: users, developers, administrators, auditors, etc. ;
- the vector database, which is generally used to store additional data indexing data in vector form, with the aim of enriching [7] user queries before sending them to the model (a concept known as RAG - *Retrieval Augmentation Gene- ration*). This database can be built from data sources internal to the organization or external from partner sources;
- filters at the input and output of the AI model, providing a defense in depth against malicious requests or undesired behavior of the AI system;
- *plug-ins* or additional components, which can be used to connect the AI system to other business or technical resources within or outside the entity.

[7] A vector database is particularly useful in the context of LLMs, as it can enable comparisons to be made, relationships between objects to be identified, and context to be understood.

4

Attack scenarios on generative AI

A generative AI system is first and foremost a standard business application, which must have the same security foundation as any other business application within the entity. However, in addition to this security base, the entity must take into account threats specific to a generative AI system.

These threats can be broken down into 3 main categories of attack[8] :

- **Manipulation** attacks: these consist in hijacking the behavior of the AI system in production by means of malicious requests. They can lead to unexpected responses, dangerous actions or denial of service;
- **Infection attacks**: these involve infecting an AI system during its training phase, by altering training data or inserting a backdoor;
- **Exfiltration attacks**: these involve stealing information from the AI system in production, such as the data used to train the model, user data or internal model data (parameters).

In the context of generative AI, these attacks can affect the following security requirements:

- **Confidentiality**: the aim is to protect an AI system against the leakage of information considered sensitive: training datasets, user queries, model parameters, additional internal data, etc... ;
- **Integrity**: the aim is to protect an AI system against unanticipated changes to its behavior. Integrity can relate directly to the model (parameters), or to training data sets (poisoning), or to the technical components that enable the AI system to function properly: scripts [9], external libraries (*supply chain attack*), service configurations, etc. ;
- **Availability**: the aim is to protect an AI system against denial of service or actions designed to degrade its performance (malicious requests);
- **Traceability**: the aim is to guarantee the explicability [10] and imputability of actions carried out on an AI system. These elements can facilitate investigation and re-mediation after a security incident.

[8] These categories are taken from the CNIL's taxonomy on the subject: https://linc.cnil.fr/petite-taxonomie-des- attaques-des-systemes-dia.

[9] These scripts can be, for example, *fine-tuning* scripts for the AI model, or deployment or maintenance scripts for the AI system.

[10] As defined by the CNIL, explicability is the ability to relate and make comprehensible the elements taken into account by the AI system to produce a result.

Figure 5 describes some examples o f attacks on a generative AI system in an IS.

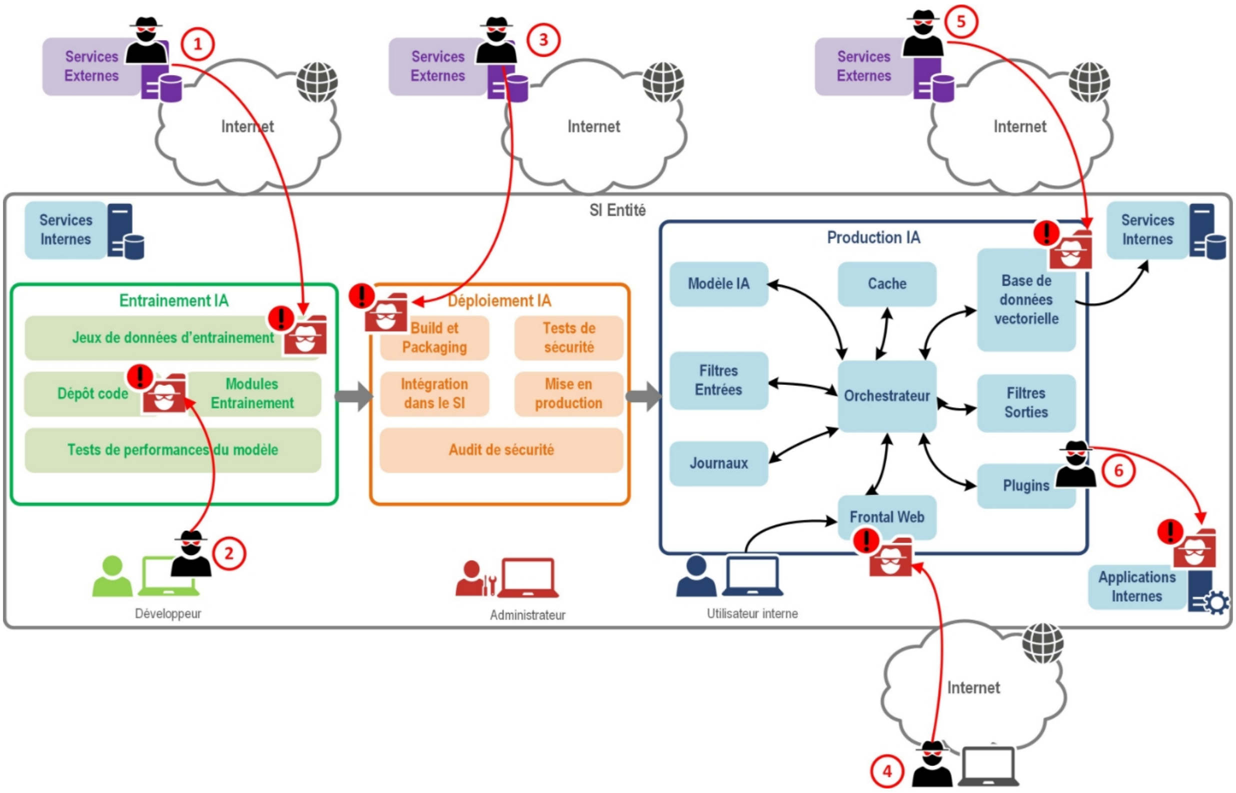

FIGURE 5 - Attack scenarios on a generative AI system in an IS

1. the attacker has access to a data source and poisons the data used to train the AI model beforehand, enabling its use to be diverted once the AI system is in production (e.g.: triggering a malicious action in a hidden way from a specific query);
2. the attacker has access to the development environment and inserts a backdoor into the AI system's code (e.g.: directly modifies the model parameters or the configuration of a technical component of the AI system);
3. the attacker has access to an external pre-deployment test service and hijacks the integration process (e.g.: sends a misleading or malicious result to the integration chain);
4. the attacker uses an adversarial attack technique to exfiltrate sensitive data processed by the AI model (e.g. retrieving training data or requests from other service users), or makes malicious requests to cause a denial of service;
5. the attacker has access to an external resource accessed by the AI system and sends a malicious response that is integrated by the model (e.g.: he sends a URL that points to a malicious website he controls);
6. the attacker gains access to a *plugin* used by the AI system and injects malicious commands when performing an action towards a business application (e.g.: inserting malicious code into the body of an e-mail generated by the AI system).

In the context of this guide (LLM generative AI), the following impacts can be identified:

- tarnishing the reputation of services exposed to the general public by impairing the proper functioning of generative AI systems (e.g. *Chatbot*);
- exfiltration of sensitive data from generative AI systems;
- stealing parameters (weights) from proprietary AI models [11];
- lateralization of an attack to other business applications interconnected to generative AI systems (e.g. internal messaging);
- sabotage business applications by injecting vulnerabilities into AI-generated source code.

Leakage of an entity's sensitive data must in all cases be a threat to be taken into account, whatever the use case of the generative AI system. The AI system must integrate the issue of access rights and need-to-know into its responses.

Particular attention should also be paid to indirect attack scenarios involving an AI system, such as the automatic generation of content for the provision of information (malicious URL insertion).

Finally, the risk analysis must take into account the responsibility-sharing scenario adopted for the project (see Chapter 2). For example, the use of a model trained by a third party may give rise to the risk of a *supply-chain* attack. The untrusted third-party provider may train the model to react differently from the expected behavior when provided with a certain query. One way of reducing the risk could be to audit the model in question, or not to use it for critical applications.

A risk analysis, using the EBIOS-RM method [19] for example, should be carried out as early as possible, i.e. before the training phase.

R2

Conduct a risk analysis on AI systems prior to the training phase

The risk analysis of an AI system must address the following issues:

- map all elements linked to the AI model: third-party libraries, data sources, interconnected applications, etc. ;
- identify the sub-sections of the AI system that will process the organization's data, particularly that contained in user queries;
- take into account the responsibility-sharing scenario and the question of subcontracting for each phase;
- identify the direct and indirect impacts of incorrect or misguided responses from the AI model to users;
- consider the protection of AI model training data.

The recommendations in the next chapter 5 are designed to address these specific threats.

[11] Knowledge of the model's weights can also enable attackers to improve the capability of certain attacks.

5
Recommendations

5.1 General recommendations

The use of uncontrolled external libraries and modules must be studied right from the project design stage, in order to identify potential vulnerabilities linked to these modules. The aim is to provide maximum protection against a so-called *supply-chain-attack* targeting components required for the AI system to function properly. See the ANSSI guide on digital risk [21] or the ICAR documentation on this subject [8].

R3

Evaluate the confidence level of libraries and external modules used in the AI system

It is recommended to map all libraries and ex- ternal modules used in the project, and to assess their level of confidence.

In the same way as for software components (libraries and external modules), it is also essential to evaluate data sources not controlled by the entity. These may be training data sets retrieved from the Internet, model performance validation data sets, or additional data sets used during the production phase.

R4

Assess the confidence level of external data sources used in the AI system

It is recommended to map all external data sources used in the project, and to assess their level of confidence [12].

In general, it is advisable to apply best practices in security-aware development when designing and implementing AI systems. These best practices are sometimes referred to as *DevSecOps* or *security by design*. For further information, please refer to the ANSSI guide [28] on this subject, or the NIST documentation [10], or follow the recommendations of NCSC-UK [5] and CISA [11].

[12] You can use the CNIL criteria (https://www.cnil.fr/fr/tenir-compte-de-la-protection-des-donnees-dans-la-collecte-et-la-gestion-des-donnees) or the *Datasheets for Datasets* proposal (https://arxiv.org/pdf/1803.09010.pdf) to evaluate an external dataset.

R5

Apply DevSecOps principles throughout all project phases

It is recommended to apply best practices for secure development throughout all phases of the project, for example :

- deploy and secure continuous integration and deployment (CI/CD) chains, applying the principle of least privilege for access to CI/CD chain tools;
- implement secure management of secrets used in all phases of the project;
- provide for automated security tests on source code (static code analysis) and during source code execution (dynamic code analysis);
- protect source code integrity and secure access (multi-factor authentication, code signature, access rights, etc.);
- use secure development languages (*fine-tuning* scripts, model development, maintenance, deployment, etc.).

In order to implement an AI model, it is necessary to store the various parameters of this model (weight, bias, etc.) in files. Several formats are available for this purpose, some of which may present a risk of arbitrary code execution, such as those imple- menting serialized object loading functionalities. It is therefore preferable to use formats that strictly separate model parameter data and executable code data.

R6

Use secure AI model formats

We recommend the use of state-of-the-art security formats, such as *safetensor*. Low-security formats such as *pickle* should be avoided.

Generative AI models need to manipulate data throughout their lifecycle. Applying confidentiality protection measures to this data can be complicated, for the following reasons:

- they can come from multiple sources and are sometimes "mixed" in the same set: public data, partner data, internal data, personal data, etc.
- their volume can be very large, particularly in the case of LLM training, which makes them difficult to process;
- It may be necessary to update them regularly, especially when you want to re-train the model;
- it may be necessary to pre-treat them to lower their level of confi- dentiality (anonymization, deletion of certain fields, etc.);
- they can be used in different phases of the project: during the model training phase, but also in production when the model needs to access additional data;

- they can include usage data for the AI system during the production phase: data from user queries and the responses provided by the AI model to these users.

It's important to understand that an AI model inherits the sensitivity of the data used to train it, as well as the data used to re-train it. An AI model may in fact be vulnerable to a phenomenon known as "regurgitation". In some cases, for example, it may generate responses close to the training data, thus revealing potentially sensitive informa- tion.

Depending on the responsibility-sharing scenario adopted (see chapter 2), data confidentiality issues will not be the same, and the technical measures required to prevent data exfiltration will need to be adapted. For example, if the entity wishes to subcontract the training phase to a service provider, it is important to ensure that the data stored and processed by this service provider is sufficiently protected in terms of confidentiality (state-of-the-art encryption, partitioning of resources from other customers, security of keys used, secure deletion after reallocation of resources, etc.).

Similarly, a proprietary AI model that we wish to protect in terms of confidentiality must be subject to specific security measures in the event that it is stored in an untrusted environment (e.g., at a *Cloud* provider or embedded in physically exposed equipment, as in the case of IoT).

R7

Take data confidentiality issues into account right from the AI system design stage

The project study will map all the datasets used in each phase of the AI system: training (training datasets), deployment (test datasets) and production (additional data, vector database, etc.).
This study must include usage data for the AI system in production, i.e. user queries and the answers provided by the AI model. The analysis may also cover the confidentiality of the model's parameters, for example, in the case of proprietary models.

Access to an AI system also complicates the application of the need to know about its users. We need to distinguish several categories of data here:

- **Training data**: due to the design of neural networks, it is not possible to manage user access rights on this data;
- **Additional data in production**: managing access rights is possible, but depends on the possibilities offered by the tools used (RBAC [13]) to store information (internal document management services, vector database, etc.);
- **Usage data**: user queries and responses may contain sensitive data. They are temporarily stored during processing in the AI system and sometimes used to re-train the model (e.g.: alignment with RLHF - *Reinforcement learning from human feedback*).

[13] *Role Based Access Control*

The question of need-to-know must therefore be asked each time the model is re-trained, including on data resulting from use of the model in production (additional business data, user queries, etc.).

R8

Take need-to-know issues into account right from the AI system design stage

It is important to define the model's structuring options upstream of the project, in order to manage the need to know:

- choice of data used for training (without the possibility of managing access rights) and additional data for production (with the possibility of managing roles and access rights);
- the model learning strategy, i.e. when to re-train the model and on the basis of what data (additional business data, user requests, model responses, etc.).

LLM generative AIs are for the most part non-deterministic in their behavior, and may also be subject to *hallucinations* [14]. This uncertainty over the response obtained for a given query implies greater vigilance over the indirect consequences of these responses. Thus, the interactions of an AI system with other IS resources must proscribe the execution of automated actions critical to the organization.

This precautionary principle must prevail, and a generative AI system must not be able to make critical decisions with a strong impact on the business or the protection of goods and people, without human control (e.g. validation in an HMI). In these particular cases, human discernment ca- pacity helps reduce the risk of scenarios that could represent a danger to the organization.

For example, it is important not to use an AI system to automate critical administration actions on the entity's technical infrastructure (e.g.: automatic discovery and deployment of network configurations or firewall rules).

R9

Prohibit the automated use of AI systems for IS-critical actions

An AI system must be configured in such a way as not to be able to perform critical IS actions automatically.
These actions can be critical from a business point of view (banking transactions, production of public content, direct impact on individuals, etc.) or critical actions on the IS infrastructure (reconfiguration of network components, creation of privileged users, deployment of virtual servers, etc.).

The roles and access rights of AI system developers and administrators must be strictly defined and enforced right from the start of the project. Principles of secure administration, such as

[14] Phenomenon in which a model generates erroneous content that is not based on real data.

as described in the ANSSI guide on this subject [26] must be applied in all phases of the AI system lifecycle.

R10

Control and secure privileged access to the AI system for developers and admi- nistrators

All privileged operations on the AI system must comply with best practices for secure administration, including :

- operations to be privileged must be defined and their triggering must be va- lid: re-training, modification of data sets, new interconnection with an application, change of hosting, etc. ;
- privileged operations must be carried out using dedicated accounts and from a dedicated administration workstation;
- the principle of least privilege should be applied, and the use of temporary *tokens* should be favored;
- the development environment must be controlled and managed to the same security level as the production environment.

The hosting of the AI system, whatever its phase, must be studied. The level of security must be consistent with the security needs of the project, and in particular the confidentiality needs of the data used in each phase.

In particular, this point must be strictly adhered to during the model's training phase, since major threats exist during this phase, as discussed in chapter 4.

Attention

AI models are considered to have the same level of sensitivity as the data used to design and train them. Rule R9 [12] of the circular
The "Cloud at the center" approach [13] must be applied in the case of public administration.

R11

Hosting AI systems in trusted environments consistent with security requirements

The hosting of the AI system during the 3 phases of the lifecycle must be consistent with the project's security needs, and in particular its confidentiality and integrity requirements. In particular, the security of model training data (at rest, in transit, during processing) must not be overlooked.

The 3 environments of training, deployment and production of an AI system must be compartmentalized. This measure prevents the risk of lateralization between environments. This is all the more important as the populations with access to each environment are generally not the same.

R12

Separate each phase of the AI system in a dedicated environment

It is recommended to compartmentalize the 3 technical environments corresponding to each phase of the AI system lifecycle. This compartmentalization may concern :

- network partitioning: each environment is integrated into a physically or logically dedicated network;
- system partitioning: each environment has its own dedicated physical serveurs or hypervisors;
- storage partitioning: each environment has its own storage hardware or dedicated disks. At the very least, logical partitioning is applied;
- partitioning of accounts and secrets: each environment has its own user and administrator accounts and distinct secrets.

In the case ofan AI system exposed to the Internet, it is recommended to follow the ANSSI's recommendations for the design of a secure Internet gateway [24].

R13

Implement a secure Internet gateway for AI systems exposed to the Internet

In the case of an AI system exposed on the Internet, it is recommended to follow the good partitioning practices of the ANSSI guide on this subject, in particular :

- dedicate a *reverse-proxy* function before accessing the AI system's web service;
- set up two logical zones for network filtering using firewalls: external filtering at the front end of the Internet, and internal filtering before accessing the AI system;
- do not expose an entity's internal directory for authentication on the AI system;
- avoid pooling security functions (firewalls, *reverse-proxy*, logging server, etc.) that are separate from the secure Internet gateway, on the same hypervisor.

If the entity chooses a public *Cloud* [15] to expose its service, it is advisable to choose a SecNumCloud qualified provider [32] if security requirements demand it.

R14

Favoring SecNumCloud hosting when deploying an AI system in a public cloud

If the entity chooses to host its data in a public *cloud,* it is recommended to use a SecNumCloud trusted offer in the following cases:

- the data processed by the AI system is considered sensitive;

[15] A public *cloud* is a hosting service shared by several customers and exposed on the Internet.

- the impact of the AI system on the business is considered critical;
- AI system users are not considered trustworthy.

When designing a project, it is essential to systematically plan for a degraded mode without AI to meet business needs, in the event of unavailability or failure of the nominal system.

R15

Providing for downgraded business services without an AI system

In order to prevent malfunctions or inconsistencies in the responses ap- ported by the AI model, it is advisable to provide at least one AI system bypass procedure for users, in order to meet business needs.

The deployment of generative AI and LLM systems generally involves the use of GPUs [16] with a view to system performance, whether in the training or production phase.

These GPUs can potentially process sensitive data linked to the operations of the AI model. To protect against data leakage, it is recommended that these GPU hardware components be dedicated to the AI system, and not shared with other IS business applications. GPUs can, however, be shared between several AI models, provided that they correspond to the same level of sensitivity and security requirements .

R16

Dedicating GPU components to AI systems

It is recommended that physical GPU components be dedicated to the processing carried out by the AI system. In the case of virtualization, it is recommended that hypervisors accessing GPU cards be dedicated to the AI system, or at the very least that a hardware filtering function (e.g. IOMMU [17]) be used to restrict virtual machine access to GPU card memory.

Like most business applications, AI systems can be subjected to attacks via auxiliary channels. These attacks can be aimed at exfiltrating sensitive information or disrupting the proper functioning of AI systems. While most of these attacks are not specific to an AI system, some may rely on mechanisms specific to generative AI systems [18].

R17

Take into account auxiliary channel attacks on the AI system

It is advisable to ensure that the AI system is not vulnerable to attacks via auxiliary channels (temporal, consumption, etc.) which could, for example, enable an attacker to reconstruct an answer provided by an AI model.

[16] *Graphics processing unit*
[17] *Input-output memory management unit*
[18] see for example https://cdn.arstechnica.net/wp-content/uploads/2024/03/LLM-Side-Channel.pdf

5.2 Recommendations for the training phase

The issue of data confidentiality was the subject of a general recommendation above (see R7). In particular, and in view of the numerous vulnerabilities published on generative AI tools, it is preferable to start from the premise that a user with access to a trained AI model could potentially have access to the training data of this same model.

To reduce the risks associated with the confidentiality of training data, it is sometimes envisaged to use an anonymization process, or to generate a synthetic dataset from the original raw data. In some cases, these measures can meet the need to protect information, but it is nevertheless important to be alert to the existence of attacks aimed at recovering the original information from anonymized or synthetic data [19]: at- tacks by attribute or membership inference, re-identification from cross-referencing with other datasets, etc.

R18

Train an AI model only with data legitimately ac- cessible to users

It is strongly recommended to train a model with data whose sensi- bility is consistent with the users' need to know.

As we have already seen, attacks specifically targeting the training phase of a model are possible, such as the injection of malicious data into training data sets, or the modification of certain data to generate a dys- functioning of the model, once it has been deployed in production.

R19

Integrity protection for AI model training data

It is advisable to ensure the integrity of model training data throughout the training cycle. This protection can take the form of systematic verification of the signature or *hash* of the files used, or of compressed archives of all this data.

R20

Integrity protection for AI system files

It is advisable to protect the integrity of trained model files, and to regularly check that they have not been altered. By extension, this recommendation also applies to all files inherent to the operation of the AI system (scripts, binaries, etc.).

In the majority of cases, a trained AI model does not require constant modification or adjustment of its parameters. In the event of a malfunction, or if you wish to optimize the model's performance, it is preferable to carry out re-training operations using the training environment dedicated to this purpose.

[19] You can refer to the CNIL's work on this subject: https://linc.cnil.fr/donnees-synthetiques-et-lhomme- crea-les-donnees-son-image-22.

In this respect, continuous learning methods, also known as on-line learning (the model learns in real time from input data), should be avoided wherever possible. Indeed, the use of off-line learning methods, based on selected and tested data sets, reduces the risk of model malfunction or poisoning.

The re-training of an AI model can be carried out on a recurring and fixed basis (e.g. monthly), triggered when a performance gap crosses a given threshold or when the training data are no longer relevant, or on demand on an ad hoc basis.

R21

Prohibit re-training of the AI model in production

It is strongly recommended not to retrain an AI model directly in production. This re-training action should start with the 3-phase cycle, in the appropriate environments for each phase.

5.3 Recommendations for the deployment phase

The deployment of a generative AI system needs to be based on a secure deployment environment, based for example on mastered and hardened CI/CD chains.

These CI/CD chains need to be operated from an administration IS and from dedicated, hardened administrator workstations.

R22

Securing the production deployment chain for AI systems

It is recommended that generative AI systems be deployed from an administration IS, in compliance with the best practices of the ANSSI secure administration guide [26].

A security audit must be carried out by specialized teams trained in the specifics of AI systems. This phase should take place before production deployment, in order to test for vulnerabilities inherent in AI systems (adversarial attacks, etc.).

R23

Safety audits of AI systems before deployment in pro- duction

Robustness and safety tests of AI systems are recommended. These tests can be :

- standard penetration tests on the usual technical components of an AI system: web servers, orchestrator, database, etc.
- security tests on developments made in the AI system (using SAST or DAST tools, for example);
- automated tests 20specifically targeting vulnerabilities linked to AI models (adversarial attacks, model extraction, etc.);
- manual auditor tests specifically aimed at testing the robustness of a generative AI model on more sophisticated attack scenarios.

To carry out safety audits on a generative AI system, you can call on PASSI [30] service providers qualified by ANSSI.

R24

Plan functional tests of AI systems prior to production deployment

It is advisable to test the performance and quality of the answers provided by a generative AI system.

Information

Functional testing of the AI system can take place continuously at a given frequency, and not just during deployment. This can help to detect model malfunctions at an early stage, so that corrections can be made more reactively.

5.4 Recommendations for the production phase

As mentioned above, it is difficult to apply the need-to-know principle to model training data, as the model may be subject to attacks aimed at extracting this data by querying the model (regurgitation).

Similarly, some malicious queries may aim to hijack the generative AI service, for example by inducing hallucinations or erroneous responses.

As part of a defense-in-depth approach, it is advisable to study the possibility of de- tecting or blocking certain malicious requests, for example those aimed at extracting information from the model or additional data (such additional data may include user input in some cases).

This protection may also be relevant to guard against the risk of model leakage. Indeed, if the model has been trained on sensitive data, the leakage of model parameters may lead to the leakage of some of this data by certain attacks (e.g.: model inversion attack or membership inference attack). As such, responses to users should be as simple as possible (string only) and should not return any score vectors used for prediction, or any other mechanism internal to the model.

Finally, depending on the use case, it may be appropriate to define a limit to the size of the responses ap- plied by the AI model. This can reduce the risk of data leakage through regurgitation.

R25

Protect the AI system by filtering user input and output

It is recommended to implement functions to protect against data leakage or model leakage in responses:

- a function to filter malicious user requests before sending them to the

[20] For example, there are several specialized tools such as https://github.com/microsoft/responsible-ai-toolbox, https://github.com/Trusted-AI/adversarial-robustness-toolbox and https://github.com/protectai/ai- exploits.

model;

- a filtering function for queries deemed not legitimate from a business point of view;
- a filter function for internal model information (parameters, training) in responses;
- a filter function for information defined as sensitive in responses (e.g. personal details, project references, etc.);
- a limit on the size of responses (maximum number of characters).

The AI system's interactions with other business applications or other IT resources can be a source of vulnerabilities.

These interactions often take the form of *plug-ins* offered by AI model vendors. These *plug-ins* enable the AI system to be interconnected with office automation tools, social networks or even potentially critical infrastructure components (identity management, network resources, etc.).

These interactions can also facilitate the lateralization of an attacker on the IS, if he or she takes advantage of a vulnerability on the AI system.

The literature often mentions the risks of indirect *prompt injection*, and the problems that can result from sending uncontrolled data to a generative AI model [21] (for example, the content of an e-mail received or a Web page resulting from a search). These uses are all the more problematic in cases where actions could be carried out without human validation (cf. recommendation R9).

It is therefore essential to control the interaction of the AI system with other IS resources .

R26

Control and secure AI system interactions with other business applications

All AI system interactions and network flows must be documented and validated. Network flows between the AI system and other resources must comply with the state of the art in terms of security:

- they must be strictly filtered at network level, encrypted and authenticated (e.g. by following the ANSSI TLS guide [22]);
- they must use secure protocols (e.g. *OpenID Connect*) when using an identity provider [23];
- they must integrate control of access authorizations to the resource as a complement to authentication;
- they must be logged at the appropriate level of granularity.

[21] It is possible to refer to the article "*Not what you've signed up for: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection*" for more details: https://arxiv.org/abs/2302.12173

R27

Limit automatic actions from an AI system dealing with uncontrolled entrées

It is strongly recommended to limit or even prohibit automatic actions on the IS triggered from an AI system and based on uncontrolled inputs (e.g. data from the Internet or e-mails, etc.).

Depending on the use case of the AI system and its criticality from a business point of view, it may be appropriate to deploy it in one or more dedicated environments, i.e. not shared with other business applications within the entity.

R28

Enclose the AI system in one or more dedicated technical environments

It is recommended that the AI system be partitioned into dedicated logical zones, to limit the risk of an attacker who has compromised the system lateralizing.

Logging of actions on the AI system must be carried out with adequate granularity of information, in particular on the inputs and outputs of the AI model.

For the purposes of traceability and understanding of the AI system, it is important to distinguish between requests made by users and the data actually sent to the AI model. Indeed, for reasons of performance and security, user queries may be pre-processed and specifically formatted before being sent to the model.

These two pieces of information are crucial to facilitating incident management, and must be traceable in the AI system's application logs. The aim is to be able to fully reconstruct an event on the AI system, in the event of a malicious request for example.

As far as the architecture of a logging system is concerned, you can refer to , the ANSSI's general guide on the subject [27].

R29

Log all processes carried out within the AI system

It is advisable to log all processing performed on the AI system at the right level of granularity, in particular :

- user requests (taking care to protect them if they contain sensitive data);
- input processing performed on this query before sending it to the model;
- calls to *plugins* ;
- calls to additional data;
- treatment by output filters;
- responses to users.

Attention

The logging of user data must comply with the requirements of the CNIL [9] regarding the protection of personal data (as provided for by the RGPD) and in particular for the length of time this data is kept on the AI system.

5.5 The special case of AI-assisted source code generation

Generative AI tools can be specialized and specifically trained to generate source code in several programming languages.

These means can save developers time, but also entail risks in terms of code quality (introduction of vulnerabilities) or backdoor insertion in the event of an attacker compromising the model.

That's why it's important to be vigilant when it comes to AI-generated source code.

R30

Systematically check AI-generated source code

The source code generated by AI must be subject to safety measures to verify its harmlessness:

- prohibit the automatic execution of AI-generated source code in the development environment;
- prohibit automatic *commit* of AI-generated source code to repositories;
- integrate an AI-generated source code sanitization tool [3, 4] into the development envi- ronment;
- check the safety of libraries referenced in the source code output generated by IA ;
- regular human checks on the quality of the source code generated from sufficiently sophisticated sample queries.

R31

Limit AI source code generation for critical application modules

It is strongly recommended not to use a generative AI tool to generate blocks 22of source code for critical application modules:

- cryptography modules (authentication, encryption, signature, etc.) ;
- modules for managing user and administrator access rights;
- sensitive data processing modules.

R32

Raising developers' awareness of the risks associated with AI-generated source code

Awareness-raising campaigns on the risks associated with the use of AI-generated source code are recommended. This can be based on public reports on the subject, or research papers [23] demonstrating the presence of vulnerabilities in AI-generated code.

In addition, developers can be trained on AI tools for optimizing their requests (*prompt engineering* [24]) to improve the qua- lity and security of the generated code.

Information

Depending on the use case, it may also be appropriate to specifically train a model (alignment step) so that it cannot generate deliberately malicious code.

5.6 Special case of AI services for the general public on the Internet

If the entity wishes to offer a service based on generative AI to the general public, particular care must be taken to secure this service, because of its high exposure.

Damage to an entity's image or reputation can be an additional threat to be identified during risk analysis.

R33

Tougher security measures for consumer AI services exposed on the Internet

It is recommended that special attention be paid to certain safety measures for services exposed to the general public, in particular :

- train the AI model using only public data;
- ensure that AI system users have been authenticated beforehand;
- systematically analyze user queries on the AI system;
- check and validate responses before sending them to users;
- protect the confidentiality of user data (history of requests and responses, etc.);

[22] A block here refers to a complete set of instructions in the source code, e.g. the complete definition of a function, procedure, object class, shell script, and so on.

[23] The reports by *Snyk* (https://snyk.io/fr/reports/ai-code-security/) can be cited as examples, or the research carried out by Stanford University on this subject (https://arxiv.org/pdf/2211.03622.pdf).

[24] For example, it is possible to combine a first request for AI to generate code, followed by a static analysis test of this code, and then a second request asking the AI to correct the vulnerabilities detected in the code that had been generated.

- implement measures against distributed denial of service (DDoS) [18];
- secure the web service at the user's front end [25].

5.7 Special case of using third-party generative AI solutions

In this last section, the guide deals with the special case where the entity is not in a position to manage and control a generative AI service, but is a client of a third-party generative AI service (cf. responsibility-sharing scenarios in chapter 2). The aim of this last section is to remind users of these third-party services of the points to watch out for.

Their ease of use makes it tempting to use generative AI tools available on the Internet to process business data, such as text translation. Sending information (text, images, documents) to a general-public generative AI service is tantamount to depositing the same information on a storage space belonging to them.

The compartmentalization between customers and the confidentiality of data sent to the AI system over the Internet are not under control, and rely solely on trust in the service provider. In this respect, it is important to note that in the majority of offers, data sent to the service is collected and used by the provider for model [25] optimization purposes.

It is therefore essential not to send sensitive data to third-party generative AI services such as *ChatGPT*, *Gemini*, *Copilot*, *DeepL* (Text Translation) or *Perplexity*, to name but a few of the most popular. The data concerned includes :

- data classified as *Restricted* [29] or Defense Classified [31];
- PPST research projects [20];
- personal data (privacy, contact details, etc.);
- contractual, legal or financial data;
- computer secrets, such as passwords or authentication tokens (API keys).

R34

Prohibit the use of generative AI tools on the Internet for pro- fessional use involving sensitive data

As the client entity does not control the generative AI service, it is not possible to ensure that the confidentiality of the data submitted as input meets the entity's security requirements.
As a precautionary measure, it is therefore mandatory never to include sensitive entity data in user queries.

[25] cf. *ChatGPT*'s usage policy, for example: https://help.openai.com/en/articles/7842364-how-chatgpt-and- our-language-models-are-developed

Attention

This recommendation also concerns the use of generative AI tools to generate synthetic datasets for training or *fine-tuning* an AI model.

Some third-party generative AI tools may offer connections with office automation tools or common business applications on the Internet. Particular attention needs to be paid to configuring access rights for generative AI tools to the entity's business data: e-mail, document space, source code repositories, audio and video conferencing services, etc.

R35

Regularly review the configuration of rights for generative AI tools on business applications

It is recommended to review the access rights of generative AI tools as soon as the product is activated in the entity, to ensure that the rights set by default are not too lax or too open by design.

Finally, access rights must be reviewed on a regular basis (e.g. monthly), to ensure that functional and security updates to the product do not impact users' need-to-know.

List of recommendations

R1	Integrating safety into all phases of the AI system lifecycle	9
R2	Conduct a risk analysis on AI systems before the training phase	13
R3	Assess the confidence level of libraries and external modules used in the AI system	14
R4	Assess the confidence level of external data sources used in the AI system	14
R5	Apply DevSecOps principles throughout all project phases	15
R6	Use secure AI model formats	15
R7	Take data confidentiality issues into account right from the system design stage AI	16
R8	Take need-to-know issues into account right from the AI system design stage	17
R9	Prohibit the automated use of AI systems for IS-critical actions	17
R10	Control and secure privileged access for developers and administrators to the AI system	18
R11	Hosting AI systems in trusted environments consistent with requirements safety	18
R12	Separate each phase of the AI system in a dedicated environment	19
R13	Implement a secure Internet gateway in the case of an AI system exposed on Internet	19
R14	Favoring SecNumCloud hosting when deploying an AI system in a public cloud	20
R15	Providing for downgraded business services without an AI system	20
R16	Dedicating GPU components to AI systems	20
R17	Take into account auxiliary channel attacks on the AI system	20
R18	Train an AI model only with data legitimately accessible by users	21
R19	Integrity protection for AI model training data	21
R20	Integrity protection for AI system files	21
R21	Prohibit re-training of the AI model in production	22
R22	Securing the production deployment chain for AI systems	22
R23	Plan safety audits of AI systems before production deployment	23
R24	Plan functional tests of AI systems prior to production deployment	23
R25	Protect the AI system by filtering user input and output	24
R26	Control and secure AI system interactions with other business applications	25
R27	Limit automatic actions from an AI system processing uncontrolled inputs	25
R28	Enclose the AI system in one or more dedicated technical environments	25
R29	Log all processes carried out within the AI system	25
R30	Systematically check AI-generated source code	26
R31	Limit AI source code generation for critical application modules	27

R32 Raising developers' awareness of the risks associated with AI-generated source code 27

R33 Tougher security measures for consumer AI services exposed on the Internet 28

R34 Prohibit the use of generative AI tools on the Internet for professional use im- plicating sensitive data 28

R35 Carry out regular reviews of the configuration of rights for generative AI tools on business applications 29

Bibliography

[1] *BSI - Artificial Intelligence.*
Corporate website, BSI.
https://www.bsi.bund.de/EN/Themen/Unternehmen-und-Organisationen/Informationen-und-Empfehlungen/Kuenstliche-Intelligenz/kuenstliche-intelligenz_node.html.

[2] *CNIL - Artificial intelligence (AI).*
Institutional website, CNIL.
https://www.cnil.fr/fr/intelligence-artificielle-ia.

[3] *NIST - Source Code Security Analyzers.*
Corporate website, NIST.
https://www.nist.gov/itl/ssd/software-quality-group/source-code-security-analyzers.

[4] *OWASP - Source Code Analysis Tools*. Technical report, OWASP.
https://owasp.org/www-community/Source_Code_Analysis_Tools.

[5] *NCSC-UK - Secure development and deployment guidance.*
Corporate website, NCSC-UK, November 2018.
https://www.ncsc.gov.uk/collection/developers-collection.

[6] *ENISA - Artificial Intelligence Cybersecurity Challenges.*
Corporate website, ENISA, December 2020.
https://www.enisa.europa.eu/publications/artificial-intelligence-cybersecurity-challenges.

[7] *ENISA - Securing Machine Learning Algorithms.*
Corporate website, ENISA, December 2021.
https://www.enisa.europa.eu/publications/securing-machine-learning-algorithms.

[8] *CISA - Securing the software supply chain.*
Website, CISA, August 2022.
https://media.defense.gov/2022/Sep/01/2003068942/-1/-1/0/ESF_SECURING_THE_SOFTWARE_SUPPLY_CHAIN_DEVELOPERS.PDF.

[9] *CNIL - AI: how to comply with the RGPD?*
Corporate website, CNIL, April 2022.
https://www.cnil.fr/fr/intelligence-artificielle/ia-comment-etre-en-compliance-with-rgpd.

[10] *NIST - Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities.*
Corporate website, NIST, February 2022.
https://csrc.nist.gov/pubs/sp/800/218/final.

[11] *CISA - Defending Continuous Integration/Continuous Delivery (CI/CD) Environments*. Website, CISA, June 2023. https://media.defense.gov/2023/Jun/28/2003249466/-1/-1/0/CSI_DEFENDING_CI_CD_ ENVIRONMENTS.PDF.

[12] *DINUM - The Cloud for public authorities*. Corporate website, DINUM, May 2023. https://www.numerique.gouv.fr/services/cloud/regles-doctrine/#contenu.

[13] *Doctrine d'utilisation de l'informatique en nuage par l'État - Cloud au center*. Corporate website, LEGIFRANCE, May 2023. https://www.legifrance.gouv.fr/download/pdf/circ?id=45446.

[14] *NCSC-UK - Guidelines for secure AI system development*. Corporate website, NCSC-UK, November 2023. https://www.ncsc.gov.uk/collection/guidelines-secure-ai-system-development.

[15] *NIST - Artificial Intelligence Risk Management Framework*. Corporate website, NIST, January 2023. https://www.nist.gov/itl/ai-risk-management-framework.

[16] *NIST - Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations*. Corporate website, NIST, January 2024. https://csrc.nist.gov/pubs/ai/100/2/e2023/final.

[17] *Guide d'hygiène informatique : renforcer la sécurité de son système d'information en 42 mesures*. Guide ANSSI-GP-042 v2.0, ANSSI, September 2017. https://cyber.gouv.fr/hygiene-informatique.

[18] *Understanding and anticipating DDoS attacks*. Guide Version 1.0, ANSSI, March 2015. https://cyber.gouv.fr/guide-ddos.

[19] *The EBIOS Risk Manager Method - The Guide*. Guide ANSSI-PA-048 v1.0, ANSSI, October 2018. https://cyber.gouv.fr/ebios-rm.

[20] *Protecting the nation's scientific and technical potential*. Guide ANSSI-PA-049 v1.0, ANSSI, April 2018. https://cyber.gouv.fr/guide-zrr.

[21] *Digital risk management - the trust asset*. Guide ANSSI-PA-070 v1.0, ANSSI, November 2019. https://cyber.gouv.fr/publications/maitrise-du-risque-numerique-latout- trust.

[22] *Security recommendations for TLS*. Guide ANSSI-PA-035 v1.2, ANSSI, March 2020. https://cyber.gouv.fr/guide-tls.

[23] *Recommendations for securing the implementation of the OpenID Connect protocol*. Guide ANSSI-PA-080 v1.0, ANSSI, September 2020. https://cyber.gouv.fr/guide-oidc.

[24] *Recommendations relating to the interconnection of an information system to the Internet*. Guide ANSSI-PA-066 v3.0, ANSSI, June 2020. https://cyber.gouv.fr/guide-interconnexion-si-internet.

[25] *Recommendations for implementing a Web site: mastering browser-side security standards*. Guide ANSSI-PA-009 v2.1, ANSSI, April 2021. https://cyber.gouv.fr/guide-sites-web.

[26] *Recommendations for the secure administration of information systems*. Guide ANSSI-PA-022 v3.0, ANSSI, May 2021. https://cyber.gouv.fr/guide-admin-si.

[27] *Security recommendations for the architecture of a logging system*. Guide DAT-PA-012 v2.0, ANSSI, January 2022. https://cyber.gouv.fr/guide-journalisation.

[28] *The essentials - DevSecOps*. Guide Version 1.0, ANSSI, February 2024. https://cyber.gouv.fr/publications/devsecops.

[29] *Interministerial Instruction n°901*. Référentiel Version 1.0, ANSSI, January 2015. https://cyber.gouv.fr/ii901.

[30] *Information systems security audit providers. Référentiel d'exigences*. Référentiel Version 2.1, ANSSI, October 2015. https://cyber.gouv.fr/referentiels-dexigences-pour-la-qualification.

[31] *Instruction générale interministérielle n°1300*. Référentiel, SGDSN, August 2021. https://cyber.gouv.fr/igi1300.

[32] *Cloud computing service providers (SecNumCloud). Référentiel d'exigences*. Référentiel Version 3.2, ANSSI, March 2022. https://cyber.gouv.fr/secnumcloud.

Version 1.0 - 04/29/2024 - ANSSI-PA-102

Licence ouverte / Open Licence (Étalab - v2.0) ISBN:
978-2-11-167156-0 (paper)
ISBN: 978-2-11-167157-7 (digital)
Legal deposit: April 2024

NATIONAL INFORMATION SYSTEMS SECURITY AGENCY

ANSSI - 51 boulevard de La Tour-Maubourg, 75700 PARIS 07 SP cyber.gouv.fr /
conseil.technique@ssi.gouv.fr

RÉPUBLIQUE
FRANÇAISE

Liberté
Égalité
Fraternité

AGENCE NATIONALE DE LA SÉCURITÉ DES SYSTÈMES D'INFORMATION
ANSSI